Research Group of Artificial Intelligence
University of Szeged

Arenberg Doctoral School
Faculty of Engineering Science
KU Leuven

Noise Robust Automatic Speech Recognition Based on Spectro-Temporal Techniques

Summary of the PhD Dissertation

by

György Kovács

Supervisors:

Dr. László Tóth
Prof. dr. ir. Dirk Van Compernolle

Szeged
2017

1 Introduction

Automatic Speech Recognition (ASR) is the automatic transcription of speech into a sequence of words or subword units (e.g. phones) by a machine. This seemingly simple task has many uses in dictation software, dialogue systems, and personal assistants. The utility of these applications depends on many factors, among these are considerations such as speed, requirements for computation power and memory capacity. The requirement our study is focusing on however is accuracy, even in adverse environments like the presence of background noise, and mismatched transfer channel characteristics. What makes meeting this requirement especially challenging, and hence of special interest is that often it is not possible to foresee every possible problem. Hence ASR systems have to be trained in such a way that they should be able to work in the kind of environments they had not been specifically prepared for.

Here, this problem is examined in the HMM/ANN framework, the overview of which is depicted in Figure 1. This model differs from the traditional HMM/GMM approach in that the generative Gaussian Mixture Model (GMM) is replaced by a discriminative neural network model. This, however, is not a new development, as the HMM/ANN hybrid approach has been around for decades [33], but its use is nowadays quite widespread thanks to the introduction of DNNs. Where we deviate from this practice is that – following Kleinschmidt [17] – we separate the feature extraction phase into primary- and secondary feature extraction phases. The reason for separating them into two is that in the primary feature extraction phase our study was confined to using standard well-established methods, while here we focused on the secondary extraction phase. Along with the acoustical model, this was the part we concentrated our efforts on, either by proposing modifications to feature extraction, or by combining the phase of secondary feature extraction with the training of the acoustic model.

Our efforts here were aimed at increasing the accuracy of speech recognition, especially in noisy environments. One approach to attain the goal of robustness is by exploiting our knowledge on Human Speech Recognition (HSR) in our ASR systems. The rationale behind this proposal was that by studying the system that is working more effectively (namely HSR [41]), we should be able to improve a system that is less effective (namely ASR). To a degree we have seen the success of this approach in the application of transformations like the Mel-scale and the Bark-scale, and also in spectro-temporal processing, on which the methods proposed here were based.

The methods proposed here will be evaluated using three tasks namely, phone classification, phone recognition and word (continuous speech) recognition. For this, we will use the "Szeged" Hungarian Broadcast news database [46], as well as the TIMIT [28] speech corpus of English speech, and the Aurora-4 database of English speech [36].

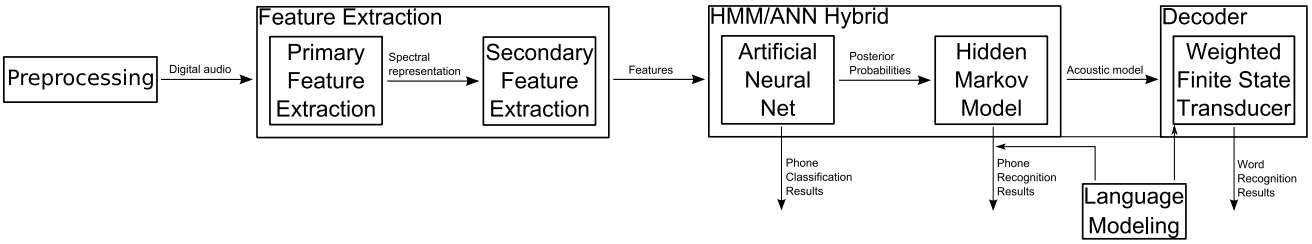

Figure 1: Overview of the speech recognition process.

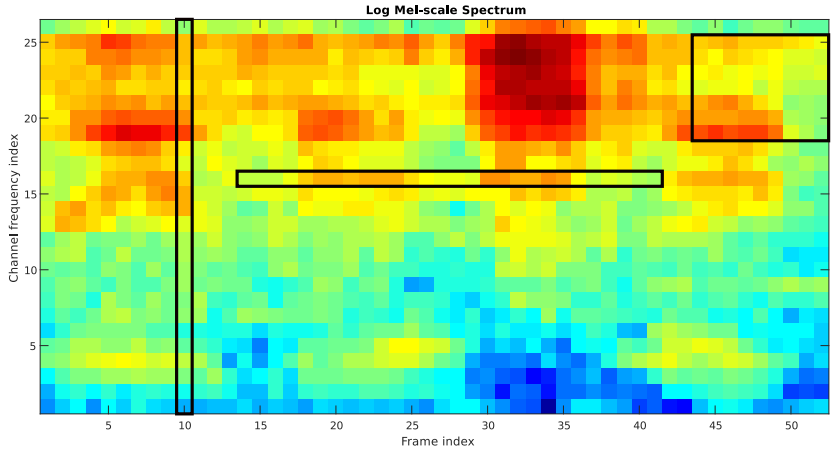

Figure 2: Log mel-spectral representation of an excerpt from the Aurora-4 database. The black boxes (left-to-right) shows the shape of the feature extraction patches used by a) the classic MFCCs, b) the TRAP features, and c) localised spectro-temporal features.

2 Spectro-Temporal Feature Extraction

More and more is known about human speech processing, but traditional feature extraction methods take into account only its most fundamental properties. Although it is not necessary that methods which seek inspiration from HSR should outperform purely mathematical algorithms, it seems reasonable to expect better behaviour from the methods that approximate the properties of human hearing more closely. One such property that speech processing may benefit from approximating is the joint spectro-temporal sensitivity of the receptive fields of cortical cells [6].

