MSTRE-NET: MULTISTREAMING ACOUSTIC MODELING FOR AUTOMATIC LYRICS TRANSCRIPTION

Emir Demirel
Queen Mary
University of London
e.demirel@qmul.ac.uk

Sven Ahlbäck
Doremir Music Research AB
sven.ahlback@doremir.com

Simon Dixon
Queen Mary
University of London
s.e.dixon@qmul.ac.uk

ABSTRACT

This paper makes several contributions to automatic lyrics transcription (ALT) research. Our main contribution is a novel variant of the Multistreaming Time-Delay Neural Network (MTDNN) architecture, called MSTRE-Net, which processes the temporal information using multiple streams in parallel with varying resolutions keeping the network more compact, and thus with a faster inference and an improved recognition rate than having identical TDNN streams. In addition, two novel preprocessing steps prior to training the acoustic model are proposed. First, we suggest using recordings from both monophonic and polyphonic domains during training the acoustic model. Second, we tag monophonic and polyphonic recordings with distinct labels for discriminating non-vocal silence and music instances during alignment. Moreover, we present a new test set with a considerably larger size and a higher musical variability compared to the existing datasets used in ALT literature, while maintaining the gender balance of the singers. Our best performing model sets the state-of-the-art in lyrics transcription by a large margin. For reproducibility, we publicly share the identifiers to retrieve the data used in this paper.

1. INTRODUCTION

Empirical studies show that it is a challenging task even for human listeners to recognize sung words, and this is more challenging than speech, due to a number of performance, environment and listener related factors [1]. Thus the automatic retrieval of sung words through machine listening, i.e. automatic lyrics transcription (ALT), can potentially be impactful in easing some of the time consuming processes involved in composing music, audio/video/music score captioning and editing, lyrics alignment, music catalogue creation, etc. Despite its potential, the current state of lyrics transcription is far from being sufficiently robust to be leveraged in such applications.

With recent advances in automatic speech recognition (ASR) research and its successful adaptation to singing data, considerable improvements have been reported in ALT research [2–4]. In addition to this, newly released datasets have accelerated the development of the research field [5, 6]. Through these improvements, the prospect of applying ALT in the music industry has become more realistic, assuming that progress continues. Although promising results have been obtained for a cappella recordings [2, 4, 7], recognition rates drop considerably in the presence of instrumental accompaniment [8, 9].

From the perspective of ASR, music accompaniment can be regarded as noise since non-vocal music signals generally include minimal or no information relevant to lyrics transcription, while their presence in the spectral domain increases the confusions during prediction. For building more robust acoustic models against noisy environments, the multistream approach in ASR was introduced [10], inspired by how the acoustic signals are split into multiple frequency bands and processed in parallel in the human auditory system [11]. While previous research suggested using multiresolution feature processing [12, 13] or reconstruction of a multi-band latent representation through autoencoders [14] to achieve multistreaming ASR, the neural network architecture recently introduced in [15], Multistreaming Time-Delay Neural Network (MTDNN), proposes a simplified solution which is utilized in producing the state-of-the-art for hybrid / Deep Neural Network - Hidden Markov Model (DNN-HMM) based ASR [16, 17]. In this work, we propose a compact variant of MTDNN, referred to as MSTRE-Net, where streams are diversified by having different numbers of layers with the goal of reducing the number of trainable parameters (i.e. model complexity), and thus the inference times and improving the word recognition rates.

Additionally, we propose a number of other novel contributions for improving lyrics transcription performance. We suggest combined training of the acoustic model on both monophonic (e.g. *DAMP*-Sing! 300x30x2 [6]) and polyphonic (e.g. *DALI* [5]) recordings, which is shown to improve performance for both cases. Furthermore, we propose tagging monophonic and polyphonic utterances with separate *music* and *silence*

ED received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 765068.

Attribution: Emir Demirel, Sven Ahlbäck, Simon Dixon, "MSTRE-Net: Multistreaming Acoustic Modeling for Automatic Lyrics Transcription", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

tokens explicitly. Our goal for this is to generate alignments that are more robust against disruptions in the decoding path, potentially caused by the musical accompaniment during the non-vocal frames.

