IDIAP RESEARCH REPORT

END-TO-END ACOUSTIC MODELING USING CONVOLUTIONAL NEURAL NETWORKS FOR AUTOMATIC SPEECH RECOGNITION

Dimitri Palaz Mathew Magimai.-Doss
Ronan Collobert

Idiap-RR-18-2016

JUNE 2016

End-to-End Acoustic Modeling using Convolutional Neural Networks for Automatic Speech Recognition

Dimitri Palaz[1,2], **Mathew Magimai-Doss**[1], **Ronan Collobert**[3,1]
[1]Idiap Research Institute, Martigny, Switzerland
[2]Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[3] Facebook A.I. Research, Menlo Park, USA
dimitri.palaz@gmail.com, mathew@idiap.ch, ronan@collobert.com

Abstract

In hidden Markov model (HMM) based automatic speech recognition (ASR) system, modeling the statistical relationship between the acoustic speech signal and the HMM states that represent linguistically motivated subword units such as phonemes is a crucial step. This is typically achieved by first extracting acoustic features from the speech signal based on prior knowledge such as, speech perception or/and speech production knowledge, and, then training a classifier such as artificial neural networks (ANN), Gaussian mixture model that estimates the emission probabilities of the HMM states. Recent advances in machine learning techniques, more specifically in the field of image processing and text processing, have shown that such divide and conquer strategy (i.e., separating feature extraction and modeling steps) may not be necessary. Motivated from these studies, we propose an end-to-end acoustic modeling approach using convolution neural networks (CNNs), where the CNN takes as input raw speech signal and estimates the HMM states class conditional probabilities at the output. Alternately, in this approach the relevant features and the classifier are jointly learned from the raw speech signal. Through ASR studies and analyses on multiple languages and multiple tasks, we show that: (a) the proposed approach yields consistently a better system with fewer parameters when compared to the conventional approach of cepstral feature extraction followed by ANN training, (b) unlike conventional method of speech processing, in the proposed approach the relevant feature representations are learned by first processing the input raw speech at sub-segmental level ($\approx$ 2 ms). Specifically, through an analysis we show that the filters in the first convolution layer automatically learn ”in-parts” formant-like information present in the sub-segmental speech, and (c) the intermediate feature representations obtained by subsequent filtering of the first convolution layer output are more discriminative compared to standard cepstral features and could be transferred across languages and domains.

1 Introduction

State-of-the-art automatic speech recognition (ASR) systems typically divide the task of recognizing speech into several sub-tasks, which are optimized in an independent manner [1, 2]. Specifically, as a first step, acoustic feature observations, such as Mel frequency cepstral coefficients (MFCCs) or perceptual linear prediction cepstral features (PLPs), are extracted from the short-term speech signal based on speech production and speech perception knowledge. Next, likelihood of subword units, which are typically based on phonemes, are estimated using a statistical model that captures the relationship between the features and subword units in either generative or discriminative manner. Finally, given the likelihood estimates of subword units the best matching word hypothesis is searched by integrating lexical and syntactical constraints.

Recent advances in machine learning have shown that systems can be trained in an end-to-end manner, i.e. systems where every step is *learned* simultaneously, taking into account all the other steps and the final task of the whole system. It is typically referred to as *deep learning*, mainly because such architectures are usually composed of many layers (supposed to provide an increasing level of abstraction), compared to classical "shallow" systems. As opposed to "divide and conquer" approaches presented previously where each step is independently optimized, deep learning approaches are often claimed to lead to more optimal systems. As they alleviate the need of finding the right features by instead training a stack of features in an end-to-end manner, for a given task of interest.

While there is a good success record of such approaches in the computer vision [3, 4, 5] or text processing fields [6], deep learning approaches for speech recognition has largely focussed on the classifier step, where a neural network with many hidden layers is typically trained to classify subword units [7]. These systems still rely on standard short-term spectral-based feature extraction. The training optionally can involve pre-training schemes. In such a case, it is referred to as deep belief neural networks (DBNs) otherwise deep neural networks (DNNs).

More recently, there has been efforts toward modeling raw speech signal with little or no preprocessing [8, 9, 10]. Towards that we proposed a novel approach based on convolution neural networks [11] in [9]. In this approach, the input to the CNN is raw speech signal. The neural network architecture consists of two stages: a feature learning stage consisting of several convolution layer followed by a classifier stage consisting of multilayer perceptron, which are jointly learned by minimizing a cost function based on relative entropy. Phoneme recognition studies on TIMIT corpus showed that the proposed approach is capable of achieving performance comparable or better than standard approach of extraction of cepstral features followed by ANN training.

The present paper builds on our previous work [9, 12, 13] along two directions,

1. From phoneme recognition to automatic speech recognition: a first set of fundamental question that arises is: does the findings on phoneme recognition task apply equally well to continuous speech recognition task across different languages and domains? In that respect, we present investigations on large vocabulary continuous speech recognition task on a variety of corpora that differ in terms of languages, namely, English, Swiss French and Swiss German as well as in terms of variability, namely, read speech and spontaneous speech. In all the studies, we observe that the proposed approach scales well to large vocabulary continuous speech recognition task.

2. Understanding the learned features: As it would be seen in the ASR studies the proposed approach yields a system that performs better than the system based on conventional approach with considerably less number of parameters. Thus, a second set of questions that arise are: what information is the neural network learning and how it is learning? Since the features are learned along with the classifier automatically from the data, yet another question that arises is: are these features domain or language dependent? To understand these aspects we first present analysis of the system that gives insight about: (a) the information that is learned by the filters at the first convolution layer and (b) the information that is modeled by between the first convolution layer and the second convolution layer. We then focus the analysis at the classifier stage, where we show that the learned features can be classified with a simple classifier such as a single layer perceptron. Furthermore, the learned features could be transferred across languages and domains.

The remainder of the paper is organized as follows. Section 2 presents a background survey on features extraction using deep neural networks. Section 3 presents the motivations. Section 4 presents the architecture of the CNN-based system. Section 5 presents the recognition studies and Section 6 presents the analyses. Section 7 presents a discussion and concludes the paper.

2 Background

This section briefly introduces standard hybrid HMM/ANN ASR system. It then presents a concise survey on two aspects of acoustic modeling: features and ANN-based classifier upon which the present paper focusses on.

2.1 Hybrid HMM/ANN ASR system

As presented in Figure 1, hybrid HMM/ANN based ASR system is composed of three parts: features extraction, classification and decoding. In the first step, input features $\mathbf{x}_t$ at time t are extracted from the short-term signal $\mathbf{s}_t$. They are then given as input to a artificial neural network (ANN). In literature, ANNs with different architectures have been proposed such as multilayer perceptron (MLP) [2], time delay neural networks [14] which is also referred to as convolutional neural networks, recurrent neural networks (RNN) [15, 16]. The ANN estimates the class conditional probabilities $P(i|\mathbf{x}_t)$ for each phone class $i \in \{1, \ldots, I\}$. The emission probabilities $p_e(\mathbf{x}_t|i)$ of the HMM states are scaled likelihoods which as given below are obtained by dividing the ANN output by the prior probability of the class $P(i)$,

$$p_e(\mathbf{x}_t|i) \propto \frac{p(\mathbf{x}_t|i)}{p(\mathbf{x}_t)} = \frac{P(i|\mathbf{x}_t)}{P(i)} \; \forall i \in 1, \ldots, I \tag{1}$$

The prior class probability $P(i)$ is often estimated by counting on the training set. The phone classes $\{1, \ldots, I\}$ can be either context-independent phones or clustered context-dependent HMM states, typically obtained by decision tree based state clustering and tying. Depending upon that the system is referred to as either context-independent phone-based ASR system or context-dependent phone-based ASR system, respectively.

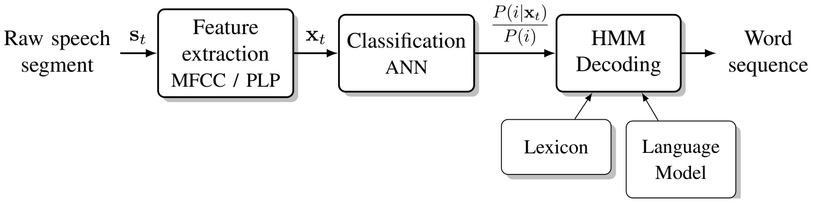


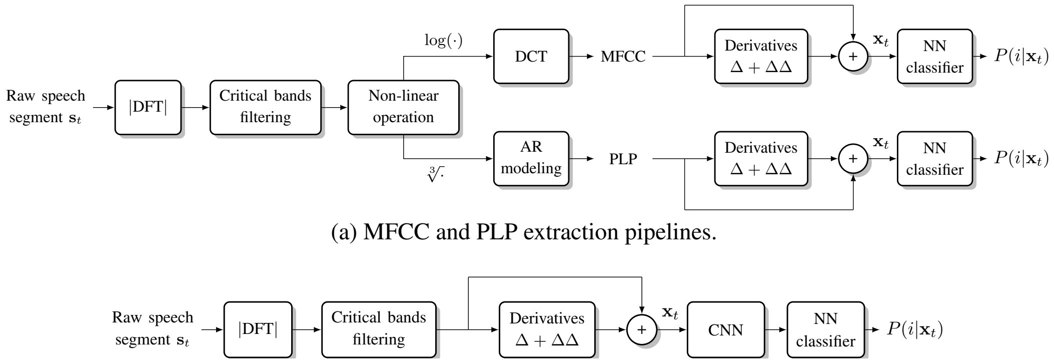
Figure 1: Hybrid HMM/ANN system. $\mathbf{s}_t$ denotes the input speech segment, $\mathbf{x}_t$ here denotes cepstral features and i denotes a phoneme class.

Given the scaled-likelihood estimates, a phonetic lexicon and a language model, the decoder finally infers the best matching word hypothesis through search.

2.2 Feature and classifier

Speech signal is a non-stationary signal. Alternately, the statistical characteristics of the signal change over the time due to various reasons such as speech sound being produced, speaker variation, emotional state variation etc. In the case of ASR, we are primarily interested in the characteristic of the speech signal that relates to or differentiates the speech sounds. In other words, the primary goal is to estimate statistical evidence about speech sounds given the speech signal. To achieve that, guided by statistical pattern recognition techniques, originally the problem has been split into two steps, namely, feature extraction and modeling of the features by a statistical classifier.

Speech coding studies in telephony have shown that speech can be processed as short segments, transformed, transmitted and reconstructed while keeping the intelligibility or message intact [17]. In particular, the studies have shown that short-term speech signal could be considered as output of a linear time invariant vocal tract filter excited by periodic or aperiodic vibration of vocal cords [17]. Furthermore, speech intelligibility can be preserved by preserving the envelop structure of the short-term spectrum of speech signal, which characterizes the vocal tract system [18]. The two most common spectral-based features Mel frequency cepstral coefficient (MFCC) [19] and perceptual linear prediction cepstral coefficient (PLP) [20] are built on those aspects while integrating the knowledge about speech and sound perception.



(a) MFCC and PLP extraction pipelines.