Although humans can barely recognise such short excerpts, MFCC processes the speech signal in 20-30 millisecond chunks. This context can be expanded by using Δ coefficients and neighbouring frames, but it is still different from having features tuned to specific temporal modulations [4]. Moreover, with the increased number of neighbouring frames used, the number of features also grow, and hence the application of dimension reduction methods may be necessary.

Another problem with using MFCC vectors is that the resulting features are still global along the frequency axis, which means that even in the presence of band-limited noise, all features will be contaminated [4]. Furthermore, there are experiments which indicate that "human speech perception is based on relatively narrow frequency channels" [32]. This suggests that the windows should be localised in frequency. The same considerations motivated the introduction of the TRAP model [12], where each frequency band is processed separately. In Figure 2, we portray this approach along with the conventional approach of MFCC. In contrast with these methods, in this study we will focus on spectro-temporal processing, where the spectral representation is processed in patches that are localised in both time and frequency (see Figure 2(c)).

Let us now formalise the general process of spectro-temporal processing. We can interpret it as applying a filter F on a patch P, to obtain an output o defined by the following formula:

$$o = \sum_{f=0}^{M-1} \sum_{t=0}^{N-1} P(f,t)F(f,t), \tag{1}$$

where M and N are the respective height and width of patch P and filter F. One can readily obtain a set of features by using several filters with different coefficients, and/or applying the same filters at different positions in the time-frequency plane. A more difficult task is to design a proper family of filters that are optimal for the given task.

2.1 2D DCT

An obvious generalisation of the MFCC feature extraction process to localised spectro-temporal windows is to replace the Discrete Cosine Transform (DCT) with its two-dimensional counterpart (2D DCT). The resulting feature extraction process can be regarded as performing the filtering defined by Eq. (1) with the following filter coefficients:

$$F_{pq}(f,t) = \cos\frac{\pi\cdot(2f+1)\cdot p}{2M}\cos\frac{\pi\cdot(2t+1)\cdot q}{2N}, \tag{2}$$

where M and N are the respective height and width of the filters for f and t, while p and q specify the modulation frequencies of the filter along the frequency and time axes.

By definition, for a patch of size $M \times N$, the 2D DCT returns the same number of coefficients. There are several findings, however (both in speech recognition and image processing), which indicate that these coefficients are not equally important for representing the underlying content, and that it may be sufficient to retain just some low-order coefficients [4, 16]. It is an open question, however, of how many of these coefficients should be retained. It also had to be experimentally determined how big our patches should be, to what degree they should overlap, and how many mel filters should be used in the mel-scale spectrum these patches operate on (if indeed they operate on the mel-scale spectrum, and not on the spectrogram).

To find an answer to these questions, we conducted several phone classification experiments on the TIMIT database. Based on the results of these experiments, we selected three candidate set of parameter values to be examined more closely. Ultimately, we made a decision to use the log mel-spectrum created with 26 filter channels as our input (and this decision was partially inspired by a better comparability with MFCC results). Furthermore, we decided that the patches applied on this input would have a height of 7 mel-channels, and a width of 9 frames, and they would be extracted at 12 positions in the frequency domain. Lastly, it was decided that from each of these patches 9 2D DCT coefficients were to be retained.

We evaluated the utility of the chosen settings on the TIMIT database using the clean core test set as well as its noise-contaminated versions. Comparing the results obtained using the above settings with those obtained using MFCCs (see Table 1), we found that using spectro-temporal processing we get lower phone error rates in noisy conditions, regardless of the level of noise, while in clean conditions our results are very similar to those we get using MFCCs.

We also presented a method for the combination of the two sets of features. Here, a neural net is trained on each feature set separately, then the posteriors of these nets are concatenated and used as features in a third neural net. We showed that by combining the two feature sets in this manner, we can further improve the performance in the case of clean speech.

Feature set	Clean speech	Pink noise			Babble noise		
		20 dB	10 dB	0 dB	20 dB	10 dB	0 dB
MFCC	29.11%	56.78%	74.78%	85.57%	48.04%	73.56%	86.29%
2D DCT	29.52%	46.62%	67.01%	79.07%	41.03%	58.36%	74.81%

Table 1: Phone error rates (PER) got on the clean and noise-contaminated test sets of the TIMIT corpus.

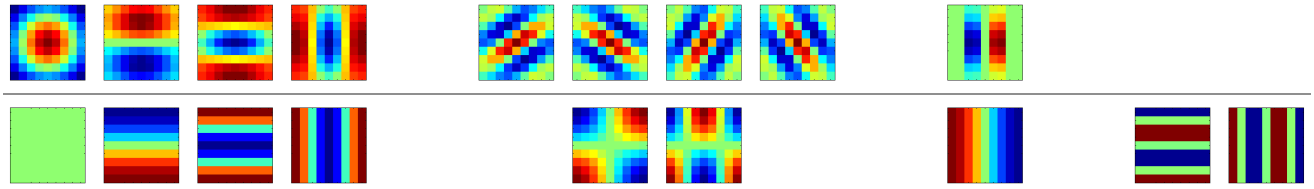

Figure 3: Manual set of Gabor filters (first row) and 2D DCT (second row) filters, with the corresponding filters vertically aligned to emphasise the similarities. The patch size is 9×9 [26].