One major challenge in ALT research has been publishing reproducible results, due to the lack of publicly available evaluation data [18]. Dabike and Barker [2] shared manually verified annotations for a subset of *DAMP* which have been utilized for evaluation in a cappella singing [4, 7]. The Jamendo (lyrics) dataset [19] consists of 20 contemporary polyphonic music recordings released under an open source license. Moreover, despite their limited nature in terms of size and musical variability, Hansen [20] and Mauch [21] have been among the two most commonly used evaluation sets for ALT. In addition to these, we present a new test set with 240 polyphonic recordings having a larger span of release dates and better singer gender balance in order to establish a more comprehensive lyrics transcription evaluation.

The rest of the paper is structured as follows: we begin with a summary of essential concepts in the state-of-the-art approach for hybrid-ASR. The following section explains how the proposed MTDNN architecture is constructed. Next, we give details of the data used in experiments, and introduce a new evaluation set. Finally, we describe the experimental setup and present results verifying our design choices through ablative tests.

2. BACKGROUND

ALT can be considered as analogous to Large Vocabulary Continuous Speech Recognition (LVCSR) for the singing voice. Similarly, the goal for ALT is predicting the most likely word sequence, $\mathbf{w}$, given a stream of acoustic observations, $\mathbf{O}$, which can be expressed in mathematical terms as follows:

$$\begin{aligned}\widehat{\mathbf{w}} = &= \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}|\mathbf{O}) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w})p(\mathbf{O}|\mathbf{w}) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}) \sum_{\mathbf{Q}} p(\mathbf{O}|\mathbf{Q})P(\mathbf{Q}|\mathbf{w}), \end{aligned} \tag{1}$$

where elements of $\mathbf{Q}$ represent the phoneme classes [1]. In Equation 1, $p(\mathbf{O}|\mathbf{Q})$ is obtained via the acoustic model. Phonemes are converted to word labels using a lexicon which defines a mapping between words and their phonemic representations. The raw word posteriors are then smoothed with the language model, $P(\mathbf{w})$ for obtaining grammatically more plausible output transcriptions, $\widehat{\mathbf{w}}$.

According to the probabilistic approach of ASR, phonemes are represented with HMMs where a transition between connected phone states occurs at every time step [22]. In our system, we employ the *Kaldi* toolkit [23], an open-source ASR framework that represents HMM states using Weighted Finite State Transducers (WFST) [24]. In operation, a WFST graph is generated through composing posteriors retrieved from the acoustic, language and pronunciation models. The resulting directed paths of states form a *lattice*, a weighted acyclic graphical structure which can represent multiple output hypotheses.

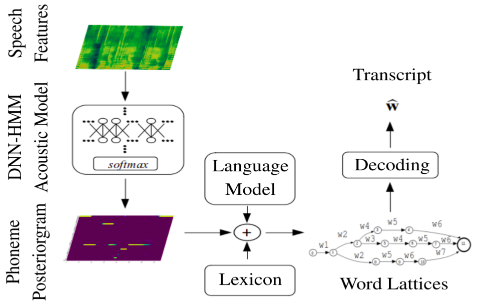


Figure 1. DNN-HMM based ASR at operation

2.1 Lattice-free Maximum Mutual Information

Most recent ALT systems utilize the state-of-the-art hybrid DNN-HMM framework where the neural networks are trained in a sequence discriminative fashion [25]. More specifically, the best performing lyrics transcribers to date [2–4, 7, 8] use Lattice-free Maximum Mutual Information training (LF-MMI) [26], where network parameters are tuned w.r.t. the MMI objective:

$$\mathcal{F}_{MMI} = \sum_{u} \log \frac{p(\mathbf{O}_u|Q_u)^{\mathcal{K}} P(W_u)}{\sum_W p(\mathbf{O}_u|Q)^{\mathcal{K}} P(W)} \tag{2}$$

where $p(\mathbf{O}_u|Q_u)$ is the probability of observing an acoustic instance O in the utterance u, in Markovian phone state Q_u, and the $P(W)$'s are the word sequence probabilities [27]. Optimization w.r.t. MMI aims at maximizing the shared information between the reference and target sequences. More explicitly, the terms in the numerator are calculated per utterance, whereas the denominator is computed over the entire training set. Hence, the network parameters are updated to maximize the probabilities in the numerator and minimize the denominator. In other words, the goal of MMI training is to discriminate a certain acoustic observation with its given utterance.