(b) Typical CNN based pipeline using Mel filterbank [22, 23]

Figure 2: Illustration of several features extraction pipelines. |DFT| denotes the magnitude of the discrete Fourier transform, DCT denotes the magnitude of the discrete cosine transform, AR modeling stands for auto-regressive modeling, Δ and $\Delta\Delta$ denote the first and second order derivatives across time, respectively. $P(i|\mathbf{x}_t)$ denotes the conditional probabilities for each input frame $\mathbf{x}_t$, for each label i. It is worth noting that typically, in addition to $\mathbf{x}_t$, the input to the ANN also consists of features from preceding and following frames.

As illustrated in Figure 2(a), the extraction of MFCC or PLP feature involves: (1) transformation of short-term speech signal to frequency domain; (2) filtering the spectrum based on critical bands analysis, which is derived from speech perception knowledge; (3) applying a non-linear operation; and (4) applying a transformation to get reduced dimension decorrelated features. This process only models the local spectral level information on a short time window typically of 20-30 ms. The information about speech sound is spread over time. To model the temporal information intrinsic in the speech signal dynamic features are computed by taking approximate first and second derivative of the static features [21].

To estimate statistical evidence of speech sounds given the speech signal, the cepstral features are modeled by classifiers such as k-means (or vector quantization), Gaussian mixture models, ANNs, k-nearest neighbor. In the beginning of hybrid HMM/ANN theory, the ANNs typically had single hidden layer. There were two particular reasons for that. First, it has been shown theoretically that ANN with single hidden layer is an universal approximator [24]. Second, both acoustic and computing resources were then limited. In the recent years, with the advancements in computing and availability of increased amount of acoustic resources, it has been shown that ANNs with deep architecture, i.e. with multiple hidden layers, can yield better systems [25, 26, 27, 7].

3 Motivation

The standard acoustic modeling mechanism can be seen as a process of applying transformations guided by prior knowledge about speech production and perception on the speech signal, and subsequent modeling of the resulting features by a statistical classifier. More recently, inspired by the success of deep learning approaches in the field of text processing and vision towards building end-to-end systems as well as by the success of DNNs in ASR, researchers have started questioning the intermediate step of feature extraction. In that direction, several studies have been carried where filterbank or critical band energies estimated from the short-term signal instead of cepstral features are used as input of convolutional neural networks based systems [28, 22, 23] or short-term magnitude spectrum is used as input to DNN proposed [29, 30]. Figure 2(b) illustrates a case where, instead of transforming the critical band energies into cepstral features, the critical band energies and its derivatives are fed as input to the ANN.

In this article, we go one step further where the features and the classifier are jointly learned. Alternately, in this approach the raw speech signal is input to an ANN that classifies speech sounds. During training the neural network automatically learns both the relevant features and the classifier.

The output of the trained neural network is then used as emission probabilities of HMM states as done in hybrid HMM/ANN approach. Such an approach can not only be motivated by recent advances in machine learning but also from previous works in the speech literature in which direct modeling of raw speech signal has been proposed for speech recognition.

The first initiative towards directly modeling the raw speech signal was inspired by speech production model, i.e. an observed speech signal can be seen as an output of a time varying filter excited by a time varying source. Specifically, one of the first theoretical work in that direction by [31] was inspired by linear prediction technique, which can deconvolve the excitation source and the vocal tract system through time domain processing. Poritz's work was later revisited as switching autoregressive HMM [32], and more recently in the framework of switching linear dynamical systems [33]. These techniques were investigated in an isolated word recognition setup where word-based models are trained. It was found that in comparison to HMM-based ASR system using cepstral features these approaches yield performance comparable under clean conditions and significantly better performance under noisy conditions [33]. In [34], an approach to model raw speech signal was proposed using auto-regressive HMM. In this approach, each sample of the speech signal is the observation as opposed to a vector of speech samples in the approach proposed in [31]. Each state models the observed speech sample as a linear combination of past samples plus a "driving sequence" (assumed to be a Gaussian *i.i.d* process). The potential of the approach was demonstrated on classification of speaker-dependent discrete utterances consisting of 18 highly confusable stop consonant-vowel syllables. However, their gain compared to conventional cepstral-based features is not clear, and they were never studied on continuous speech recognition task.

More recently, using raw speech signal as input to discriminative systems has been investigated. Combination of raw speech and cepstral features in the framework of support vector machine has been investigated for noisy phoneme classification [35]. Features learning from raw speech using neural networks-based systems has been investigated in [8]. In this approach, the learned features are post-processed by adding their temporal derivatives and used as input for another neural network. Thus, this approach still follows the "divide and conquer" approach. In comparison to that, in our approach, the features are learned jointly with the acoustic model in an end-to-end manner. There are other more recent works that have followed the proposed approach. We discuss them later in Section 7.

4 Proposed CNN-based Approach

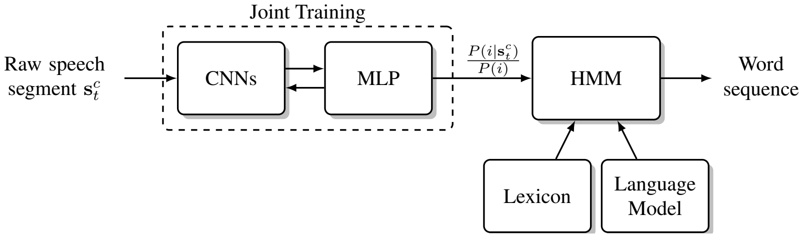

Joint Training
Raw speech segment $\mathbf{s}_t^c$
CNNs
MLP
$\frac{P(i|\mathbf{s}_t^c)}{P(i)}$
HMM
Word sequence
Lexicon
Language Model

Figure 3: Overview of the proposed CNN-based approach.

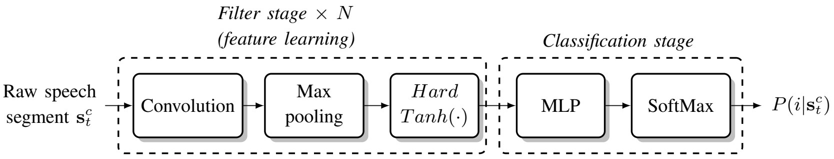


Figure 4: *Overview of the convolutional neural network architecture. Several stages of convolution/pooling/HardTanh might be considered. Our network included 3 stages. The classification stage can have multiple hidden layers.*

We propose a novel acoustic modeling approach based on convolutional neural networks (CNN), where the input speech signal $\mathbf{s}_t^c = \{s_{t-c} \dots s_t \dots s_{t+c}\}$ is a segment of the raw speech signal taken in context of $2c$ frames spanning w_{in} milliseconds. The input signal is processed by several convolution layers and the resulting intermediate representations are classified to estimate $P(i|\mathbf{s}_t^c)$, $\forall i$, as illustrated in Figure 3. $P(i|\mathbf{s}_t^c)$ is subsequently used to estimate emission scaled-likelihood $p_e(\mathbf{s}_t^c|i)$. As presented in Figure 4, the network architecture is composed of several filter stages, followed by a classification stage. A filter stage involves a convolutional layer, followed by a temporal pooling layer and a non-linearity, $HardTanh(\cdot)$. The number of filter stages is determined during training. The feature stage and the classifier stage are jointly trained using the back propagation algorithm.

The proposed approach employs the following understanding:

1. Speech is a non-stationary signal. Thus, it needs to be processed in short-term manner. Traditionally, in the literature guided by Fourier spectral theory and speech analysis-synthesis studies the short-term window size is set as 20-40 ms. The proposed approach follows the general idea of short-term processing. However, the size of the short-term window is a hyper-parameter which is automatically determined during training.
2. Feature extraction is a filtering operation. This can be simply observed from the fact that generic operations such as Fourier transform, discrete cosine transform etc. are filtering operations. In conventional speech processing, the filtering takes place in both frequency (e.g. filter-bank operation) and time (e.g. temporal derivative estimation). The convolution layers in the proposed approach build on these understandings. However, aspects such as the number of filter-banks and their parameters are automatically learned during training.
3. Though the speech signal is processed in short-term manner, the information about the speech sounds is spread across time. In conventional approach, the information spread across time is modeled by estimating temporal derivatives and by using contextual information, i.e. by appending features from preceding and following frames, at the classifier input. In the proposed approach the intermediate representations feeding into the classifier stage are estimated using long time span of input speech signal, which is again determined during training. Alternately, w_{in} is a hyper-parameter.

In essence the proposed approach with minimal assumptions or prior knowledge learns to process the speech signal to estimate $P(i|\mathbf{s}_t^c)$.

4.1 Convolutional layer

While "classical" linear layers in standard MLPs accept a fixed-size input vector, a convolution layer is assumed to be fed with a sequence of T vectors/frames: $\{\mathbf{y}_1 \dots \mathbf{y}_t \dots \mathbf{y}_T\}$. As illustrated in Figure 5, a convolutional layer applies the same linear transformation over each successive (or interspaced by dW frames) windows of kW frames. In this work, $\mathbf{y}_t$ is either a segment of input raw speech $\mathbf{s}_t^c$ (for the first convolution layer) or an intermediate representation output by the previous convolution layer. Formally, the transformation at frame t is written as:

$$M \begin{pmatrix} \mathbf{y}_{t-(kW-1)/2} \\ \vdots \\ \mathbf{y}_{t+(kW-1)/2} \end{pmatrix}, \qquad (2)$$

where M is a $d_{out} \times d_{in}$ matrix of parameters, d_{in} denotes the input dimension and d_{out} denotes the output dimension of each frame. In other words, d_{out} filters (rows of the matrix M) are applied to the input sequence.

4.2 Max-pooling layer

These kind of layers perform local temporal $\max$ operations over an input sequence. More formally, the transformation at frame t is written as:

$$\max_{t-(kW_{mp}-1)/2 \leq k \leq t+(kW_{mp}-1)/2} \mathbf{y}_k^d \qquad \forall d \qquad (3)$$

with $\mathbf{y}$ being the input and $d \in \{1, \cdots d_{out}\}$ These layers increase the robustness of the network to minor temporal distortions in the input.