2.2 Gabor filters

Another method for extracting spectro-temporal features is to apply the real part of Gabor filters on our patches, defined as the product of a two-dimensional Gaussian, namely

$$W(f,t) = \frac{1}{2\pi\sigma_f\sigma_t} e^{-\frac{1}{2}\left(\frac{(f-f_0)^2}{\sigma_f^2}+\frac{(t-t_0)^2}{\sigma_t^2}\right)}, \tag{3}$$

and the real part of an oriented sinusoid,

$$S_{\Omega,\omega}(f,t) = cos\left(\frac{\pi \cdot f \cdot 2\Omega}{M} + \frac{\pi \cdot t \cdot 2\omega}{N}\right). \tag{4}$$

Here, f and t iterate over the frequency and time span of windows. Furthermore, f_0 and t_0 define the centre of the Gaussian in the frequency and time domains respectively, while σ_f^2 and σ_t^2 define their bandwidth in the respective domain. Lastly, M and N specify the transform size, while Ω and ω specify the slanting and the periodicity of the sinusoid.

Although it is clear how the parameter values of Ω and ω shape the resulting Gabor filters, it is not so obvious how these parameters should be selected. It is also not clear how many filters should be used, and what size these filters should have for optimal speech recognition performance. Much effort has been devoted to this question over the years [8, 18, 42], and for this reason we also examined this question in detail. First, we described two automatic feature selection methods for the task. Namely, the Feature Finding Neural Network (FFNN), used by Kleinschmidt and Gelbart in creating their Gabor filter set [18], and the Sequential Forward Floating Selection (SFFS), introduced by Pudil et al. [38]. We also created two filter sets using these methods. Later, based on experimental results on the TIMIT corpus and the "Szeged" Hungarian Broadcast news database, we compared the following Gabor filter sets:

- The Gabor filter sets introduced by Kleinschmidt and Gelbart [18] ($G1$, $G2$, $G3$)
- The Gabor filter set introduced by Schädler et al [42] (SMK set)
- A Gabor filter set we created using the FFNN algorithm (FFNN set)
- A Gabor filter set we created using the SFFS algorithm (SFFS set)
- A Gabor filter set we created manually using some simple heuristics, such as the similarity between Gabor filters and 2D DCT coefficients (Manual set – see Figure 3)
- 10 Gabor filter sets created in a random manner.

On the TIMIT database we found that the Manual set outperformed all other Gabor filter sets. This includes those filter sets that had been created using some sophisticated automatic feature selection method. The same was true in the case of the clean and noise contaminated speech. Furthermore, we found that in many cases filter sets created in a random fashion perform similarly or even better than their carefully crafted counterparts. And similar to what we saw in the case of 2D DCT, we found that using a proper set of spectro-temporal features we can attain a better performance than that with the MFCCs, with clean speech as well as in a noise contaminated environment.

Regarding the relative performance of various Gabor filters, the results obtained on the "Szeged" Hungarian Broadcast news database were quite similar. For one, the Manual filter set provided lower error rates than any other Gabor filter set examined. This point clearly shows that the heuristics applied in the manual selection process are more general than the heuristics of the database-driven feature selection methods. When comparing the results got using various filter selection algorithms with each other, we found that the feature set created by SFFS significantly outperformed its FFNN-created counterpart. As we observed a similar tendency with TIMIT, SFFS here seems to be a better selection algorithm than FFNN. However, the resulting filter set still performs significantly worse than MFCCs (and the *Manual set*). The same could be said about the *G1-3* filter sets created by Kleinschmidt and Gelbart, and the SMK filter set created by Schädler et al. So unfortunately our hope that the filter optimisation would not have to be repeated for each training database did not materialise. Also, we can see that the best randomly selected filter set gave scores that are almost as good as those of the *SFFS set*, and are better than the scores obtained with the *G2, G3* or the *SMK set*. This again suggests that the filter selection algorithms actually fail to achieve their goal.

Summary of Theses

I/1. We demonstrated that 2D DCT feature extraction can be performed on the conventional critical-band log-energy representation in such a way that it leads to similar or better phone classification and phone recognition accuracy scores than those obtained using the conventional MFCCs. We also found that the advantage of the former is even more pronounced in noise contaminated speech (published in [19, 20]).

I/2. We presented a simple yet effective strategy for the combination of the conventional (MFCC) features, and the new – 2D DCT – feature set. We demonstrated on the core test set of the TIMIT corpus that this combination produces a better performance than either of the two feature sets (published in [20]).

I/3. We introduced and evaluated a new Gabor filter set. We demonstrated on the TIMIT English database and the "Szeged" Hungarian Broadcast news database that it performs better than Gabor filter sets introduced earlier, as well as filter sets created for this study, using either the SFFS or the FFNN feature selection method. On the TIMIT corpus, we also demonstrated that this new filter set performed better on both clean and noise contaminated speech in English. And in the case of Hungarian speech, its performance matched that of MFCCs the most closely (published in [26]).

3 Joint Training of Spectro-Temporal Features and Neural Nets

In the traditional approach, the extraction of a fix set of features, and the training of the adaptable classifier are quite separate. In spectro-temporal speech processing this means that the extraction of secondary features and the training of the acoustical model are traditionally done in two separate steps. While this separation is technically convenient, it might result in suboptimal features for the actual machine learning method.

It is easy to see the flaw in this approach of separate feature extraction and model training, when the feature set is entirely hand-crafted: the accuracy of recognition largely depends on the ability of our expert to design a useful feature set. Our earlier results with various Gabor filter sets highlighted the complications associated with automatically generated feature sets: the feature selection algorithms examined not only proved to be slow, but they also failed to produce feature sets that would be suitable for other data sets as well. In fact, our handcrafted filter ("Manual set") set not only gave better results in the cross-database tests, but in most cases it surpassed the feature sets created by automatic means on our original task as well. Unfortunately, the selection heuristics applied in the design of the "Manual set" do not provide a guarantee either that the resulting set is optimal.