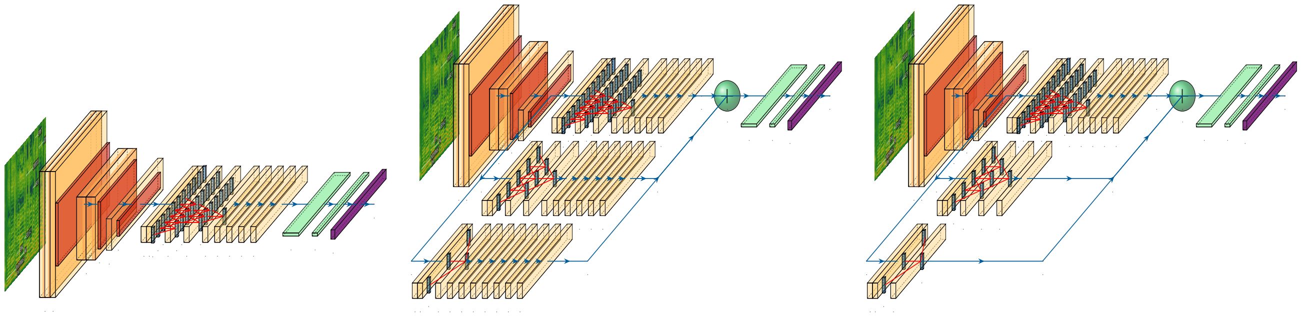
3. MULTISTREAMING TIME-DELAY NEURAL NETWORKS

The main body of MTDNN architectures consists of multiple streams of TDNN layers trained in parallel, where each stream has a unique time dilation rate (τ). Our proposed MTDNN variant differs from the original models [16, 17] by having different numbers of layers in the TDNN streams, depending on τ (Figures 2(b) and 2(c)).

Prior to the TDNN streams, input features $\mathbf{x}$ are first processed by a single stream 2-D Convolutional Neural Network (CNN) in the front-end of the network,

$$\mathbf{h} = \mathit{Stacked\text{-}2D\text{-}CNN}(\mathbf{x}) \tag{3}$$

[1] A phoneme is the basic sonic unit of speech. In linguistics, words are considered to be composed of sequences of phonemes.



(a) Single-stream TDNN (τ=3) (b) MTDNN with identical streams (τ={3,6,9}) (c) MTDNN with distinct streams (τ={3,6,9})

Figure 2. Different variants of TDNN architectures. From left-to-right, the orange, yellow and cyan blocks represent the front-end 2-D CNN, TDNN streams and final FC layers preceded by the purple softmax layer

where $\textit{Stacked-2D-CNN}$ stands for the stack of 2-D convolutional layers with 3×3 kernels. Inspired by [4], we apply subsampling on the height axis after each alternate layer with a factor of 2, in order to get compact embeddings, $\mathbf{h}$, which are then fed into multiple streams of TDNNs [2]. Each stream of TDNNs has a unique time dilation rate, τ, encoding information in different temporal resolutions,

$$\mathbf{z}_t^n = \textit{Stacked-TDNN}(\mathbf{h}|\tau = t, N = n), \tag{4}$$

where $\mathbf{z}_\tau^n$ are the latent variables at the output of the final (N^{th}) TDNN layer, and $t \in \mathbb{Z}$. These are concatenated and projected to the classification ($softmax$) layer by a pair of fully connected (FC) layers,

$$a(s) = softmax(2 \times FC(Concat(\mathbf{z}_{\tau_1}^N, \mathbf{z}_{\tau_2}^N, ..., \mathbf{z}_{\tau_K}^N))), \tag{5}$$

where $a(s)$ is the activation of the $softmax$ layer corresponding to the phoneme state s and K is the number of TDNN streams. We decide the number of layers per stream w.r.t. to the receptive field (RF) of the nodes at the top TDNN layer,

$$RF_{\mathbf{z}_\tau^N} = 2 \times l \times \tau \times N, \tag{6}$$

where l is the frame length of the acoustic feature vectors. Note that we include an additional 1-D convolutional layer just before the TDNN streams. This layer does not use dilation, in order not to skip any frames.

4. DATA

4.1 Training Set

The acoustic models of the previously presented ALT systems in the literature are built on either monophonic or polyphonic music recordings. In general, monophonic models are trained on the *DAMP*^{train} dataset [2, 4, 7], while *DALI* is utilized for polyphonic models [9]. We merge these two datasets, exploiting their size and musical variability. We curated the polyphonic subset of the dataset on the recordings from the most recent version (v2.0) of the *DALI* dataset [30], and included only those songs for which the Youtube links were available and still in use at the time of the audio retrieval.