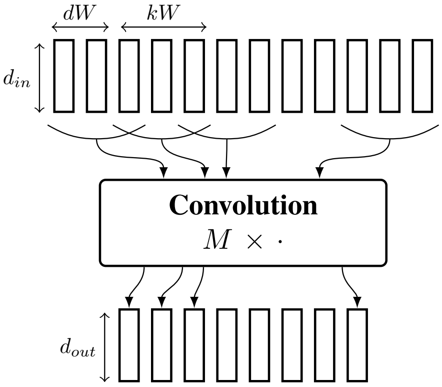


Figure 5: *Illustration of a convolutional layer.* d_{in} *and* d_{out} *are the dimensions of the input and output frames.* kW *is the kernel width (here* $kW = 3$*) and* dW *is the shift between two linear applications (here,* $dW = 2$*).*

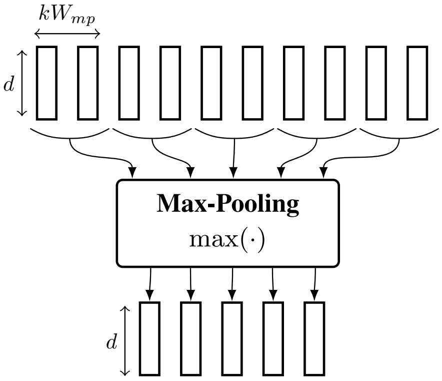


Figure 6: *Illustration of max-pooling layer.* kW *is the number of frame taken for each* $\max$ *operation (here,* $kW_{mp} = 2$ *and* $dW_mp = 2$*) and* d *represents the dimension of input/output frames (which are equal).*

4.2.1 Non-linearity

This kind of layer applies a non-linearity to the input. In this work, we use the $HardTanh$ layer, defined as:

$$HardTanh(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases} \tag{4}$$

4.3 SoftMax layer

The $Softmax$ [36] layer interprets network output scores $f_i(\mathbf{s}_t^c)$ as conditional probabilities, for each class label i:

$$P(i|\mathbf{s}_t^c) = \frac{e^{f_i(\mathbf{s}_t^c)}}{\sum\limits_j e^{f_j(\mathbf{s}_t^c)}} \tag{5}$$

4.4 Network training

The network parameters θ are learned by maximizing the log-likelihood $\mathcal{L}$, given by:

$$\mathcal{L}(\theta) = \sum_t \log(P(i|\mathbf{s}_t^c, \theta)) \tag{6}$$

for each speech segment $\mathbf{s}_t^c$ and label i, over the whole training set, with respect to the parameters of each layer of the network. Defining the `logadd` operation as:

$$\underset{j}{\text{logadd}}(z_j) = \log(\sum_j e^{z_j}) \tag{7}$$

the likelihood can be expressed as:

$$\mathcal{L} = \log(P(i|\mathbf{s}_t^c)) = f_i(\mathbf{s}_t^c) - \underset{j}{\text{logadd}}(f_j(\mathbf{s}_t^c)) \tag{8}$$

where $f_i(\mathbf{s}_t^c)$ described the network score for class i. Maximizing this likelihood is performed using the stochastic gradient ascent algorithm [37].

4.5 Illustration of a trained network

In the proposed approach, in addition to the number of hidden units in each hidden layer of the classification stage, the filter stage has number of hyper-parameters, namely, time span of input speech signal w_{in} used to estimate $P(i|\mathbf{s}_t^c)$, number of convolution layers, kernel or temporal window width kW at input of each convolution layer, dW shift of the temporal window at the input of each convolution layer, max pooling kernel width kW_{mp} and shift of max pooling kernel dW_{mp}. In the present work, all of these hyper-parameters are determined during training based on frame level classification accuracy on validation data.

Figure 7 illustrates the trained feature stage of the proposed CNN approach on TIMIT corpus. The details of the training can be found in the following Section 5. The filter stage has three convolution layers and it takes a window of 250 ms speech signal w_{in} as input to estimate $P(i|\mathbf{s}_t^c)$ every 10 ms. The figure also illustrates the temporal information κ modeled by the output of each convolution layer and the temporal shift δ. Briefly, the first convolution layer models in a fine grain manner the changes in the signal characteristics over time, i.e. processes 1.8 ms of speech ($kW = 30$ samples) every 0.6ms ($dW = 10$ samples). The subsequent convolution layers then filter and temporally integrate the output of the first convolution layer to yield an intermediate feature representation that is input to the classifier stage, which eventually yields an estimate of $P(i|\mathbf{s}_t^c)$

It is worth pointing out that the dimensionality of the intermediate representation at the feature learning stage output depends upon the number of convolution stages and the max-pooling kernel width. As it can be seen that max-pooling is done without temporal overlap. So at each convolution stage, in addition to filtering minor temporal distortions, max-pooling operation acts as a down sampler.

5 Recognition Studies

In this section, we present automatic speech recognition studies to show the potential of the proposed approach. We compare it against the conventional approach of spectral-based feature extraction followed by ANN training on different tasks and languages, namely, (a) TIMIT phoneme recognition task, (b) Wall street journal (WSJ) 5k task, (c) Swiss French Mediaparl task and (d) Swiss German Mediaparl task. The objective of these studies is to demonstrate the potential of the proposed end-to-end acoustic modeling approach by comparing it against the standard cepstral feature-based acoustic modeling for estimating phoneme class posterior probability.

The reminder of the section is organized as follows. Section 5.1 presents the different datasets and setup used for the studies. Section 5.2 presents the different systems that are trained and evaluated. Section 5.3 presents the recognition studies.

5.1 Databases and setup

5.1.1 TIMIT

The TIMIT acoustic-phonetic corpus [38] consists of 3,696 training utterances (sampled at 16kHz) from 462 speakers, excluding the SA sentences. The validation set consists of 400 utterances from 50 speakers. The core test set was used to report the results. It contains 192 utterances from 24

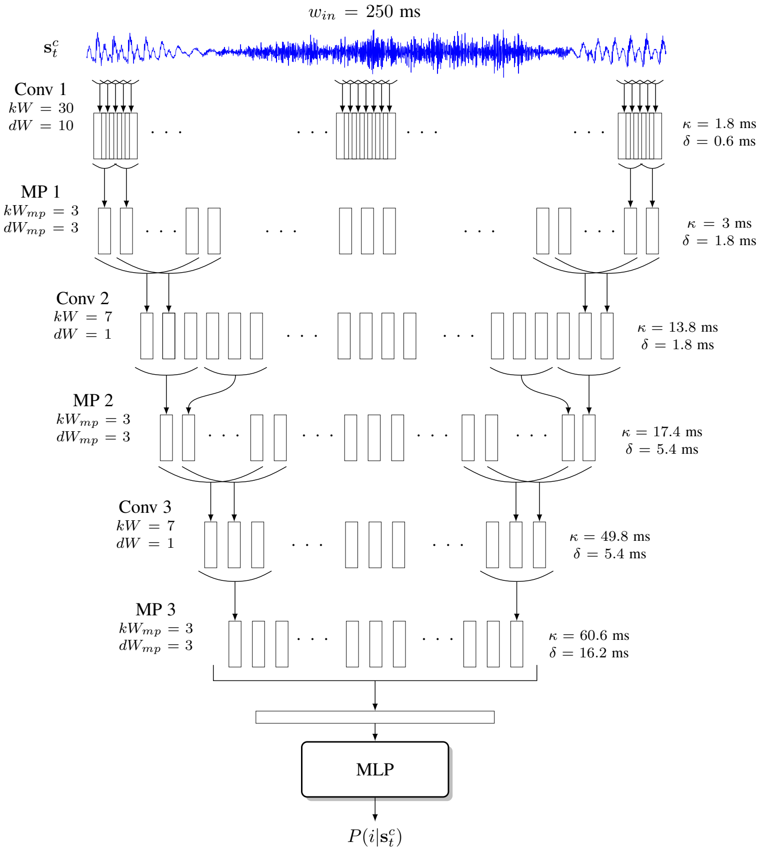


Figure 7: *Illustration of the feature stage of CNN trained on TIMIT to classify 183 phoneme classes. κ and δ indicates the temporal information modeled by the layer and the shift respectively. Non-linearity layers are applied after each max-pooling.*

speakers, excluding the validation set. Experiments were performed using 61 phoneme labels, with three states, for a total of 183 targets as in [39]. After decoding, the 61 hand labeled phonetic symbols are mapped to 39 phonemes, as presented in [40].

5.1.2 Wall Street Journal

The Wall Street Journal (WSJ) corpus is an English corpus consisting of read microphone speech [41]. The SI-284 set of the corpus is formed by combining data from WSJ0 and WSJ1 databases [42]. The set contains 36416 sequences sampled at 16 kHz, representing around 80 hours of speech. Ten percent of the set was taken as validation set. The Nov'92 set was selected as test set. It contains 330 sequences from 10 speakers. The dictionary was based on the CMU phoneme set, 40 context-independent phonemes. We obtained 2776 clustered context-dependent (cCD) units, i.e. tied-states, by training a context-dependent HMM/GMM system with decision tree based state tying using HTK [43]. We used the bigram language model provided with the corpus. The test vocabulary contains 5000 words.

5.1.3 Mediaparl

MediaParl is a bilingual corpus [44] containing data (debates) in both Swiss German and Swiss French which were recorded at the Valais parliament in Switzerland. Valais is a state which has both French and German speakers with high variability in local accents specially among German speakers. Therefore, MediaParl provides a real-speech corpus that is suitable for ASR studies. In our experiments, audio recordings with 16 kHz sampling rate are used.

The Swiss German part of the database, referred to as *MP-DE*, is partitioned into 5955 sequences from 73 speakers for training (14 hours), 876 sequences from 8 speakers for validation (2 hours) and and 1692 sequences from 7 speakers (4 hours) for test. 1101 tied-states were used in the experiments, following the best system available on this corpus [45]. The vocabulary size is 16,755 words. The dictionary is provided in SAMPA format with a phone set of size 57 (including sil) and contains all the words in the train, development and test set. A bigram language model was used.

The Swiss French part of the database, referred to as *MP-FR*, is partitioned into 5471 sequences from 107 speakers for training (14 hours) , 646 sequences from 9 speakers for validation (2 hours) and and 925 sequences from 7 speakers (4 hours) for test. 1084 tied-states were used in the experiments, as presented in [46]. The vocabulary size is 12,035 words. The dictionary is provided in SAMPA format with a phone set of size 38 (including sil) and contains all the words in the train, development and test set. A bigram language model was used.

5.2 Systems

In this section, for each task studied, we present the details of the conventional spectral feature based baseline systems (Section 5.2.1) and the proposed CNN-based system using raw speech signal as input (Section 5.2.2).

5.2.1 Conventional cepstral feature based system

On each task, we have two baseline hybrid HMM/ANN systems which differ in terms of ANN architecture. More precisely, 1 hidden layer MLP (denoted as ANN-1H) based system and 3 hidden layers MLP (denoted as ANN-3H) based system. These ANNs estimate $P(i|\mathbf{x}_t)$ where $\mathbf{x}_t$ is a cepstral feature vector. The details of the baseline systems for the different tasks are as follows,

- TIMIT: We treat the one hidden layer MLP based system and the three hidden layers MLP based system without pre-training i.e. random initialization reported in [29, Figure 6] as the baseline systems. Our motivation in doing so is that they are one of the best cepstral feature based systems reported in the literature on this task. In these systems, the input to the MLPs were 39 dimensional MFCC features ($c_0 - c_{12} + \Delta + \Delta\Delta$) with five frames preceding and five frames following context (i.e. input dimension 39 × 11). ANN-1H has 2048 nodes in the hidden layer and ANN-3H has 1024 nodes in each of the three hidden layers.

- WSJ: We trained an ANN-1H and an ANN-3H to classify 2776 tied-states. The input to the MLP was 39 dimensional MFCC features ($c_0 - c_{12} + \Delta + \Delta\Delta$) with four frames preceding and four frames following context (i.e. input dimension 39 × 9). The MFCC features were computed using HTK with a frame size of 25ms and a frame shift of 10 ms. ANN-1H had 1000 nodes in the hidden layer and ANN-3H had 1000 nodes in each hidden layer. The ANNs were trained using Torch7 toolbox [47].

- MP-DE: We use the setup of the best performing hybrid HMM/ANN system using a three hidden layers MLP classifying 1101 clustered context-dependent units reported in [45] as the baseline ANN-3H system. The ANN has 1000 nodes in each hidden layer. We trained an ANN-1H with 1000 hidden units for the present study using Torch7 toolbox. The inputs to the ANNs were 39 PLP cepstral features ($c_0 - c_{12} + \Delta + \Delta\Delta$) extracted using HTK with four frames preceding and four frames following context. The frame size and frame shift were 25 ms and 10ms, respectively.