For the above reasons, our suggestion was to combine the feature selection step and the statistical modelling step into one. This is a similar concept to a recent study where a Convolutional Neural Net (CNN) was extended in order to optimise the feature extraction filter bank [40]. The difference is that our method uses spectro-temporal filters instead of a conventional spectral filter bank. Here, we treat the feature extraction filters as the lowest layer of a neural net, which also makes it possible for the training algorithm to fine-tune the filter coefficients.

To understand how this was possible, let us examine the formula specifying the output o of a perceptron:

$$o = a\left(\sum_{i=1}^{L} x_i \cdot w_i + b\right), \tag{5}$$

where $\mathbf{x}$ is the input of the neuron, L is the length of the input, w is the weight vector, and b is a bias corresponding to that neuron. For the activation function a we usually apply the sigmoid function; but it is also possible to create a linear neuron by setting a to the identity function. If we select the identity function as the activation function of our perceptron, and set the bias b to zero, we can write (5) in the following form:

$$o = \sum_{i=1}^{L} x_i \cdot w_i. \tag{6}$$

If we represent P and F in (1) in vector form (as $\overline{P}$ and $\overline{F}$, which is just a notational change), we get the following formula:

$$o = \sum_{i=1}^{M \cdot N} \overline{F}_i \cdot \overline{P}_i. \tag{7}$$

Now it is easy to see that (6) is just a special case of (7) – and by extension that (5) is a special case of (1) – by chosing the vectorised version of the patch P ($\overline{P}$) as our input vector x, and the vectorised version of filter F ($\overline{F}$) as our weight vector w. This means that the spectro-temporal filters can indeed be integrated into an ANN classifier system as special neurons, with the filter coefficients being the weights of the given neuron.

Initial filter weights	filter weights unaltered	trained
Random	32.96%	30.27%
2D DCT	31.19%	30.21%
Gabor (Manual)	32.41%	30.29%

Table 2: Phone error rates (PER) got on the core test set of TIMIT (the average of 20 independently trained neural nets).

First, as a proof on concept, we conducted experiments on clean speech using the TIMIT speech corpus. Here, traditional sigmoid neural networks were applied where the length of the input context did not exceed the length of spectro-temporal patches applied. The results of these experiments (see Table 2) indicated that the joint optimisation scheme indeed resulted in a better recognition performance. This held true irrespective of the scheme we used for the initialision of filter coefficients. Next, we extended our investigation to bigger neural networs (having more neurons in their hidden layer) that use a wider input context. Here, we repeated our experiments using both the clean and noise contaminated test set of TIMIT, as well as using the "Szeged" Hungarian Broadcast news database. The results of these experiments confirmed our earlier findings on the benefits of joint training. We also demonstrated that the advantage of joint training is also apparent with noise contaminated speech.

What is more, the advantage of our proposed method is also clearly discernible in the cross-database and cross-language experiments. This last observation is supported in Table 3. Here, using the "Szeged" Hungarian Broadcast news database, we compare the results obtained using traditional MLPs (where the input of the MLP was the MFCC, or one of the Gabor filter sets introduced by Kleinschmidt and Gelbart [18], or a similar filter set introduced by Schädler et al. [42], or one of the Gabor filter sets we created by means of automatic methods) with those results obtained using the joint framework. From Table 3, we ascertain that the best results were obtained when the joint framework was applied and its feature coefficients were initialised based on either our handcrafted Gabor filter set (Gabor Manual), or a modified version of the very same filter set that had been first fine-tuned using the training set of the TIMIT database (Gabor Manual+TIMIT).

Joint Training	(Initial) Filter weights	Filter weights Unaltered	Trained
✓	Random	26.94%	25.06%
✓	Gabor (Manual)	26.36%	24.75%
✓	Gabor (Manual+TIMIT)	25.10%	**24.64%**
	MFCC + Δs	25.03%	–
	SFFS set + Δs	26.06%	–
	FFNN set + Δs	26.49%	–
	G1 + Δs	25.91%	–
	G2 + Δs	27.16%	–
	G3 + Δs	36.32%	–
	SMK set	26.95%	–

Table 3: Phone error rates (PER) got on the "Szeged" Hungarian speech database (the average of 10 independently trained neural nets).

Initial Filter weights	Original framework	DRN framework	DCRN framework
Gabor	26.24%	23.37%	**22.98%**
2D DCT	26.54%	24.15%	**23.22%**
Random	26.53%	23.58%	**23.25%**

Table 4: Phone error rates (PER) got on the clean core test set of TIMIT (reported error rates are the average of 10 independently trained neural nets).

After the initial success of the joint trainig framework, we extended our experiments to include experiments on deep learning. Here, following Glorot and Bengio [10], we created our Deep Neural Networks (DNNs) by replacing the sigmoid activation function in the hidden layer with the rectifier activation function. This means that we used Deep Rectifier Neural Nets (DRNs), where the output of neurons in the hidden layer can be calculated as follows:

$$o = max\left(0, \sum_{i=1}^{L} x_i \cdot w_i + b\right). \tag{8}$$

The resulting framework was again evaluated using the TIMIT speech corpus. We found that the results of these experiments corroborated our earlier findings on the utility of the proposed method. What is more, these results supported our supposition on the benefit of using DRNs instead of traditional neural nets (see Table 3).

Lastly, the neural nets applied in joint training were again modified by incorporating into it convolution in the time domain. This was carried out by processing neighbouring patches (using the same weights), in such a way that between two patches processed there were always patches left out (i.e. we increased the step size). This means the conversion of our joint framework into a Deep Convolutional Rectifier Neural Net (DCRN). Once again, the introduction of this modification significantly decreased our error rates obtained on the TIMIT phone recognition task in clean speech (see Table 4), and in most cases with noise contaminated speech.