4.2 Evaluation Sets

We perform model selection and optimization on the subsets of the *DAMP* and *DALI* datasets, representing monophonic and polyphonic domains respectively. For *DAMP*^{test}, we use the test split introduced in [2]. For testing the lyrics transcription performance on polyphonic recordings, we curated a new subset of *DALI*-v1.0, which we give the data selection procedure below. Finally, we evaluate our best performing model on the three benchmark datasets used in the literature, namely the Jamendo, Hansen and Mauch sets and provide a comparison with the state of the art in Section 6.5.

Set	Words	Uniq. Words	# Utt.	# Rec	# Singers	Avg. Utt. Dur.	Total Dur.
LM-corpus	13M	100k	2M	N/A	N/A	N/A	N/A
DAMP^{train}	686k	5.3k	80k	4.2k	3k	5sec	112h
DALI^{train}	1.1M	25.5k	227k	4.1k	1.5k	2.48sec	156h
DAMP^{dev}	4k	695	482	66	38	5.12sec	41min
DALI^{dev}	5.7k	941	1.7k	34	16	2.41sec	48min
DAMP^{test}	4.6k	840	479	70	40	6sec	48min
DALI^{test}	62.8k	4.2k	240	240	160	233sec	15.5h
Jamendo	5.7k	1k	20	20	20	216sec	72min
Hansen	2.8k	585	10	10	9	214sec	35min
Mauch	5.2k	820	20	20	18	245sec	82min

Table 1. Statistics of datasets used in experiments

For tuning the hyperparameters during evaluation, the language model scaling factor and the word insertion penalty, we have used the combination of the data from *DAMP*^{dev} split [2] and 20 recordings from *DALI*-v2.0 [3].

4.2.1 The DALI-test set

In this section, we give details of the curation procedure for the *DALI*^{test} set. We began from the subset presented in [31], which initially had 513 recordings and filtered it according to a number of criteria. Numerous audio samples were not retrievable from the links provided. We obtained the Youtube links through *automatic search*

[2] A time-delay neural network consists of 1-D convolutional layers where the convolution is applied with frames that are dilated on the time axis [28]. In our architecture, we employ the factorized variant of TDNN introduced in [29].

[3] This combined development set is denoted as *dev* in Section 6.

using relevant key words. We discarded songs where the automatically retrieved version was a live performance, had low audio quality or contained extra background speech sections unrelated to its corresponding lyrics. For consistency and fair evaluation, we did not include songs where the dominant language was not English. We allowed for an artist to have at most 5 songs. Among the remaining recordings, we manually selected a subset having a relatively balanced distribution of singers' gender, official release dates over decades (see Figure 3) and variability in terms of singing styles, vocal effects and music genre. Lyrics were initially obtained from the annotations provided in [5] and manually verified following the steps explained in Section 5.1. The final version of *DALI*^{test} consists of 240 recordings, which sets the largest test set for lyrics transcription with clean annotations. For open science, we publicly share the data identifiers, cleaned lyrics annotations and a tutorial to retrieve the corresponding Youtube links at "*https://github.com/emirdemirel/DALI-TestSet4ALT*".

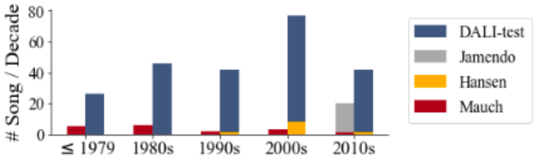


Figure 3. Songs per decade in ALT evaluation sets

5. EXPERIMENTAL SETUP

5.1 Lyrics Preprocessing

Prior to being utilized for training, raw lyrics data automatically retrieved from online resources (as in *DALI*) needs to be normalized, as the transcription rules applied by lyrics providers are not standardized. We remove all special ASCII characters except apostrophes. We convert numeric characters to their alphabetic correspondence. All text is converted to upper case. We observed several samples with erroneous hyphenation, explicit syllabification and repeating letters (possibly indicating longer uttered syllables or vowels). To cope with these, we apply automatic hyphenation correction and canonicalization using the standard open-source *NLTK* tools [4]. The output lyrics are then verified and corrected manually.

5.2 Language and Pronunciation Models

Lyrics often contain uncommon words that are not very likely to exist in standard pronunciation dictionaries. For such words that are not in the lexicon, or out-of-vocabulary (OOV) words, we generate pronunciations using a pretrained grapheme-to-phoneme (G2P) converter [32]. In order to produce fair and reproducible results, we generate pronunciations for the OOV words in the evaluation sets as well, and do not skip these during evaluation. We utilize the commonly used CMU English Pronunciation Dictionary [5] as the lexicon and generate alternative pronunciations by duplicating the vowel phonemes for each word pronunciation, inspired by the improvements observed in [7, 33]. A 4-gram language model (LM) is constructed using the SRILM Toolkit [34]. We use the combination of the lyrics corpus in [2] [6] and *DALI*^{train}. For scientific evaluation, we exclude any songs which overlap with those in the evaluation sets.