- MP-FR: We use the setup of the best performing hybrid HMM/ANN system using a three hidden layers MLP classifying 1084 clustered context-dependent units reported in [46] as the baseline ANN-3H system. The ANN has 1000 nodes in each hidden layer. We trained an ANN-1H with 1000 hidden units for the present study using Torch7 toolbox. The inputs to the ANNs were 39 PLP cepstral features ($c_0 - c_{12} + \Delta + \Delta\Delta$) extracted using HTK with four frames preceding and four frames following context. The frame size and frame shift were 25 ms and 10ms, respectively.

5.2.2 Proposed CNN-based system

We trained the proposed CNN-based $P(i|\mathbf{s}_t^c)$ estimator using raw speech signal. The inputs are simply composed of a window of the speech signal (hence d_{in} = 1, for the first convolutional layer). The utterances are normalized such that they have zero mean and unit variance, which is in line with the literature [34]. No further pre-processing is performed. The hyper-parameters of the network are: the time span of the input signal (w_{in}), the kernel width kW and shift dW of the convolutions, the number of filters d_{out}, maxpooling kernel width kW_{mp}, maxpooling kernel shift dW_{mp} and the number of nodes in the hidden layer(s). Note that the input d_{in} for the first convolution layer is one (i.e. a sample of the speech signal). For the remaining layers, the d_{in} is the product of d_{out} of the previous layer and kW of that layer. These hyper parameters were determined by early stopping on the validation set, based on frame classification accuracy. The ranges which were considered for a coarse grid search are reported in Table 1. We used the TIMIT task to narrow down the hyper-parameters search space, as it provided fast turn around experiments.

Table 1: Range of hyper parameters for the grid search.

Parameters	Units	Range
Input window size (w_{in})	ms	100-700
Kernel width of the first conv. (kW_1)	samples	10-90
Kernel width of the n^{th} conv. (kW_n)	frames	1-11
Number of filters per kernel (d_{out})	filters	20-100
Max-pooling kernel width (kW_{mp})	frames	2-6
Number of hidden units in the classifier	units	200-1500

For each of the tasks, we trained CNNs with one hidden layer (denoted as CNN-1H) and three hidden layers (denoted as CNN-3H) similar to the different MLP architectures in the baseline systems. The CNNs were trained using the Torch7 toolbox [47]. We found that three convolution layers consistently yields the best cross validation accuracy across all the tasks. The CNN architecture found for each of the task is presented in Table 2. The shift of max-pooling kernel $dW_{mp} = 3$ was found for all the layers on all the tasks. As we will observe later, the complexity of the CNN-based approach in terms of number of parameters lies at the classifier stage. So, for fair comparison with the baseline systems, we restricted the search for the number of hidden nodes in the hidden layer(s) such that the number of parameters are comparable to the respective baseline systems. The output classes were same as the case of cepstral feature-based system, i.e. for TIMIT task 183 phone classes, for WSJ task 2776 cCD units, for MP-DE task 1101 cCD units and for MP-FR task 1084 cCD units.

Table 2: Architecture of CNN-based system for different tasks. HL=1 denotes CNN-1H and HL=3 denotes CNN-3H. w_{in} is expressed in terms of milliseconds. The hyper-parameters kW, dW, d_{out} and kW_{mp} for each convolution layer is comma separated. HU denotes the number of hidden units. 3×1000 means 1000 hidden units per hidden layer.

	HL	w_{in}	kW	dW	d_{out}	kW_{mp}	HU
TIMIT	1	250	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	250	30,7,7	10,1,1	80,60,60	3,3,3	3x1000
WSJ	1	210	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	310	30,7,7	10,1,1	80,60,60	3,3,3	3x1000
MP-DE	1	210	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	310	30,7,7	10,1,1	80,60,60	3,3,3	3x1000
MP-FR	1	190	30,7,7	10,1,1	80,60,60	3,3,3	1000
	3	310	30,7,7	10,1,1	80,60,60	3,3,3	3x1000

5.3 Results

In this section we present the results of the studies on different tasks. For the sake of completeness, for the speech recognition studies we also report performance on HMM/GMM system. For MP-DE and MP-FR, the best performing HMM/GMM systems reported in [45] and [46], respectively are presented. They have more number of tied states than the hybrid HMM/ANN and the CNN-based system presented here.

5.3.1 TIMIT

Table 3 presents the results on TIMIT phone recognition task in terms of phoneme error rate (PER). It can be observed that the proposed CNN based approach outperforms the conventional cepstral feature based system. In [29, Figure 6], ANNs with different hidden layers were investigated with cepstral feature as input. The best performance of 23.0% PER for the case of random initialization is achieved with 7 hidden layers, 3072 hidden nodes per layer and 17 frames temporal context (8 preceding and 8 following). With pre-training, the best performance of 22.3% PER is achieved with 6 hidden layers, 3072 hidden nodes per layer and 17 frames temporal context. The CNN-3H system performs better than those systems as well.

Table 3: Phoneme error rate of different systems on the core test set of the TIMIT corpus. The ANN-1H and ANN-3H performances are reported in [29].

Input	System	#Conv. params.	#Class. params.	PER (in %)
MFCC	ANN-1H	na	1.2M	24.5
MFCC	ANN-3H	na	2.6M	22.6
RAW	CNN-1H	63k	920k	22.8
RAW	CNN-3H	52k	2.9M	21.9

Table 4 contrasts our results with a few prominent results on TIMIT using ANNs. Inputs of these systems are either MFCCs (computed as presented in Section 5.2.1), Mel filterbanks energies (abbreviated FBANKs) or "improved" MFCC features (denoted MFCC+LDA+MLLT+fMLLR), which are obtained by applying decorrelation processes (linear discriminant analysis and maximum likelihood linear transform) and speaker normalization (feature-space maximum likelihood linear regression) [48] to the original MFCC coefficient. One can see that the proposed approach outperforms most of the systems using MFCCs features. Systems using improved MFCCs features yields better results than the proposed approach, mainly due to the speaker normalization technique, which could be developed for the proposed approach. Finally, one can see that RNN-based systems (the three last entries of Table 4) clearly yield the best performance.

Table 4: Phoneme error rate of different systems reported in literature on the core test set of the TIMIT corpus.

Method (input)	PER (in %)
Augmented CRFs (MFCC) [49]	26.6
HMM/DNNs 6 layers (MFCC) [29]	22.3
Deep segmental NN (MFCC) [50]	21.9
Proposed approach	**21.9**
HMM/DNNs 6 layers (MFCC+LDA+MLLT+fMLLR) [51]	18.5
CTC transducers (FBANKs) [16]	17.7
Attention-based RNN (FBANKs) [52]	17.6
Segmental RNN (MFCC+LDA+MLLT+fMLLR) [51]	17.3

5.3.2 WSJ

The results for the LVCSR study on the WSJ corpus in presented in Table 5. for the baseline systems and the proposed system. As it can be observed, the CNN-1H based system outperforms the ANN-1H based baseline system, and the CNN-3H based system also outperforms the ANN-3H based system with as many parameters.

Table 5: Word Error Rate on the Nov'92 testset of the WSJ corpus

Input	System	#Conv. params.	#Class. params.	WER (in %)
MFCC	GMM	na	4M	5.1
MFCC	ANN-1H	na	3.1M	7.0
MFCC	ANN-3H	na	5.6M	6.4
RAW	CNN-1H	46k	3.1M	6.7
RAW	CNN-3H	61k	5.6M	5.6

5.3.3 MP-DE

The results on the Mediaparl German corpus are presented in Table 6. The CNN-1H based system outperforms the GMM-based system, the ANN-1H based system and the ANN-3H system with four times less parameters. The CNN-3H system also outperforms the baseline.

Table 6: Word Error Rate on the testset of the MP-DE corpus. The GMM and ANN-3H baseline performances are reported in [45]

Input	System	#Conv. params.	#Class. params.	WER (in %)
PLP	GMM	na	3.8M	26.6
PLP	ANN-1H	na	2.2M	26.7
PLP	ANN-3H	na	8.8M	25.5
RAW	CNN-1H	61k	1.6M	24.4
RAW	CNN-3H	92k	8.7M	23.5

5.3.4 MP-FR

The results on the Mediaparl French corpus are presented in Table 7. Again, a similar trend can be observed, i.e. the CNN-1H based system outperforms the ANN-1H baseline and the CNN-3H outperforms the ANN-3H based system.

Table 7: Word Error Rate on the testset of the MP-FR corpus. The GMM and ANN-3H performances are reported in [46]

Input	System	#Conv. params.	#Class. params.	WER (in %)
PLP	GMM	na	3.8M	26.8
PLP	ANN-1H	na	2.2M	27.0
PLP	ANN-3H	na	8.8M	25.5
RAW	CNN-1H	61k	1.5M	25.9
RAW	CNN-3H	92k	8.7M	23.9

In summary, these studies show that with minimal assumptions the proposed approach is able to learn to process the speech signal to estimate phone class conditional probabilities $P(i|\mathbf{s}_t^c)$ and yield a system that outperforms conventional cepstral feature based system using DNNs. Furthermore, we consistently observe that the CNN-1H system yields performance comparable to ANN-3H system with considerably fewer parameters.

6 Analysis

The aim of this section is to gain insight into the proposed approach. Towards that this section focuses on analysis at two levels: (a) analysis of the first convolution layer (Section 6.1) which operates on the speech signal directly. Thus, can be related to and contrasted with traditional speech processing; and (b) analysis of the intermediate feature representations obtained at the output of the feature stage (Section 6.2).

6.1 First convolution layer

In this section, we present an analysis of the first convolution layer. We first provide an input level analysis, where the hyper-parameters of the layer (found experimentally) are compared against the conventional speech processing approach. We then show that the convolution layer can be interpreted as a bank of matching filters. Finally, we analyze how these filters respond to various inputs and present a method to understand the filtering process.