Summary of Theses

II/1. Here, we introduced an algorithm for the joint training of spectro-temporal features and neural networks. We tested the algorithm in a phone recognition tasks on the TIMIT speech corpus and the "Szeged" Hungarian Broadcast news database. The results confirmed the viability of joint optimisation, and showed that it leads to significantly improved recognition scores as well as to an improved cross-database and cross-language performance (published in [21, 26]).

II/2. We incorporated several advances from the current research in neural networks into the joint framework. Namely, rectifier linear units (ReLUs) and convolutional neural networks (CNNs). This highlights the flexibility of the framework and its capacity to accomodate new advances. The results obtained using the TIMIT speech database demonstrate that with these modifications, the framework is capable of achieving a significantly improved performance (published in [22]).

4 The Multi-Band Processing of Speech using Spectro-Temporal Features

In the multi-stream speech processing scheme the information extracted from the speech signal is divided into separate sources. Each of these sources may represent a different aspect or property of the input, and it is processed independently up to a recombination stage. A special case of multi-stream processing is multi-band processing (first described by Duchnowski [7]), where different frequency regions are treated as separate sources. In this paradigm the input is decomposed into spectral bands, then after independent processing of the bands (which usually involves carrying out a partial recognition step), information from the different bands is recombined to produce an overall recognition result.

The use of this method was motivated by various considerations, such as signal processing, the opportunity to exploit the potential of parallel computing systems, human speech perception, and robustness (in the case of a noise that is limited to certain bands, recognisers working with features originating from those bands that were unaffected by the noise would also be unaffected – in contrast with recognisers that rely on full-band features). When outlining the motivation for spectro-temporal speech processing, our expectation of robustness and the observations in human speech perception were major factors as well. The similarity between the spectro-temporal and multi-band processing does not stop there, as their schematic representation indicates (see Figure 4). In fact, the approach we were following in Chapter 2 (where features extracted from different frequency bands of the full spectrum are concatenated into one feature vector), is sometimes referred to as a branch of multi-band processing, under the label "feature recombination" method [35]. Figure 4, however, not only shows the similarity between the two methods – both having been based on observations on human speech perception, and designed to increase the robustness of speech recognisers – it also shows their compatibility. That they are compatible is clear if we consider that we arrived at the multi-band approach by having the exact same local features as we did earlier and feeding them into band classifiers, instead of concatenating them. But while these methods have been in use for a long time now, few attempts have been made to combine them. The couple of studies that attempted to combine spectro-temporal processing with the multi-stream approach mostly concentrated on separating the streams based on the properties of the filters, rather than based on the frequency domain these filters were applied on for extracting the necessary features (i.e. multi-band approach) [31, 47].

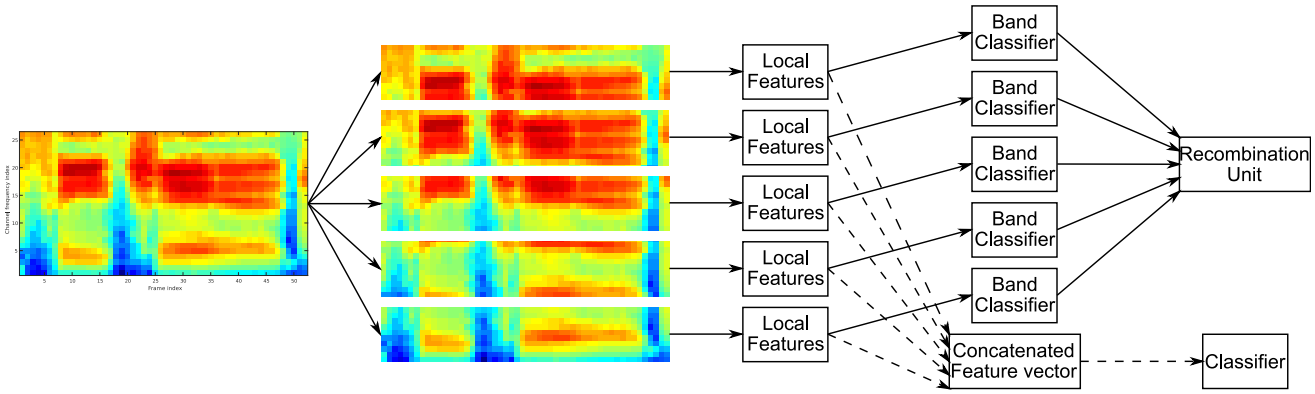

Figure 4: Schematic representation of a spectro-temporal processing schemes framework (see Chapter 2) and a typical multi-band processing framework. Where the two methods diverge, their process is represented by dashed/solid lines, respectively.

Multi-band processing is an umbrella term that describes a wide range of methods with quite different properties. The number of sub-bands used, for example, can range from two to twenty two. Another question is the overlap of the subbands. While in many cases the separation of subbands is carried out using rectangular windows with no overlap (apart from that originating from the filter-banks overlapping in the frequency domain), overlapping rectangular or smooth windows can also be used. And once the subbands have been selected, there are several decisions to be made on processing them and on combining the resulting information. Subbands may for example be processed using GMMs, and HMM/ANN hybrids. But typically this is performed via ANNs [3, 32]. Furthermore, after processing the bands, a decision has to be made about how much of the resulting information to employ (e.g. whether to use just the label of the winning hypothesis, the order of different hypotheses, or the probability predictions). Let us now assume that the recombination will be carried out based on the posterior estimates of ANNs. Several methods can still be chosen for recombination, ranging from simple fixed linear combinations to sophisticated methods trying to dynamically assess the reliability of the bands, including ANNs [32]. Here (as Figure 4 suggests), we experimented with a particular setup of this broad term. In our experiments we divided the input stream into four to six overlapping sub-bands, and processed each of these sub-bands using a neural network. In all our experiments this initial stage was followed by a recombination net, using the output of the preceding neural networks (this means either using neural net posterior probabilities or bottleneck features).