5.3 Discriminating Instrumental and Silent Regions

The hybrid DNN-HMM ASR framework approaches the continuous word recognition task essentially as a sequence decoding problem. Within this scope, the presence of instrumental accompaniment, especially during non-vocal regions, may disrupt the decoding path, potentially causing cumulative errors during transcription and alignment. Traditionally, non-speech regions are represented with a special silence token within the target class set during recognition. Here, we propose using separate tokens for the non-vocal instances in monophonic and polyphonic recordings. Prior to training, we associate these tokens with their corresponding silence/music instances by explicitly adding tags at the beginnings and the ends of the ground truth lyrics of each utterance (see Table 2). These tags are represented as words in the lexicon where their pronunciations correspond to the relevant silence token.

	w
Raw	$w_1\ w_2\ ...\ w_N$
DAMP	*<silence>* $w_1\ w_2\ ...\ w_N$ *<silence>*
DALI	*<music>* $w_1\ w_2\ ...\ w_N$ *<music>*

Table 2. Music / silence tagging w.r.t. the dataset

For silence and music tagging, we exploit our training data. As we know the recordings in *DAMP* and *DALI* are monophonic and polyphonic respectively, we apply tagging w.r.t. the dataset. The pronunciations of these pseudo-word tags are represented with distinct phonemes in the lexicon.

5.4 Generating Phoneme Alignments

Since the neural network optimization is performed w.r.t. phoneme posteriors (as explained in Section 2.1), we need to extract their timings, i.e. *alignments*. To generate these, we train a triphone Gaussian Mixture Model (GMM) - HMM model on "singer adaptive" features [35], following the standard Kaldi recipe [7]. At this stage, we compute per-word pronunciation probabilities following the steps in [36], and retrain another triphone model using the updated lexicon transducer. Using this new model, we apply *forced alignment* [22] on the training data for generating the phoneme and word alignments.

[4] These steps are potentially language specific.

[5] Link: *https://github.com/Alexir/CMUdict/blob/master/cmudict-0.7b*.

[6] This corpus contains lyrics from all the songs of the artists included in the Billboard charts between 2015-2018, plus the lyrics in *DAMP*^{train}.

[7] We execute the GMM-HMM pipeline at *https://github.com/emirdemirel/ALTA*, which is almost the same procedure as the standard *librispeech* recipe, with tuned hyperparameters for singing data.

5.5 Neural Network Training

DNN training is based on the *Kaldi - chain* recipe. In the feature space, we use 40-band filterbank features extracted with a hop size of 10ms and window size of 30ms. To achieve singer-adaptive training, we utilize i-Vectors [37] which represent the singer identity information via global embedding vectors. Frame subsampling is applied with a factor of 3 in this training scheme where each subsampled frame in the input of the neural network is considered to represent $l = 3 \times 10$ms = 30ms of context. The data is fed into the network as audio chunks of 4.2 seconds (140 frames) in minibatches of 32. We apply a decaying learning rate with beginning and final rates of 10^{-4} and 10^{-5} respectively. Stochastic gradient descent is used as the optimizer. The training is done for 6 epochs.

6. RESULTS

We report lyrics transcription results based on word error rate (WER). We begin by comparing performances obtained using *DAMP* and/or *DALI* in training the acoustic model. Then we test our proposed idea of discriminating silence and accompaniment instances by using separate tokens, and perform experiments testing different topologies of the MTDNN architecture. To boost the performance further, we train a final model on augmented data and provide a comparison of our results with previously published models.

6.1 Multi-Domain Training

According to Table 3, the model trained on *DAMP*^{train} performs relatively well on *DAMP*^{test}, however its performance drops dramatically on polyphonic recordings. On the other hand, much better recognition rates are observed on *DALI*^{test} when a polyphonic model is used, but then the polyphonic model performs poorly on a cappella recordings. Finally, using recordings from both the monophonic and polyphonic domains results in improved performance on both polyphonic and monophonic test sets, although the improvement is marginal on the monophonic *DAMP*^{test} set.

