EXPLORING DATA AUGMENTATION FOR IMPROVED SINGING VOICE DETECTION WITH NEURAL NETWORKS

Jan Schlüter and Thomas Grill
Austrian Research Institute for Artificial Intelligence, Vienna
jan.schlueter@ofai.at thomas.grill@ofai.at

ABSTRACT

In computer vision, state-of-the-art object recognition systems rely on label-preserving image transformations such as scaling and rotation to augment the training datasets. The additional training examples help the system to learn invariances that are difficult to build into the model, and improve generalization to unseen data. To the best of our knowledge, this approach has not been systematically explored for music signals. Using the problem of singing voice detection with neural networks as an example, we apply a range of label-preserving audio transformations to assess their utility for music data augmentation. In line with recent research in speech recognition, we find pitch shifting to be the most helpful augmentation method. Combined with time stretching and random frequency filtering, we achieve a reduction in classification error between 10 and 30%, reaching the state of the art on two public datasets. We expect that audio data augmentation would yield significant gains for several other sequence labelling and event detection tasks in music information retrieval.

1. INTRODUCTION

Modern approaches for object recognition in images are closing the gap to human performance [5]. Besides using an architecture tailored towards images (Convolutional Neural Networks, CNNs), large datasets and a lot of computing power, a key ingredient in building these systems is *data augmentation*, the technique of training and/or testing on systematically transformed examples. The transformations are typically chosen to be label-preserving, such that they can be trivially used to extend the training set and encourage the system to become invariant to these transformations. As a complementary measure, at test time, aggregating predictions of a system over transformed inputs increases robustness against transformations the system has not learned to (or not been trained to) be fully invariant to.

While even earliest work on CNNs [13] successfully employs data augmentation, and research on speech recognition – an inspiration for many of the techniques used in music information retrieval (MIR) – has picked it up as well [9], we could only find anecdotal references to it in the MIR literature [8, 18], but no systematic treatment.

In this work, we devise a range of label-preserving audio transformations and compare their utility for music signals on a benchmark problem. Specifically, we chose the sequence labelling task of singing voice detection: It is well-covered, but best reported accuracies on public datasets are around 90%, suggesting some leeway. Furthermore, it does not require profound musical knowledge to solve, making it an ideal candidate for training a classifier on low-level inputs. This allows observing the effect of data augmentation unaffected by engineered features, and unhindered by doubtable ground truth. For the classifier, we chose CNNs, proven powerful enough to pick up invariances taught by data augmentation in other fields.

The following section will review related work on data augmentation in computer vision, speech recognition and music information retrieval, as well as the state of the art in singing voice detection. Section 3 describes the method we used as our starting point, Section 4 details the augmentation methods we applied on top of it, and Section 5 presents our findings. Finally, Section 6 rounds up and discusses implications of our work.

2. RELATED WORK

For computer vision, a wealth of transformations has been tried and tested: As an early example (1998), Le et al. [13] applied translation, scaling (proportional and disproportional) and horizontal shearing to training images of handwritten digits, improving test error from 0.95% to 0.8%. Krizhevsky et al. [12], in an influential work on large-scale object recognition from natural images, employed translation, horizontal reflection, and color variation. They do not provide a detailed comparison, but note that it allowed to train larger networks and the color variations alone improve accuracy by 1 percent point. Crucially, most methods also apply specific transformations at test time [5].

In 2013, Jaitly and Hinton [9] pioneered the use of label-preserving audio transformations for speech recognition. They find pitch shifting of spectrograms prior to mel filtering at training and test time to reduce phone error rate from 21.6% to 20.5%, and report that scaling mel spectra either in time or frequency dimensions or constructing examples from perturbated LPC coefficients did not help. Concurrently, Kanda et al. [10] showed that combining pitch shift-

ing with time stretching and random frequency distortions reduces word errors by 10%, with pitch shifting proving most beneficial and effects of the three distortion methods adding up almost linearly. Cui et al. [3] combined pitch shifting with a method transforming speech to another speaker's voice in feature space and Ragni et al. [20] combined it with unsupervised training, both targetting uncommon languages with small datasets. To the best of our knowledge, this comprises the full body of work on data augmentation in speech recognition.