In the first phase we examined various neural net structures for the processing of spectro-temporal features (introduced in Chapter 2) in separate bands, as well as for the recombination of information produced by these neural networks. In one setup we applied traditional, fully connected neural networks in both stages, where the size of various layers was selected so that the overall parameter number matched that in the "feature recombination" method (MB small). We used a variation of this same setup with a higher number of neurons on each layer. Finally, we also created a setup where both for individual band processing and for recombination of their outputs, we used DNNs (MB deep). Here, unlike in the earlier (and later) chapters, deep learning was performed using the pre-training algorithm of Hinton et al. [13]. We utilised the TIMIT database to compare the results of the multi-band approach with those obtained using the "feature recombination" method (see Table 5). But we also compared the results of the various architectures used in the the multi-band approach with one another. We found that the multi-band approach (MB) produced better recognition rates than the "feature recombination" method (FC), irrespective of the specific neural net architecture that was used. But the best results were obtained when using the multi-band processing approach with deep learning. We also found that on this task Gabor filters gave us a better performance in each testing condition than 2D DCT coefficients did, when using the best performing deep learning setup (MB deep).

Settings	2D DCT	Gabor	No. of ANN parameters
FC	26.85%	26.78%	~ 6
MB small	26.07%	25.45%	~ 6
MB big	24.70%	24.79%	$\sim$16
MB deep	**23.47%**	**22.81%**	$\sim$17

Table 5: Phone error rates (PER) got on the clean core test set of TIMIT (reported results are the average of 10 independently trained neural nets), and the number of free parameters (in millions) for the different settings. For 2D DCT and Gabor filters, the best scores are highlighted in bold.

Processing method	Test set A	Test set B	Test set C	Test set D	Avg.
Feature recombination	3.9%	13.4%	12.0%	28.6%	19.3%
Bottleneck	3.7%	**12.9%**	**11.6%**	**26.2%**	**17.8%**
Ganapathy [9]	**3.0%**	**12.9%**	11.7%	27.7%	18.5%

Table 6: Word Error Rates (WERs) got on the test sets of the Aurora-4 corpus (reported scores are the average of 3 independently trained neural nets).

In the second phase, we applied the joint training framework (introduced in Chapter 3) on each frequency band. Here, once again, the recombination of the values from the separate bands was performed using a recombination net. In this case, however, instead of using the posterior probability estimates got from the band processing neural networks as input, the recombination net used the output of bottleneck layers that had been positioned in each band processing neural network in front of the output layer. We evaluated the resulting bottleneck multi-band scheme using the Aurora-4 speech corpus and its clean training scenario. The results of these experiments (see Table 6) are reported in four groups. Under test set A, word error rates are reported for clean speech that had been recorded using a Sennheiser close talking microphone. Error rates reported under test set B were obtained on speech that had been recorded using the same Sennheiser microphone, but it had been contaminated with one of 6 different noise types (car, babble, restaurant, street, airport, or train noise). The results achieved using clean speech recorded with various secondary microphones is reported under test set C. Lastly, under test set D, average error rates are reported that were got on noise contaminated speech, also recorded with the various secondary microphones. For the sake of comparison, Table 6 also contains a recent state-of-the-art result on the same task, taken from Ganapathy [9]. We found that the multi-band approach is also beneficial when applied in the joint framework for feature extraction and neural net training. The results of these experiments also revealed that the multi-band approach in combination with the ARMA features [9] can provide word error rates that are competitive with those of the state-of-the-art.

Summary of Theses

III/1. Utilising the inherent compatibility of multi-band speech processing and spectro-temporal feature extraction, we proposed a method for the combination of the two. We did so by applying the spectro-temporal features introduced in Chapter 2. Then using the TIMIT speech corpus we showed that the multi-band approach was advantageous in clean speech as well as speech contaminated with various types artificial and real-life noise (published in [25]).

III/2. We presented a method that combines multi-band processing with the joint training of spectro-temporal features and neural networks. The proposed method was evaluated using the clean training scenario of the Arma-4 speech corpus. It was demonstrated that this method performs better than the earlier version of the joint training framework; and in fact to the best of our knowledge its results are among the best reported for the task among methods that do not employ speaker adaptation (published in [23]).

Method	PER
Baby et al. [2]	19.6%
Plahl et al. [37]	19.1%
Tóth [44]	18.7%
Current work	*18.5%*
Graves et al. [11]	17.7%
Tóth [45]	**16.7%**

Table 7: Phone error rates (PERs) presented in the literature, on the core test set of TIMIT. The best score is highlighted in bold. The score got by us is highlighted in Italics.

5 Band dropout

In most of this study, the key question we asked following the introduction of a new method or modification was whether we were able to achieve better recognition results with the newly introduced method or modifications than we had been able to achieve beforehand. However, as our research is not carried out in a vacuum, it is also important to see how these results got using these modifications measure up to those got using similar methods in the speech recognition community.

Here, core ideas of the previous chapters were amalgamated and further developed to demonstrate that beyond providing relative improvements, these methods are also capable of yielding competitive results. Before introducing a new method into the joint framework, however, we first revisited the settings of the said framework, in order to examine whether it was capable of providing competitive results just by itself. For this, we first fine-tuned such parameters of the joint training approach that we had not examined earlier. Then, motivated by the successful use of Δ and $\Delta\Delta$ coefficients in Chapter 2, we included the use of Δ-like coefficients into the joint training framework. We examined the effect of these two modifications using the TIMIT speech corpus and the Aurora-4 database. The results showed that both modifications lead to significant improvements in the recognition results using both speech database. We also compared the performance of our modified framework with results reported in the recent speech recognition literature. We did so using the TIMIT speech corpus (see Table 7), as well as using the Aurora-4 database (see Table 8). We see from the given tables that better results have been achieved on both databases. We also notice, however, that due to the modifications described here, our method can produce (taking into account all relevant factors) competitive results with those got using similar methods.