6.1.1 Input level analysis

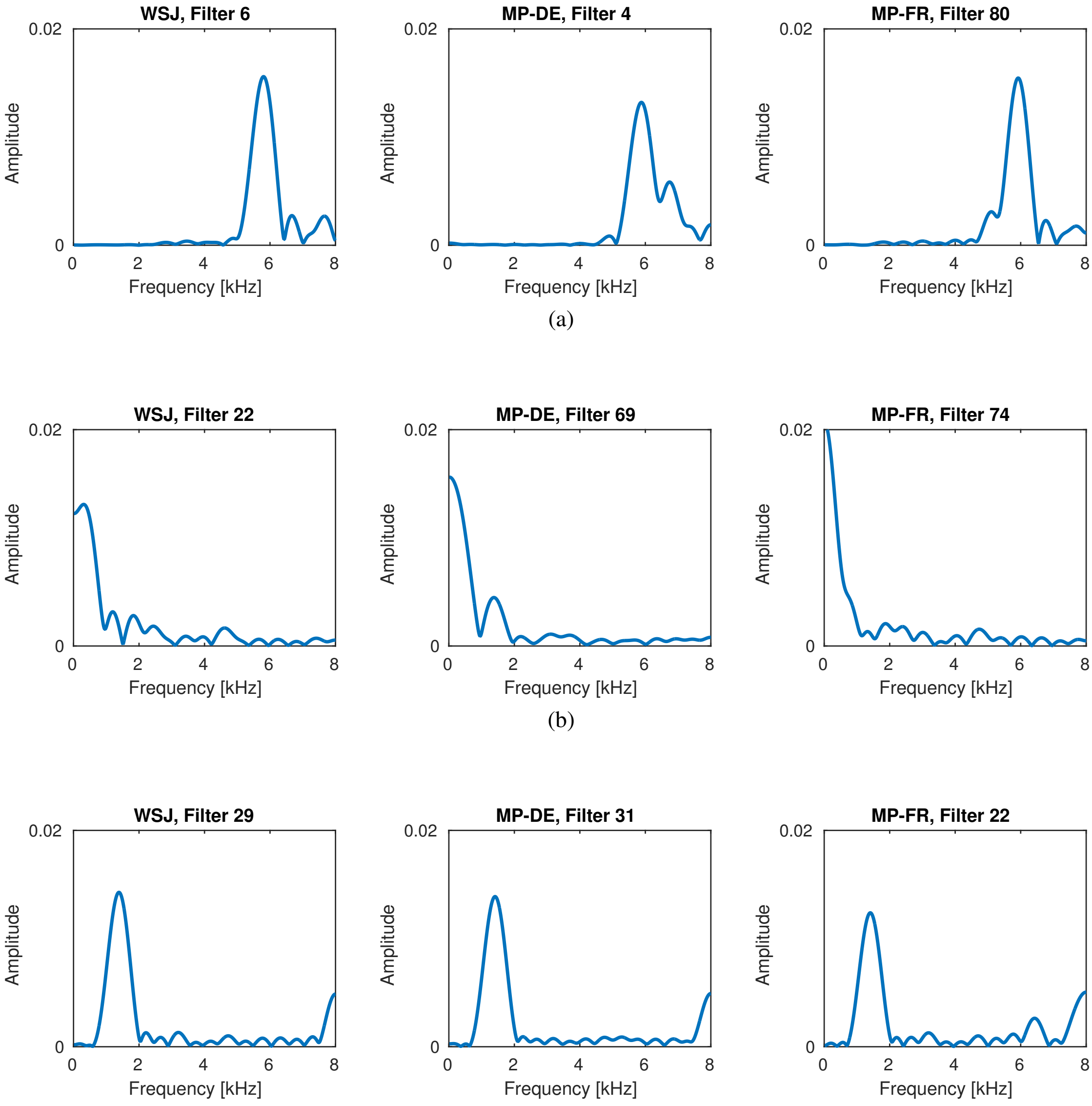
To learn to process raw speech signal and estimate $P(i|\mathbf{s}_t^c)$ the proposed approach employs many hyper-parameters which are decided based on validation data. We can get insight into the approach by relating or contrasting a few of the hyper-parameters to the traditional speech processing. First among that is time span of the signal w_{in} used to estimate $P(i|\mathbf{s}_t^c)$. From Table 2, we can observe that w_{in} varies from 190 ms - 310 ms. This is consistent with the literature which supports the idea of processing syllable length speech signal (around 200 ms) for classification of phones [53]. This aspect can be also observed in another way. Usually, in hybrid HMM/ANN system the input is the cepstral features (static + Δ + $\Delta\Delta$) at the current time frame and features of four preceding frames and four following frames. If the frame shift is 10 ms and the temporal derivatives are computed using two frames preceding and two frames following context then the 9 frame feature input models 170 ms of speech signal.

Next we can understand how the speech signal of time span of 190 ms - 310 ms is processed at the input of the network through the kernel width (kW) and kernel shift (dW) of the first convolution stage. We can see from Table 2 that for all tasks kW is 30 speech samples and dW is 10 speech samples. Given that the sampling frequency is 16 kHz, this translates into a window of 1.8 ms and shift of about 0.6 ms. This is contrary to the conventional speech processing where typically the window size is about 25 ms, the shift is about 10 ms and the resulting features are concatenated at the classifier input. Note that in our case w_{in} is shifted by 10ms, however with in the window of 190 ms - 310 ms the speech is processed at sub-segmental level at the first convolution layer and subsequently processed by later convolution layers to estimate $P(i|\mathbf{s}_t^c)$.

Such a sub-segmental processing at the first convolution layer could possibly be reasoned through signal stationarity assumptions. More precisely, the convolution filters at the first stage are learned by discriminating the phone classes at the output of the CNN. So, for the output of the convolution filter to be informative (for phone classification), the filter has to operate on stationary segments of the speech signal spanned by w_{in}. It can be argued that such a stationary assumption would clearly hold for one glottal cycle or pitch period of the speech signal. In such a case suppose if the limit of the observed pitch frequency is assumed to be 500 Hz, i.e. beyond adult speakers pitch frequency range, then a window size of 2 ms or less would ensure that the filters operate on stationary segments i.e. with in a glottal cycle. This line of argument is also consistent with traditional feature extraction methods which tend to model the smooth envelop of the short-term spectrum, i.e. information more related to vocal tract response, with quasi-stationarity assumptions.

6.1.2 Learned filters

The first convolution layer learns a set of filters that operates on the speech signal in a similar way to filter bank analysis during MFCC or PLP cepstral feature extraction. In the case of MFCC or PLP cepstral feature extraction the number of filter banks and their characteristics are determined a priori using speech perception knowledge. For instance, the filters are placed either on Mel scale or on Bark scale. Further, each of the filters cover only a part of the bandwidth, out of which the response is strictly zero. The number of filters are chosen based on bandwidth information. For instance, in the case of Mel scale around 24 filters for 4 kHz bandwidth (narrow band speech) and 40 filters for 8 kHz bandwidth (wide band speech) are typically used. While in the case of Bark scale, there are 15 filters for 4 kHz bandwidth and 19 filters for 8 kHz bandwidth [54].



(a)

(b)

(c)

Figure 8: *Examples of three close pairs of filters learned. The left column is from CNN-1H WSJ, the center one is from CNN-1H MP-DE, the right one is from CNN-1H MP-FR.*

In contrast, in the proposed approach the number filters and their responses are automatically learned in data-driven manner, i.e., while learning to estimate $P(i|\mathbf{s}_t^c)$. It can be observed from Table 2 that the number of filters for all the tasks is 80. This is well above the range typically used in speech processing. In order to understand the learned filter characteristics, we analyzed the filters learned on WSJ, MP-DE and MP-FR task in the following manner:

(i) The complex Fourier transform $\mathcal{F}$ of the filters learned on the WSJ, MP-DE and MP-FR tasks for CNN-1H case are computed using 1024 point FFT. The 512 point magnitude spectrum $|\mathcal{F}_m|$ of each filter m is then normalized, i.e. converted into a probability mass function. F_m denotes the normalized magnitude spectrum of filter m.

(ii) For each filter $m = 1, \ldots, 80$ learned on WSJ, we find the closest filter $n = 1, \ldots, 80$ learned on MP-DE and MP-FR using symmetric Kullback-Leibler divergence,

$$d(F_m, F_n) = \frac{1}{2} \cdot [D_{KL}(F_m || F_n) + D_{KL}(F_n || F_m)], \tag{9}$$

$$D_{KL}(F_m || F_n) = \sum_{u=1}^{512} F_m^u \ln \frac{F_m^u}{F_n^u}, \tag{10}$$

where F_m^u is the normalized magnitude at u^{th} point of FFT of filter m of WSJ CNN-1H and F_n^u is the normalized magnitude at u^{th} point of FFT of filter n of MP-DE CNN-1H or MP-FR CNN-1H.

Figure 8 presents normalized frequency responses of a few filters learned on WSJ (on the left column) and the closest filters learned on the MP-DE task (on the middle column) and on the MP-FR task (on the right column). We can make two observations. First, the filters are focussing on different parts of the spectrum. However, unlike the filter banks in the MFCC or PLP cepstral feature extraction, the frequency response of the filters cover the whole bandwidth. Second, it can be observed that similar filters can be found across domain and languages, although there is a difference in the spectral balance, especially as observed in the case of Figure 8(b).

To further understand the characteristics of the learned filters, we estimated the cumulative frequency response of all the filters in the filterbank:

$$F_{cum} = \sum_{n=1}^{80} F_n \tag{11}$$

Figure 9 presents the gain normalized cumulative frequency responses for CNN-1H WSJ, CNN-1H MP-DE and CNN-1H MP-FR. We can make three key observations,

(i) Emphasis is given to frequency regions below 3500 Hz (telephone bandwidth) and high frequency region in the range of 6000 Hz - 8000 Hz.

(ii) Though the filters are learned on different languages and corpora, we can see that below 4000 Hz and above 6500 Hz the frequency response for WSJ, MP-DE and MP-FR are similar. As the filters are operating on sub-segmental speech, we speculate that the peaks (high energy regions) are more related to the resonances in the vocal tract or phoneme discriminative invariant information. Between 4000 Hz and 6500 Hz, we can see that MP-DE and MP-FR have responses that closely match, but are different than WSJ. Overall we observe that the spectral balance for WSJ is different than for MP-DE and MP-FR. We attribute this balance mismatch mainly to the fact that the WSJ and the Mediaparl corpora are different domains in terms of type of speech (read vs. spontaneous) and recording environment (controlled vs real world). In the following sub-section and Section 6.2.2 we touch upon this aspect again.

(ii) Auditory filterbanks such as Mel scale filterbanks or Bark scale filterbanks are usually designed to have a cumulative frequency response that is flat. In other words, constant Q bandpass filterbank. In contrast to that, it can be seen that the cumulative frequency response of the learned filters is not constant Q bandpass. The main reason for that is standard filterbanks emerged from human sound perception studies considering the complete auditory frequency range or the bandwidth, so as to aid analysis and synthesis (reconstruction) of the audio signal. However, in our case these filters are learned for the purpose of discriminating phones, and the speech signal contains information other than just phones. The figure suggests that, for discriminating only phones, constant Q bandpass filterbank is not a necessary condition.

6.1.3 Response of filters to input speech signal

In Section 6.1.1, we observed that the speech signal of time span 190 ms - 310 ms is processed in sub-segmental manner. In the previous section, we observed that the filters that operate on sub-segment of speech signal are tuned to different parts of the spectrum during training. In other words,

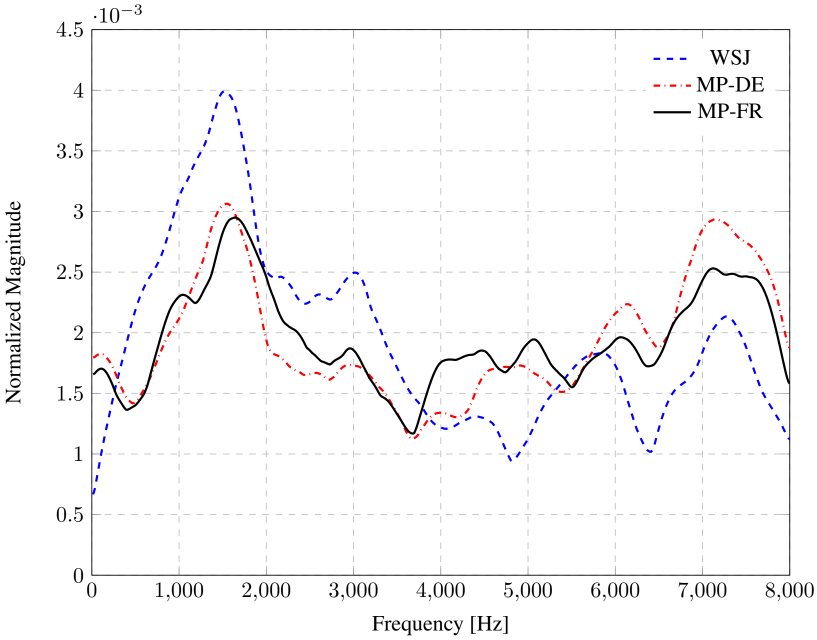


Figure 9: *Cumulative frequency responses of the learned filterbank on WSJ, MP-DE and MP-FR.*

matched to different parts of the spectrum relevant for phone discrimination. In this section, we ascertain that by analyzing the response of the filters to the the input speech signal in relationship with phones.