6.2 Music / Silence Modeling

Next, we test whether the explicit music/silence tagging improves transcription results. At this stage, we use a single stream architecture (M_8^{single} in Table 4). Tagging is applied only in constructing the GMM-HMM model and for generating alignments. The music/silence tags were removed during neural network training. Table 3 shows that alignment with music/silence tags did result in considerably improved recognition results for polyphonic recordings, but no improvement was evident for the monophonic case.

6.3 Neural Architecture Design

Here, we test various parameterizations of the multistreaming architecture. In this stage, we did not use the explicit music and silence tagging for training the models. As mentioned in Section 3, we diversify each stream of TDNNs in terms of the number of hidden layers and/or their dimensions. In addition to achieving improved performance, the goal of these modifications is to exploit the temporal context to its full extent. For this, we calculate the number of TDNN layers included w.r.t. the resulting $RF_{\mathbf{z}_\tau^N}$.

Train Set	*DAMP*^{test}	*DALI*^{test}
DAMP	17.64	78.42
DALI	61.95	59.19
DAMP + *DALI*	**17.14**	53.86
+ music/sil tag	17.29	**47.00**

Table 3. Multi-domain training and music/silence tagging results

In all MTDNN variants tested, we use 3 TDNN streams with $\tau \in \{3, 6, 9\}$. We begin with finding the optimal number of TDNN layers for the stream with the smallest τ. For rapid experimentation, we use single-streaming TDNN models (M_N^{single} in Table 4). According to Table 4, using 9 layers sets the optimal setup for $\tau = 3$, having $RF_{\mathbf{z}_{\tau=3}^{N=9}} = 1620$ms. Note that further increasing the number of TDNN layers to 10 ($RF_{\mathbf{z}_{\tau=3}^{10}} = 1800$ms) did not result in improved recognition, and the model complexity was much higher (Figure 4). Therefore, we chose as our baseline a single-stream model with 9 TDNN layers.

	Stream	Layers	Dimension	*dev*	*DAMP*^{test}	*DALI*^{test}
M_7^{single}	3	7	512	28.06	17.08	54.52
M_8^{single}	3	8	512	28.05	17.14	53.44
M_9^{single}	3	9	512	27.68	**17.08**	**52.25**
, M_{10}^{single}	3	10	512	*27.67*	17.21	53.58
$M_{9,a}^{multi}$	3-6-9	(9,9,9)	(512,512,512)	26.69	16.75	51.38
$M_{9,b}^{multi}$	3-6-9	(9,4,3)	(512,512,512)	*26.65*	**16.45**	**49.32**
$M_{9,c}^{multi}$	3-6-9	(9,9,9)	(512,256,172)	27.13	16.08	52.54
$M_{9,d}^{multi}$	3-6-9	(9,4,3)	(512,256,172)	27.38	16.62	51.92

Table 4. Experiments on NN design

Next, we perform ablative tests on four variants of the multistream architecture (notated as $M_{9,\{a,b,c,d\}}^{multi}$). Model $M_{9,a}^{multi}$ have identical TDNN structures (except for τ), whereas the variants $M_{9,\{b,c\}}^{multi}$ have reduced N or hidden dimensions respectively w.r.t. τ at each stream. Both dimensions of model reduction are applied on $M_{9,d}^{multi}$. In models $M_{9,\{b,d\}}^{multi}$, we reduced the number of layers, N for the streams with larger τ to keep $RF_{\mathbf{z}_\tau^N}$ similar across all streams. $M_{9,\{b,d\}}^{multi}$ have 4 and 3 layers at the streams with $\tau = 6$ and $\tau = 9$ having RF values of 1440 and 1620ms respectively. On the other hand, adding one more layer on the streams with $\tau = 6, 9$ would result in having $RF_{\mathbf{z}_\tau^N} \geq 1800$ms which is shown above to be suboptimal in the single-stream case (see results for M_{10}^{single}).

6.4 Model Selection

The proposed multistreaming setups except $M_{9,c}^{multi}$ outperformed their single-stream counterpart, M_9^{single}, particularly on *DALI*^{test}. The best results are achieved with $M_{9,b}^{multi}$ which has unique N layers across all streams with the same hidden layer dimension.

To increase confidence in model selection, we investigate other operational aspects of the tested models. In Figure 4, we compare the number of trainable parameters which is a variable related to model complexity, and the real-time factor (RTF) that measures how fast a model operates during inference. We compute RTF's based on the inference times across all the data used in evaluation. We repeat this 5 times and report the mean of all iterations per model. These iterations are performed on an Intel® Xeon® Gold 5218R CPU.