Method	WER
Chang and Morgan [5]	16.6%
Seltzer et al. [43]	12.4%
Martinez et al. [30]	12.3%
Baby et al. [1]	11.9%
Current work	*11.6%*
Narayanan and Wang [34]	11.1%
Rennie et al. [39]	**10.3%**

Table 8: Word error rates (WERs) presented in the literature on Aurora-4 using multi-condition training. The best score is highlighted in bold, while our result is highlighted in Italics.

Method	WER
CNN with ARMA features, band dropout	**16.0%**
Multi-band CNN with ARMA features	17.8%
DNN with ARMA features plus DCT [9]	18.5%
DNN with DNN speech enhancement of FBANK [15]	17.5%
DNN with Spectral masking [29]	22.8%
CNN with PNS features plus Gabor Filter Kernels [5]	22.9%
DNN with Exemplar Based Enhancement [1]	26.8%

Table 9: Comparing the band dropout result on ARMA features with our baseline, our earlier results, and with results cited in the recent literature, when using the clean training scenario.

Lastly, we supplemented our framework with band dropout, a method inspired by the input dropout scheme [14]. Here, however, instead of discarding features independently, we proposed a version of input dropout that discards whole frequency bands. We hypothesise that this strategy forces the network to rely less on the whole spectrum, and makes it more robust to channel mismatches. An evaluation was carried out on the Aurora-4 database with both the multi-condition and the clean training scenario, using mel filterbank and ARMA features. Our most competitive results were obtained with the ARMA features using the clean training scenario, where the performance got using our method was on par with the state-of-the-art (see Table 8).

Summary of Theses

IV/1. We presented two modifications of our joint training framework. We then evaluated these modifications using the TIMIT speech corpus and the Aurora-4 database. Using the results of these experiments, we demonstrated that both modifications actually improve the performance of speech recognition, and that the joint framework is capable of producing results that are competitive with the state-of-the-art (published in [24]).

IV/2. We presented a novel input dropout method, which is indeed beneficial in combination with CNN based acoustic modelling. The effectiveness of this method was demonstrated using the Aurora-4 database with different input representations, and for different training scenarios (published in [27]).

	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]
I/1.	•	•							
I/2.		•							
I/3.								•	
II/1.			•					•	
II/2.				•					
III/1.							•		
III/2.					•				
IV/1.						•			
IV/2.									•

Table 10: The connection between the thesis points and the publications of the Author.

References

[1] Baby, D., Gemmeke, J. F., Virtanen, T., and Van hamme, H. Exemplar-based speech enhancement for Deep Neural Network based automatic speech recognition. In *Proc. ICASSP* (2015), pp. 4485–4489.

[2] Baby, D., and Van hamme, H. Investigating modulation spectrogram features for Deep Neural Network-based automatic speech recognition. In *Proc. Interspeech* (2015), pp. 2479–2483.

[3] Bourlard, H., and Dupont, S. Subband-based speech recognition. In *Proc. ICASSP* (1997), pp. 1251–1254.

[4] Bouvrie, J., Ezzat, T., and Poggio, T. Localized spectro-temporal cepstral analysis of speech. In *Proc. ICASSP* (2008).

[5] Chang, S.-Y., and Morgan, N. Robust CNN-based speech recognition with Gabor filter kernels. In *Proc. Interspeech* (2014), pp. 905–909.

[6] Chi, T., Ru, P., and Shamma, S. A. Multiresolution spectrotemporal analysis of complex sounds. *J. Acoust. Soc. Am. 118*, 2 (2005), 887–906.

[7] Duchnowski, P. *A New Structure for Automatic Speech Recognition.* PhD thesis, MIT, 1993.

[8] Ezzat, T., Bouvrie, J. V., and Poggio, T. A. Spectro-temporal analysis of speech using 2D Gabor filters. In *Proc. Interspeech* (2007), pp. 506–509.

[9] Ganapathy, S. Robust speech processing using ARMA spectrogram models. In *Proc. ICASSP* (2015), pp. 5029–5033.

[10] Glorot, X., and Bengio, Y. Understanding the difficulty of training Deep Feedforward Neural Networks. In *Proc. AISTATS* (2010).

[11] Graves, A., Mohamed, A., and Hinton, G. E. Speech recognition with Deep Recurrent Neural Networks. In *Proc. ICASSP* (2013), pp. 6645–6649.

[12] Heřmanský, H., and Sharma, S. TRAPS-classifiers of temporal patterns. In *Proc. ICSLP* (1998), pp. 1003–1006.

[13] Hinton, G., Deng, L., Yu, D., Abdel-rahman, M., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Dahl, G., and Kingsbury, B. Deep Neural Networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine 29*, 6 (2012), 82–97.

[14] Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Improving Neural Networks by preventing co-adaptation of feature detectors. *CoRR abs/1207.0580* (2012).

[15] Jun, D., Qing, W., Tian, G., Yong, X., Li-Rong, D., and Chin-Hui, L. Robust speech recognition with speech enhanced Deep Neural Networks. In *Proc. Interspeech* (2014), pp. 616–620.

[16] Kanedera, N., Arai, T., Heřmanský, H., and Pavel, M. On the relative importance of various components of the modulation spectrum for automatic speech recognition. *Speech Communication 28*, 1 (1999), 43–55.

[17] Kleinschmidt, M. *Robust Speech Recognition Based on Spectro-Temporal Processing*. PhD thesis, Carl-von-Ossietzky Universitt Oldenburg, 2002.