The CNNs in the WSJ, MP-DE and MP-FR studies were trained to classify cCD units, which can be quite distinctive across languages. So, in order to facilitate the analysis across languages, we trained CNNs with single hidden layer on WSJ, MP-DE and MP-FR data to classify context-independent phones with same hyper parameters. We denote these CNNs as CNN-1H-mono WSJ, CNN-1H-mono MP-DE and CNN-mono MP-FR, respectively.

As a first step, we analyzed the energy output of the filters to the input speech signal. Formally, for a given input $\mathbf{s}_t = \{s_{t-(kW-1)/2} \dots s_{t+(kW-1)/2}\}$, the output $\mathbf{y}_t$ of the first convolution layer is given by:

$$\mathbf{y}_t[m] = \sum_{l=-(kW-1)/2}^{l=+(kW-1)/2} f_m[l] \cdot s_{t+l} \quad \forall m = 1, .., d_{out} \tag{12}$$

where f_m denotes the m^{th} filter in first convolution layer and $\mathbf{y}_t[m]$ denotes the output of the filter at time frame t. Figure 10 presents the output of the filters of CNN-1H-mono WSJ given a segment of speech signal corresponding to phoneme $/I/$ as input. It can be seen that at each time frame only a few filters out of the 80 filters have high energy output. An informal analysis across different phones showed similar trends, except that the filters with high energy output were different for different phones. Together with the findings of the previous section, this suggests that the learned filters in could be a *dictionary* that models the information in the frequency domain *in-parts* for each phone. With that assumption, we performed extended the analysis where,

1. the magnitude spectrum or frequency response $\mathcal{S}_t$ of the input signal $\mathbf{s}_t$ based on the dictionary is estimated as:

$$\mathcal{S}_t = |\sum_{m=1}^{M} \mathbf{y}_t[m] \cdot \mathcal{F}_m|, \tag{13}$$

   where $\mathbf{y}_t[m]$ is the output of filter m as in Equation (12) and $\mathcal{F}_m$ is the complex Fourier transform of filter f_m.

   It is worth noting that if the filter-bank was to correspond to a bank of Fourier sine and cosine bases then $\mathcal{S}_t$ is nothing but the Fourier magnitude spectrum of the input signal $\mathbf{s}_t$.

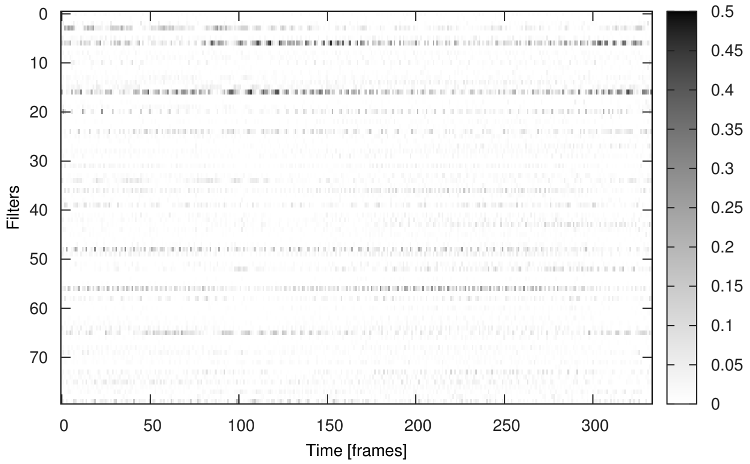


Figure 10: *Normalized energy output of each filter in the first convolution layer of CNN-1H-mono WSJ for an input speech segment corresponding to phoneme $/I/$.*

As $\mathbf{y}_t[m]$ would be a projection on to the Fourier basis corresponding to discrete frequency m, and $\mathcal{F}_m$ would *ideally* be a Dirac delta distribution centred at the discrete frequency m.

2. gain-normalized magnitude spectrum $\mathcal{S}_t$ is computed and averaged across different frames and speakers for each phone. The resulting average magnitude spectrums for the phones are then compared.

We performed the analysis on the validation data of WSJ, MP-DE and MP-FR using the filters in the first convolution layer of respective CNN-1H-mono. Figure 11 displays the magnitude spectrums of a few prominent vowels (notated in SAMPA format) for WSJ, MP-DE and MP-FR. It can be observed that the average power spectrum is capturing envelop of the sub-segmental speech and it is different for each vowel. This difference is particularly observable in the frequency regions below 4000 Hz and in the frequency regions between 6000 Hz and 8000 Hz. We had earlier observed in Section 6.1.2 that these are frequency regions that the learned filters give emphasis to. The prominent spectral peaks could be related to the formants. However, a detailed formant analysis is practically infeasible for three main reasons:

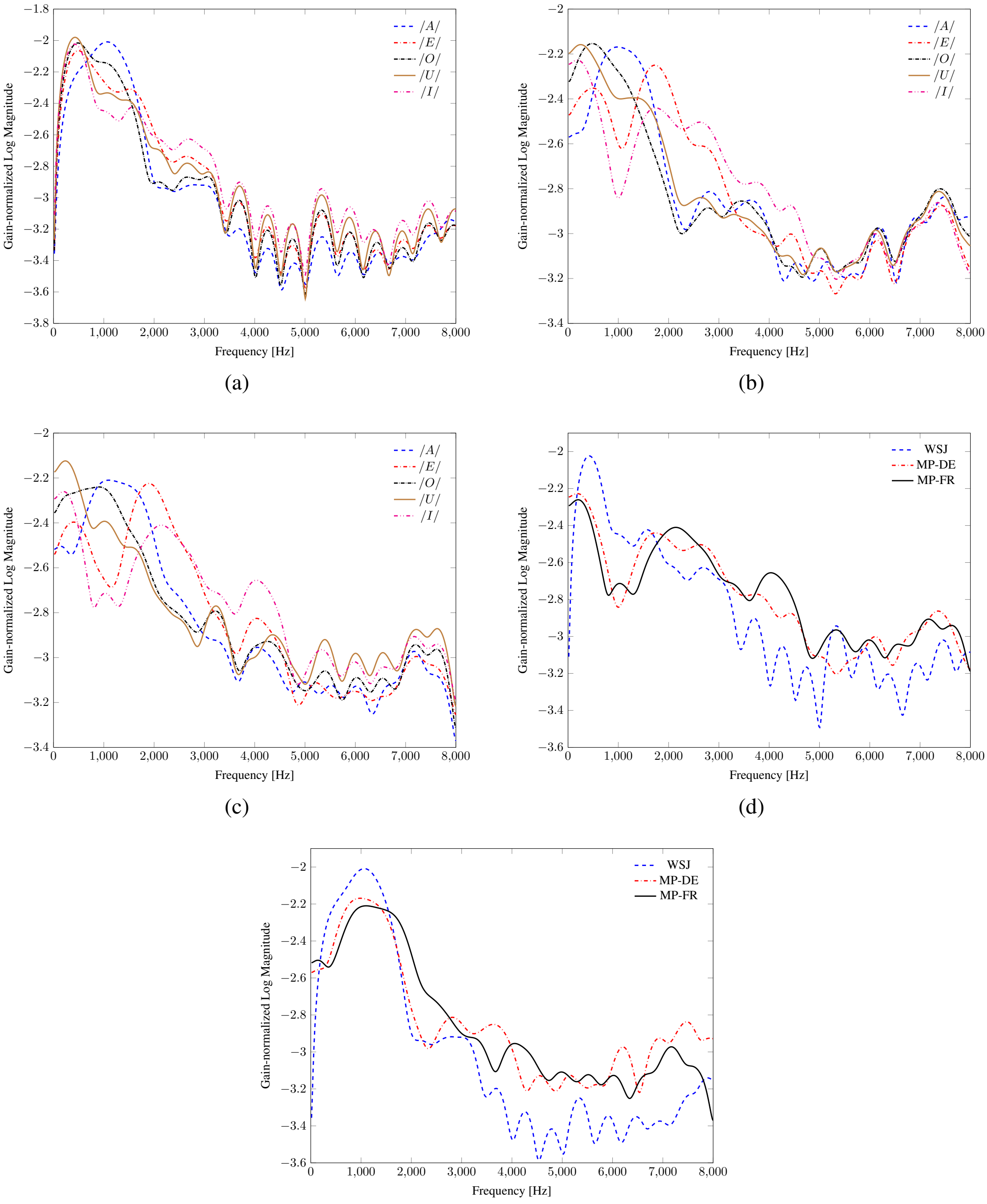
(a) First, poor frequency resolution. The filters are operating on sub-segmental speech of about 1.8ms. This leads to poor frequency resolution. It can be also noticed from the ripples in the magnitude spectrums (especially in the high frequency region);

(b) Second, the formant frequencies and their bandwidths for males and females are different. The frequency responses here are result of averaging over several male and female speakers in the respective validation data set; and

(c) Third, the analysis here has been carried on validation data, not on actual training data. So there can be spurious information present due to unseen condition or variation.

For instance, in the case of /A/, see Figure 11(e), we observe a prominent peak at around 1000 Hz, which could be seen as merger of first formant and second formant as a consequence of window effect and averaging over male and female speakers. Taking these aspects into account, we examined the frequency responses in the case of WSJ (Figure 11(a)). We found that the prominent spectral peak locations tend to relate well to the first formant, second formant and third formant information provided for English vowels in [55, p. 233]. When comparing across the languages (Figure 11(d) and Figure 11(e)) we observe a trend similar to the cumulative response of the filters (Figure 9). Specifically, the spectral peak locations and spectral balance match well for MP-DE and MP-FR. However, in the case of WSJ the spectral peak locations tend to match but the spectral balance is different than MP-DE and MP-FR.

Given the understanding gained by the first convolution layer analysis and the CNN architecture, it can be hypothesized that the second convolution layer model the modulation of the first layer filter outputs.



(a)

(b)

(c)

(d)

(e)

Figure 11: Mean power spectrum (a) for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$ estimated by CNN-1H-mono WSJ; (b) for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$ estimated by CNN-1H-mono MP-DE; (c) for phonemes $/E/$, $/A/$, $/O/$, $/I/$ and $/U/$ estimated by CNN-1H-mono MP-FR; (d) for phoneme /I/ in WSJ, MP-DE and MP-FR; and (e) for phoneme $/A/$ in WSJ, MP-DE and MP-FR. The phonemes are notated in the SAMPA format.