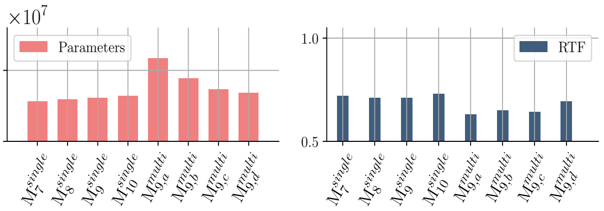


Figure 4. Num. trainable parameters (left) & RTFs (right)

According to Figure 4, our best performing model $M_{9,b}^{multi}$ has the second largest number of trainable parameters. Its model complexity is however much lower than that of $M_{9,a}^{multi}$, the architecture presented in [16]. In terms of run time, all multistreaming models performed faster than single-stream models, with $M_{9,b}^{multi}$ being among the fastest. This shows that our compact variant has a reduced inference time with an improved recognition rate as hypothesized in Section 1.

6.5 Comparison with the State of the Art

At this last step, we train a final model combining the music / silence aware alignment with the best performing MTDNN architecture, $M_{9,b}^{multi}$. To boost the performance further, we apply data augmentation via speed perturbation with the factors of 0.9 and 1.1, tripling the size of the training data. In Table 5, we compare our final model with other lyrics transcribers reported in the literature. We retrained the acoustic models in [2] ($M_{[2]}$), and [4] ($M_{[4]}$), using the corresponding publicly shared repositories. $M_{[9]}$ is based on the pretrained acoustic model shared at *https://github.com/chitralekha18/AutoLyrixAlign*. The same language model is used in constructing the decoding graphs for all the models in Table 5. Note that $M_{[2],[4]}$ are trained on *DAMP* (monophonic) and $M_{[9]}$ is trained on *DALI* (polyphonic) datasets. We used the best performing language model scaling factor when reporting the results in Table 5. We were not able to generate results on *DALI*^{test} using $M_{[9]}$ due to the model being highly memory intensive, as also reported in [8].

In addition to these, we provide a comparison with the state of the art. The best WER score reported on *DAMP*^{test} is based on $M_{[4]}$ and applies rescoring on the word lattices generated after the first-pass decoding using an RNNLM [38]. We did not apply RNNLM rescoring as we did not achieve consistent improvements across different test sets according to our empirical observations. For a fair comparison, we also include the best results in [4] achieved via n-gram LMs (the scores in paranthesis in Table 5). On Jamendo, the best WER scores were reported in [8] where the inference was performed on source separated vocals. For Hansen and Mauch datasets, the best results are provided as reported in [9] [8]. In order to evade optimistic results, we have discarded the overlapping songs between Hansen, Mauch and *DALI*^{train} during training the final model.

	WER				
	DAMP^{test}	*DALI*^{test}	Jamendo	Hansen	Mauch
$M_{[2]}$	16.86	67.12	76.37	77.59	76.98
$M_{[9]}$	56.90	N/A	50.64	39.00	40.43
$M_{[4]}$	17.16	76.72	66.96	78.53	78.50
S.O.T.A	**14.96** (17.01) [4]	N/A	51.76 [8]	47.01 [9]	44.02 [9]
MSTRE-Net	**15.38**	**42.11**	**34.94**	**36.78**	**37.33**

Table 5. Comparison with the state-of-the-art.

The results above show that MSTRE-Net outperforms all of the previously presented models on the polyphonic sets, with more than 15% , 7% and 6% absolute WER improvements achieved on the Jamendo, Hansen and Mauch datasets compared to the previous state of the art respectively. Notably, we achieved less than 50% WER on the large *DALI*^{test} set indicating more than half of the words across 240 songs were correctly predicted. Our model also has the best results on *DAMP*^{test} achieved via n-gram LM.

7. CONCLUSION

We have introduced MSTRE-Net, a novel compact variant of the multistreaming neural network architecture, which outperforms previously proposed automatic lyrics transcription models. Our model achieved these results with lower model complexity and inference time. In addition, we showed that recognition rates improved across all evaluation sets after leveraging both polyphonic and monophonic data in training the acoustic model. We proposed a novel data preprocessing method for generating alignments prior to neural network training which resulted in considerably better word recognition rates from polyphonic recordings compared to the baseline approach. Finally, we curated a new evaluation set that is more comprehensive and varied, while having a much larger size compared to the previous test data used in research. For reproducibility and open science, the identifiers and a tutorial on making use of this data will be shared with the research community.