[18] Kleinschmidt, M., and Gelbart, D. Improving word accuracy with Gabor feature extraction. In *Proc. ICSLP* (2002).

[19] Kovács, G., and Tóth, L. Localized spectro-temporal features for noise-robust speech recognition. In *Proc. ICCCONTI* (2010), pp. 481–485.

[20] Kovács, G., and Tóth, L. Phone recognition experiments with 2D-DCT spectro-temporal features. In *Proc. SACI* (2011), pp. 143–146.

[21] Kovács, G., and Tóth, L. The joint optimization of spectro-temporal features and Neural Net classifiers. In *Proc. TSD* (2013), pp. 552–559.

[22] Kovács, G., and Tóth, L. Joint optimization of spectro-temporal features and Deep Neural Nets for robust automatic speech recognition. *Acta Cybernetica 22*, 1 (2015), 117–134.

[23] Kovács, G., and Tóth, L. Multi-band noise robust speech recognition using Deep Neural Networks (in Hungarian). In *Proc. MSZNY* (2016), pp. 287–294.

[24] Kovács, G., and Tóth, L. Optimisation of a spectro-temporal feature selection method integrated in Deep Neural Networks (in Hungarian). In *Proc. MSZNY* (2017), pp. 158–169.

[25] Kovács, G., Tóth, L., and Grósz, T. Robust multi-band ASR using Deep Neural Nets and spectro-temporal features. In *Proc. SPECOM* (2015), pp. 386–393.

[26] Kovács, G., Tóth, L., and Van Compernolle, D. Selection and enhancement of Gabor filters for automatic speech recognition. *IJST 18*, 1 (2015), 1–16.

[27] Kovács, G., Tóth, L., Van Compernolle, D., and Ganapathy, S. Increasing the robustness of CNN acoustic models using autoregressive moving average spectrogram features and channel dropout. *Pattern Recognition Letters* (2017).

[28] Lamel, L. F., Kassel, R., and Seneff, S. Speech database development: design and analysis of the acoustic-phonetic corpus. In *Proc. DARPA Speech Recognition Workshop, Report no. SAIC-86/1546* (1986).

[29] Li, B., and Sim, K. C. Improving robustness of Deep Neural Networks via spectral masking for automatic speech recognition. In *Proc. ASRU* (2013), pp. 279–284.

[30] Martínez, A. M. C., Moritz, N., and Meyer, B. T. Should Deep Neural Nets have ears? The role of auditory features in deep learning approaches. In *Proc. Interspeech* (2014), pp. 2435–2439.

[31] Mesgarani, N., Thomas, S., and Heřmansky, H. A multistream multiresolution framework for phoneme recognition. In *Proc. Interspeech* (2010), pp. 318–321.

[32] Mirghafori, N. *A Multi-Band Approach to Automatic Speech Recognition.* PhD thesis, International Computer Science Institute, 1999.

[33] Morgan, N., and Bourlard, H. Continuous speech recognition using multilayer perceptrons with Hidden Markov Models, 1990.

[34] Narayanan, A., and Wang, D. Joint noise adaptive training for robust automatic speech recognition. In *Proc. ICASSP* (May 2014), pp. 2504–2508.

[35] Okawa, S., Bocchieri, E., and Potamianos, A. Multi-band speech recognition in noisy environments. In *Proc. ICASSP* (1998), pp. 641–644.

[36] Parihar, N., and Picone, J. DSR front end LVCSR evaluation. Aurora Working Group AU/384/02, Institue for Signal and Information Processing, December 2002.

[37] Plahl, C., Sainath, T. N., Ramabhadran, B., and Nahamoo, D. Improved pre-training of deep belief networks using sparse encoding symmetric machines. In *Proc. ICASSP* (2012), pp. 4165–4168.

[38] Pudil, P., Novovičová, J., and Kittler, J. Floating search methods in feature selection. *Pattern Recogn. Lett. 15*, 11 (Nov. 1994), 1119–1125.

[39] Rennie, S. J., Dognin, P. L., Cui, X., and Goel, V. Annealed dropout trained maxout networks for improved LVCSR. In *Proc. ICASSP* (2015), pp. 5181–5185.

[40] Sainath, T. N., Kingsbury, B., rahman Mohamed, A., and Ramabhadran, B. Learning filter banks within a Deep Neural Network framework. In *Proc. ASRU* (2013), pp. 297–302.

[41] Saon, G., Kurata, G., Sercu, T., Audhkhasi, K., Thomas, S., Dimitriadis, D., Cui, X., Ramabhadran, B., Picheny, M., Lim, L., Roomi, B., and Hall, P. English conversational telephone speech recognition by humans and machines. *CoRR abs/1703.02136* (2017).

[42] Schädler, M. R., Meyer, B. T., and Kollmeier, B. Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition. *J. Acoust. Soc. Am. 131* (2012), 4134–4151.

[43] Seltzer, M. L., Yu, D., and Wang, Y. An investigation of Deep Neural Networks for noise robust speech recognition. In *Proc. ICASSP* (May 2013), pp. 7398–7402.

[44] Tóth, L. Convolutional Deep Rectifier Neural Nets for phone recognition. In *Proc. Interspeech* (2013), pp. 1722–1726.

[45] Tóth, L. Combining time- and frequency-domain convolution in Convolutional Neural Network-based phone recognition. In *Proc. ICASSP* (2014), pp. 190–194.

[46] Tóth, L., and Grósz, T. A comparison of Deep Neural Network training methods for large vocabulary speech recognition. In *Proc. TSD* (2013), pp. 36–43.

[47] Zhao, S. Y., Ravuri, S. V., and Morgan, N. Multi-stream to many-stream: using spectro-temporal features for ASR. In *Proc. Interspeech* (2009), pp. 2951–2954.