6.2 Intermediate feature level analysis

In this section, we focus on the analysis of intermediate feature representations that are being learned at the output of the feature learning stage. In that regard, Section 6.2.1 focuses on the discriminative aspects of the learned feature representations. Section 6.2.2 then focuses on the cross-domain and cross-lingual aspects.

6.2.1 Discriminative features

In the recognition studies presented earlier in Section 5, it was observed that CNN-1H system with much fewer parameters outperforms ANN-3H system on all the tasks. Furthermore, we also observed that the complexity of the proposed CNN-based system lies more at the classifier stage. Given that the intermediate feature representations are learned in the process of training $P(i|\mathbf{s}_t^c)$ estimator, it can be presumed that these features are more discriminative compared to cepstral-based feature representations, and thus needs less parameters at the classifier stage. To fully ascertain that aspect we conducted an experiment to compare the cepstral features and the intermediate feature representations learned by the CNN. Specifically, we trained and tested three single layer perceptron (SLP) based systems on WSJ task. One with the MFCCs with temporal context (39 × 9) as input and the others with intermediate features learned by CNN-1H and CNN-3H. In the case of CNN-3H, w_{in} was kept same as CNN-1H i.e. 210 ms. Table 8 presents the performances of the three systems. We can observe that the learned features lead to a better system than the cepstral features. Thus, indicating that the learned features are indeed more discriminative than the cepstral feature representation. Furthermore, it is interesting to note that the features learned by CNN-1H and CNN-3H yield similar systems. It suggests that the gain in ASR performance for WSJ task using CNN-3H is largely due to more hidden layers

Table 8: Single layer perceptron based system results on the Nov'92 testset of the WSJ task.

Features	Dimension	WER (in %)
MFCC	351	10.6
CNN-1H	540	7.9
CNN-3H	540	7.9

6.2.2 Cross-domain and cross-lingual studies

Conventional cepstral-based features, like MFCC, are known to be independent of the language or the domain, which is one of the main reason they become "standard" features. In the proposed system, the features are learned in a data-driven manner, thus they may have some level of dependencies on the data. In order to ascertain to what level the learned features are domain or language independent, we conducted cross-domain and cross-lingual experiments. More precisely, as illustrated in Figure 12, in these experiments the filter stage is first trained on one domain or language. It is then used as feature extractor to train the classifier stage of another domain or language.

We used the TIMIT task and WSJ task for cross-domain experiments. We investigated

1. the use of feature stage of CNN-1H of WSJ task as feature extractor for TIMIT task. The classifier stage with single hidden layer was trained on TIMIT to classify 183 phone classes.
2. the use of feature stage of CNN-1H of TIMIT task as feature extractor for WSJ task. The classifier stage with single hidden layer was trained to classify 2776 clustered context-dependent units.

In both the studies, we set the number of hidden nodes to 1000, similar to the systems reported in Section 5. The results of the two studies are presented in Table 9. In the case of TIMIT task the results are presented in terms of PER, and in the case of WSJ task in terms of WER. In the TIMIT task, we can observe that, despite the feature stage being trained to classify clustered context dependent units on much larger corpus, the PER is inferior to the case where the feature stage is learned on TIMIT. In the case of WSJ task, we observe that with feature stage trained on TIMIT the WER is slighty superior.

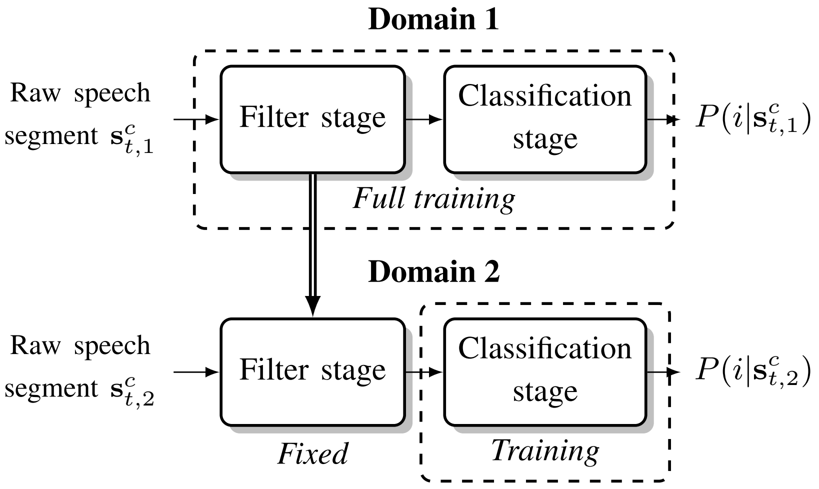

Domain 1
Raw speech segment $\mathbf{s}^c_{t,1}$
Filter stage
Classification stage
$P(i|\mathbf{s}^c_{t,1})$
Full training
Domain 2
Raw speech segment $\mathbf{s}^c_{t,2}$
Filter stage
Classification stage
$P(i|\mathbf{s}^c_{t,2})$
Fixed
Training

Figure 12: Illustration of the cross-domain experiment. The filter stage is trained on domain 1, then used as feature extractor on domain 2.

Table 9: Cross-domain results on English. The TIMIT results are in terms of PER. The WSJ task results are in terms of WER.

Classifier stage (Domain 2)	Feature stage (Domain 1)	Error Rate (in %)
TIMIT	Learned on TIMIT	22.8
	Learned on WSJ	23.3
WSJ	Learned on WSJ	6.7
	Learned on TIMIT	7.8

In addition to the fact that TIMIT and WSJ are two different corpora, there are two other differences which could have had influence. First, WSJ is a much larger corpus than TIMIT in terms of data. Second, in TIMIT CNN-1H the feature stage is learned by classifying context-independent phones, while in WSJ CNN-1H the feature stage is learned by classifying clustered context-dependent units. So, we conducted a study on WSJ task to understand the influence of the type of units at the output of the CNN on the feature stage learning, while negating the data effect. More precisely, we used the feature stage of WSJ CNN-1H-mono (presented earlier in Section 6.1.3) as feature extractor and trained the classifier stage to classify 2776 clustered context-dependent units. This system leads to a performance of 7.3% WER, which is inferior to 6.7% WER. This shows that indeed the type of units in the output of CNN has an influence on the feature learning stage. When compared to the case where the feature stage is learned on TIMIT, this result indicates that the performance gap is combined effect of the difference between the WSJ and TIMIT data sets and the units used at the output of the CNN learn the features. Finally, it is worth observing that TIMIT is a very small corpus compared to WSJ (3 hours vs 88 hours). However, the performance difference is not drastic, which suggests that the relevant features can be learned on relatively small amount of data.

We investigated the cross-lingual aspects on WSJ, MP-DE and MP-FR tasks. We conducted studies where the feature stage is learned on one language and the classifier stage is learned on the other language. For these studies, we used the feature stages of WSJ CNN-1H, MP-DE CNN-1H and MP-FR CNN-1H systems presented in Section 5. The classifier stage in all the studies consisted of a single hidden layer with 1000 nodes. The classes at the output of classifier stage remained same as before, i.e. 2776 cCD units for WSJ task, 1101 cCD units for MP-DE task and 1084 cCD units for MP-FR task. Table 10 presents the results of the study.

Before we analyze the results in detail, we can consider broader aspects. Specifically, in terms of family of languages, English and German belong to Germanic language family while French belongs to Romance language family. Given that, it can be expected that the feature stage learned

Table 10: Crosslingual studies result on English, German and French. The feature stage is learned on Domain 1 and the classifier stage is learned on Domain 2.

Classifier stage (Domain 2)	Feature stage (Domain 1)	WER (in %)
WSJ	Learned on WSJ	6.7
	Learned on MP-DE	12.1
	Learned on MP-FR	12.8
MP-DE	Learned on MP-DE	24.4
	Learned on MP-FR	26.1
	Learned on WSJ	30.9
MP-FR	Learned on MP-FR	25.9
	Learned on MP-DE	26.8
	Learned on WSJ	31.7

on MP-DE to suit well for WSJ task when compared to feature stage learned on MP-FR and vice versa. In the case of WSJ task this trend is observed (12.1% vs. 12.8%). However, it is not observed in the case of MP-DE task (30.9% vs. 26.1%). In general we observe that feature stage learned on another language leads to inferior system. The performance gap is drastic when the feature stage is learned on WSJ and the classifier stage is learned on Medialparl (MP-DE or MP-FR) and vice versa. In addition to language differences, this can be attributed to the other differences in WSJ corpus and Medialparl corpus. More precisely, WSJ corpus contains read speech collected in controlled environment while Mediaparl contains spontaneous speech collected in real world conditions. This is also supported by the findings of the analysis presented in Section 6.1.2. Since MP-DE and MP-FR are similar kind of data except for the language, the drop in performance is small (24.4% to 26.1% in the case of MP-DE task and 25.9% to 26.8% in the case of MP-FR task). Languages typically have different phone sets and this difference gets further enhanced when modeling context-dependent phones. As we saw earlier in the cross-domain studies the choice of output units influences the feature stage. So, the small drop in performance in this case can be more attributed to the phonetic level differences between German language and French language.

7 Discussion and Conclusions

Motivated from recent advances in deep learning, the present paper investigated a novel CNN-based acoustic modeling approach that automatically learns relevant representations from the speech signal and estimates phone class conditional probabilities for ASR. In this approach, the acoustic model consists of a feature stage and a classifier stage which are jointly learned during training. Specifically, the input to the acoustic model is raw speech signal, which is processed by several convolution layers (feature stage) and classified by an MLP (classifier stage) to estimate phone class conditional probabilities. We evaluated the approach against the conventional acoustic modeling approach, which consists of independent steps: short-term spectral based feature extraction and classifier training. Phone recognition studies on English and ASR studies on multiple languages (English, French, German) showed that the proposed acoustic modeling approach can yield better recognition systems.

To gain further insight, we performed analysis that largely focused on the filter stage of the approach. The key findings of the analysis are the following:

1. Both the conventional acoustic modeling approach and the proposed approach tend to model spectral information present in time span of about 200 ms for phone classification. However, they differ in the manner analysis is performed over that time span and feature representations are obtained. Indeed in the proposed approach, contrary to the conventional wisdom of short-term processing, the signal is processed at sub-segmental level (speech signal of about 2 ms) by the first convolution layer. The subsequent convolution layers temporally filter and integrate the output of first convolution layer to yield an intermediate representation. In other words, as illustrated in Figure 7, the intermediate representation is obtained by processing the information at multiple temporal resolutions.

2. The filters in the first convolution layer learn from the sub-segmental speech a spectral dictionary that discriminate phones. Specifically, this dictionary was found to model formant-like information in the spectral envelop of the sub-segmental speech. These findings are particularly interesting. First, it validates the notion of formants and phone discrimination in a data-driven manner, i.e. without making an explicit assumption about speech production model. Secondly, sub-segmental spectral processing means high time resolution and low frequency resolution. Conventional method of short-term processing (i.e. determination of the window size) has been developed considering the trade-off between time resolution and frequency resolution. Our investigations show that loss of frequency resolution due to sub segmental speech processing is not affecting the ASR performance.
3. The intermediate feature representations learned at the output of the convolution stage are more discriminative than standard cepstral-based features. This reaffirms the point that learning the features and the classifiers jointly leads to more optimal systems when compared to conventional "divide and conquer" approach.
4. The intermediate feature representations learned have some level of invariance across domains and languages. More specifically, in our analysis we observed that the variation of the learned features seems to come more from the domain characteristics as opposed to the set of subword units from the languages. This suggests that learning features in data-driven manner, as done using the proposed approach, could lead to language-independent features. This needs to be further investigated.