Our final model outperformed the previously reported best ALT results by a large margin, setting the new state-of-the-art. Through these results, we have taken an important step in increasing the potential and the possibility for ALT being an applicable technology in both Music Information Retrieval research and the music technology industry.

[8] Note that the reason for the WER difference between $M_{[9]}$ and the scores reported in [9] is due to the bigger language model we used, despite both models having the same acoustic model.

8. REFERENCES

[1] P. A. Fine and J. Ginsborg, "Making myself understood: Perceived factors affecting the intelligibility of sung text," *Frontiers in Psychology*, vol. 5, p. 809, 2014.

[2] G. R. Dabike and J. Barker, "Automatic lyrics transcription from karaoke vocal tracks: Resources and a baseline system," in *Interspeech*, 2019.

[3] C. Gupta, R. Tong, H. Li, and Y. Wang, "Semi-supervised lyrics and solo-singing alignment," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[4] E. Demirel, S. Ahlbäck, and S. Dixon, "Automatic lyrics transcription using dilated convolutional neural networks with self-attention," in *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[5] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "DALI: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[6] "Smule sing! 300x30x2 dataset," accessed April, 2021, https://ccrma.stanford.edu/damp/.

[7] E. Demirel, S. Ahlbäck, and S. Dixon, "Computational pronunciation analysis in sung utterances," in *European Conference on Signal Processing (EUSIPCO)*, 2021.

[8] ——, "Low resource audio-to-lyrics alignment from polyphonic music recordings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[9] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics transcription in polyphonic music: Does background music help?" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[10] H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," in *Fourth International Conference on Spoken Language Processing (ICSLP)*, 1996.

[11] J. Allen, "How do humans process and recognize speech?" *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 567–577, 1994.

[12] H. Hermansky and P. Fousek, "Multi-resolution RASTA filtering for TANDEM-based ASR," in *Interspeech*, 2005.

[13] Z. Tüske, R. Schlüter, and H. Ney, "Acoustic modeling of speech waveform based on multi-resolution, neural network signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[14] S. H. Mallidi, T. Ogawa, K. Veselỳ, P. S. Nidadavolu, and H. Hermansky, "Autoencoder based multi-stream combination for noise robust speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] K. J. Han, R. Prieto, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1D convolutions," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019.

[16] K. J. Han, J. Pan, V. K. N. Tadala, T. Ma, and D. Povey, "Multistream CNN for robust acoustic modeling," in *Interspeech*, 2020.

[17] J. Pan, J. Shapiro, J. Wohlwend, K. J. Han, T. Lei, and T. Ma, "ASAPP-ASR: Multistream CNN and self-attentive SRU for SOTA speech recognition," in *Interspeech*, 2020.

[18] A. M. Kruspe, "Training phoneme models for singing with "songified" speech data." in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[19] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[20] J. K. Hansen, "Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients," in *Ninth Sound and Music Computing Conference (SMC)*, 2012.

[21] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 200–210, 2011.

[22] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Foundations and Trends in Signal Processing*, 2008.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[24] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, 2002.

[25] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Interspeech*, 2013.

[26] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI." in *Interspeech*, 2016.

[27] L. Bahl, P. Brown, P. De Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1986.

[28] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[29] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Interspeech*, 2018.

[30] G. Meseguer-Brocal, R. Bittner, S. Durand, and B. Brost, "Data cleansing with constrastive learning for vocal note event annotations," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[31] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d'Alché Buc, "Multilingual lyrics-to-audio alignment," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[32] J. R. Novak, D. Yang, N. Minematsu, and K. Hirose, "Phonetisaurus: A WFST-driven phoneticizer," in *International Workshop on Finite State Methods and Natural Language Processing*, 2012.

[33] C. Gupta, H. Li, and Y. Wang, "Automatic pronunciation evaluation of singing." in *Interspeech*, 2018.

[34] T. Alumäe and M. Kurimo, "Efficient estimation of maximum entropy language models with N-gram features: An SRILM extension," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[35] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Fourth International Conference on Spoken Language Processing*, 1996.

[36] G. Chen, H. Xu, M. Wu, D. Povey, and S. Khudanpur, "Pronunciation and silence probability modeling for ASR," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[37] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[38] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned RNNLM lattice-rescoring algorithm for automatic speech recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.