More recently, there are other works, similar to ours, that have investigated modeling of raw speech signal directly using ANNs [10, 56, 57]. In [10], use of DNNs (or fully connected MLP) was investigated. It was found that such an acoustic model yields inferior system when compared to standard acoustic modeling. In a subsequent follow up work [56], it was found that addition of convolution layers at the input helps in improving the system performance and reducing the performance gap w.r.t standard acoustic modeling technique. In [57], an approach was proposed using convolutional long short-term memory deep neural network (CLDNN), where the input to CLDNN is raw speech signal. This approach was found to yield performance comparable to the case where the input to CLDNN is log filter bank energies. In comparison to these works, our work mainly differs at the feature stage or convolution layers. Specifically, in these works the short-term window size is set to about 16ms based on prior knowledge, while in our case it is a hyper-parameter and was determined to be around 2ms. Furthermore, in these works the filters learned at the first convolution layer were found to be similar to auditory filter-banks. In [57], these filters were close to Mel filter banks, while in [56] the filters were found to be close to spectro-temporal filters such as MRASTA filters [58] and Gabor filters [59]. In our case, the filters are a spectral dictionary or a set of matched filters that model in-parts formant-like information in the sub-segmental speech. As a whole, these works, similar to ours, show that the relevant features from the speech signal can be automatically learned along with the classifier to estimate $P(i|\mathbf{s}_t^c)$ and ASR systems can be effectively developed.

The proposed approach paves path for further research and development. We enumerate and discuss them briefly below.

1. noise robustness: as relevant features and classifier are automatically learned, a question that arises is: whether such an approach is robust in noisy conditions? In the analysis part, we have seen that the first convolution layer models envelop of sub-segmental speech signal spectrum. In particular formant-like information, which can be considered as high signal-to-noise ratio regions in the spectrum. Furthermore, subsequent processing through max pooling could be seen as filtering of spurious temporal information present each filter output, while the second convolution layer filters could be interpreted along the lines of modeling envelop modulations in piecewise manner and combining them. Thus, the proposed approach could be expected to be robust. A preliminary investigation reported in [13] indeed indicates that.
2. rapid adaptation of acoustic model: we have observed that the feature stage has considerably fewer parameters than the classifier stage. This provides new means to adapt the acoustic model. Specifically, one of the main challenge often faced in adapting the acoustic model to new domains is the amount of adaptation data available. The data may not be sufficient to effectively adapt all the parameters in the acoustic model. In the proposed approach, this challenge could be addressed by only adapting the feature stage. Such an

approach would be analogous to maximum likelihood linear regression (MLLR) [60] adaptation approach where MLLR is used to transform the features as opposed to the models (i.e. means and variances of the Gaussians). However, in comparison to that, adaptation in the proposed framework would present two distinctive advantages. First, the adaptation would by default be discriminative, i.e. learned by improving discrimination between the phone classes. Second, upon availability of more adaptation data both the feature stage and classifier stage can be adapted in a straight-forward manner.

3. End-to-end sequence prediction: in this article, we focused on an acoustic modeling approach where time local information $P(i|\mathbf{s}_t^c)$ is estimated in end-to-end manner. A natural step forward to that is end-to-end sequence prediction where the input is raw speech signal and the output is sequence of phones or words. Alternately, the parameters of the sequence prediction model, i.e. decoder are also jointly learned. In our recent works, we have shown that the proposed approach can be extended using conditional random fields to perform end-to-end phoneme sequence recognition [61]. However, performing full fledged speech recognition through end-to-end sequence prediction is not trivial. One of the main reason being that to search effectively and efficiently the word hypothesis the relationship between words need to be learned or modeled. As evident from the present state-of-the-art HMM-based approach, the textual data that is needed to learn the relationship between words is very different than the textual data contained in the acoustic model training data. So, joint optimization of the acoustic model and the decoder in end-to-end manner from scratch using a common data set is a highly challenging problem, and is an up-and-coming research direction [62, 63, 51].

Acknowledgment

This work was supported by the HASLER foundation (www.haslerstiftung.com) through the grant "Universal Spoken Term Detection with Deep Learning" (DeepSTD). The authors would like to thank their colleagues Dr. Ramya Rasipuram and Marzieh Razavi for their help with setting up Wall Street Journal and Mediaparl studies. The authors would also like to thank Dr. Michael Liebling for his critical inputs regarding the analysis of the first convolution layer in Section 6.1.3.

References

[1] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice-Hall, Inc., 1993.

[2] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach.* Springer, 1994, vol. 247.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2012, pp. 1106–1114.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. N. Sainath, "Deep neural networks for acoustic modeling in speech recognition: the Shared Views of Four Research Groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, p. 8297, 2012.

[8] N. Jaitly and G. Hinton, "Learning a better representation of speech soundwaves using restricted Boltzmann machines," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5884–5887.

[9] D. Palaz, R. Collobert, and M. Magimai.-Doss, "Estimating Phoneme Class Conditional Probabilities from Raw Speech Signal using Convolutional Neural Networks," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, September 2013.

[10] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Singapore, Sep. 2014, pp. 890–894.

[11] Y. LeCun, “Generalization and Network Design Strategies,” in *Connectionism in Perspective*, R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels, Eds. Zurich, Switzerland: Elsevier, 1989.

[12] D. Palaz, M. Magimai.-Doss, and R. Collobert, “Convolutional Neural Networks-based Continuous Speech Recognition using Raw Speech Signal,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015.

[13] D. Palaz, M. Magimai-Doss, and R. Collobert, “Analysis of CNN-based Speech Recognition System using Raw Speech as Input,” in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[14] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 3, pp. 328 –339, mar 1989.

[15] T. Robinson, “An Application of Recurrent Nets to Phone Probability Estimation,” *IEEE Transactions on Neural Networks*, vol. 5, pp. 298–305, 1994.

[16] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 6645–6649.

[17] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*. Prentice Hall, 1978.

[18] M. R. Schroeder and B. S. Atal, “Code-excited linear prediction (CELP): High-quality speech at very low bit rates,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP’85.*, vol. 10. IEEE, 1985, pp. 937–940.

[19] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[20] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *The Journal of the Acoustical Society of America*, vol. 87, p. 1738, 1990.

[21] S. Furui, “Speaker-independent isolated word recognition based on emphasized spectral dynamics,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 11. IEEE, 1986, pp. 1991–1994.

[22] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, “Deep convolutional neural networks for LVCSR,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8614–8618.

[23] P. Swietojanski, A. Ghoshal, and S. Renals, “Convolutional Neural Networks for Distant Speech Recognition,” *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, September 2014.

[24] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[25] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[26] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks,” in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 437–440.

[27] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, p. 3042, 2012.

[28] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4277–4280.

[29] A. Mohamed, G. Dahl, and G. Hinton, “Acoustic Modeling Using Deep Belief Networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14 –22, jan. 2012.

[30] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1096–1104.

[31] A. Poritz, “Linear predictive hidden Markov models and the speech signal,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 7, May 1982, pp. 1291–1294.

[32] Y. Ephraim and W. J. J. Roberts, “Revisiting autoregressive hidden markov modeling of speech signals,” *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 166–169, Feb. 2005.

[33] B. Mesot and D. Barber, "Switching Linear Dynamical Systems for Noise Robust Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1850–1858, Aug. 2008.

[34] H. Sheikhzadeh and L. Deng, "Waveform-based speech recognition using hidden filter models: Parameter selection and sensitivity to power normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, p. 8089, 1994.

[35] J. Yousafzai, Z. Cvetkovic, and P. Sollich, "Tuning support vector machines for robust phoneme classification with acoustic waveforms," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2009, pp. 2391–2394.

[36] J. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neuro-computing: Algorithms, Architectures and Applications*, 1990, pp. 227–236.

[37] L. Bottou, "Stochastic Gradient Learning in Neural Networks," in *Proceedings of Neuro-Nmes 91*. Nimes, France: EC2, 1991.

[38] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, 1993.

[39] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[40] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[41] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91, 1992, pp. 357–362.

[42] P. Woodland, J. Odell, V. Valtchev, and S. Young, "Large vocabulary continuous speech recognition using HTK," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. ii, apr 1994, pp. II/125 –II/128 vol.2.

[43] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The HTK book," *Cambridge University Engineering Department*, vol. 3, 2002.

[44] D. Imseng, H. Bourlard, H. Caesar, P. N. Garner, G. Lecorv, A. Nanchen, and others, "MediaParl: Bilingual mixed language accented speech database." in *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 263–268.

[45] M. Razavi, R. Rasipuram, and M. Magimai.-Doss, "On Modeling Context-Dependent Clustered States: Comparing HMM/GMM, Hybrid HMM/ANN and KL-HMM Approaches," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[46] M. Razavi and M. Magimai.-Doss, "On Recognition of Non-Native Speech Using Probabilistic Lexical Model," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[47] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A Matlab-like Environment for Machine Learning," in *BigLearn, NIPS Workshop*, 2011.

[48] S. P. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ, "Improved feature processing for deep neural networks." in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2013, pp. 109–113.

[49] Y. Hifny and S. Renals, "Speech recognition using augmented conditional random fields," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 354–365, 2009.

[50] O. Abdel-Hamid, L. Deng, D. Yu, and H. Jiang, "Deep segmental neural networks for speech recognition." in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2013, pp. 1849–1853.

[51] L. Lu, L. Kong, C. Dyer, N. A. Smith, and S. Renals, "Segmental Recurrent Neural Networks for End-to-end Speech Recognition," *arXiv preprint arXiv:1603.00223*, 2016.

[52] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[53] H. Hermansky, "Should recognizers have ears?" *Speech communication*, vol. 25, no. 1, pp. 3–27, 1998.

[54] F. Hönig, G. Stemmer, C. Hacker, and F. Brugnara, "Revising perceptual linear prediction (plp)." in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2005, pp. 2997–3000.

[55] L. Deng and D. O'Shaughnessy, *Speech processing: a dynamic and optimization-oriented approach*. CRC Press, 2003.

[56] P. Golik, Z. Tüske, R. Schlüter, and H. Ney, "Convolutional Neural Networks for Acoustic Modeling of Raw Time Signal in LVCSR," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015, pp. 26–30.

[57] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the Speech Front-end With Raw Waveform CLDNNs," *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[58] H. Hermansky and P. Fousek, "Multi-resolution RASTA filtering for TANDEM-based ASR," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2005.

[59] S. Chang and N. Morgan, "Robust CNN-based speech recognition with Gabor filter kernels." in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014, pp. 905–909.

[60] M. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249–264, 1996.

[61] D. Palaz, R. Collobert, and M. Magimai. -Doss, "End-to-end Phoneme Sequence Recognition using Convolutional Neural Networks," *NIPS Deep Learning Workshop*, December 2013.

[62] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2014, pp. 1764–1772.

[63] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. C. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," *arXiv preprint arXiv:1512.02595*, 2015.