In MIR, literature is even more scarce. Li and Chan [18] observed that Mel-Frequency Cepstral Coefficients are sensitive to changes in tempo and key, and show that augmenting the training and/or test data with pitch and tempo transforms slightly improves genre recognition accuracy on the GTZAN dataset. While this is a promising first step, genre classification is a highly ambiguous task with no clear upper bound to compare results to. Humphrey et al. [8] applied pitch shifting to generate additional training examples for chord recognition learned by a CNN. For this task, pitch shifting is not label-preserving, but changes the label in a known way. While test accuracy slightly drops when trained with augmented data, they do observe increased robustness against transposed input.

Current state-of-the-art approaches for singing voice detection build on Recurrent Neural Networks (RNNs). Leglaive et al. [15] trained a bidirectional RNN on mel spectra preprocessed with a highly tuned harmonic/percussive separation stage. They set the state of the art on the public Jamendo dataset [21], albeit using a "shotgun approach" of training 20 variants and picking the one performing best on the test set. Lehner et al. [16] trained an RNN on a set of five high-level features, some of which were designed specifically for the task. They achieve the second best result on Jamendo and also report results on RWC [4, 19], a second public dataset. For perspective, we will compare our results to both of these approaches.

3. BASE METHOD

As a starting point for our experiments, we design a straightforward system applying CNNs on mel spectrograms.

3.1 Feature Extraction

We subsample and downmix the input signal to 22.05 kHz mono and perform a Short-Time Fourier Transform (STFT) with Hann windows, a frame length of 1024 and hop size of 315 samples (yielding 70 frames per second). We discard the phases and apply a mel filterbank with 80 triangular filters from 27.5 Hz to 8 kHz, then logarithmize the magnitudes (after clipping values below 10^{-7}). Finally, we normalize each mel band to zero mean and unit variance over the training set.

3.2 Network architecture

As is customary, our CNN employs three types of feedforward neural network layers: Convolutional layers convolving a stack of 2D inputs with a set of learned 2D kernels, pooling layers subsampling a stack of 2D inputs by taking the maximum over small groups of neighboring pixels, and dense layers flattening the input to a vector and applying a dot product with a learned weight matrix.

Specifically, we apply two 3×3 convolutions of 64 and 32 kernels, respectively, followed by 3×3 non-overlapping max-pooling, two more 3×3 convolutions of 128 and 64 kernels, respectively, another 3×3 pooling stage, two dense layers of 256 and 64 units, respectively, and a final dense layer of a single sigmoidal output unit. Each hidden layer is followed by a $y(x) = \max(x/100, x)$ nonlinearity [1].

The architecture is loosely copied from [11], but scaled down as our datasets are orders of magnitude smaller. It was fixed in advance and not optimized further, as the focus of this work lies on data augmentation.

3.3 Training

Our networks are trained on mel spectrogram excerpts of 115 frames (~1.6 sec) paired with a label denoting the presence of voice in the central frame.

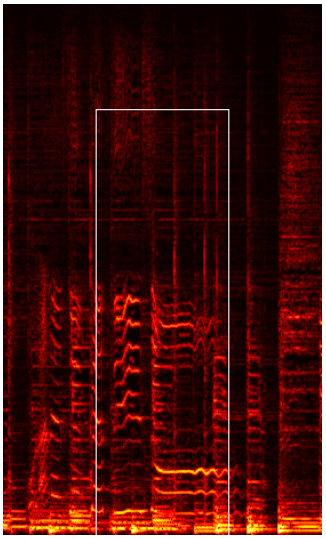
Excerpts are formed with a hop size of 1 frame, resulting in a huge number of training examples. However, these are highly redundant: Many excerpts overlap, and excerpts from different positions in the same music piece often feature the same instruments and vocalists in the same key. Thus, instead of iterating over a full dataset, we train the networks for a fixed number of 40,000 weight updates. While some excerpts are only seen once, this visits each song often enough to learn the variation present in the data. Updates are computed with stochastic gradient descent on cross-entropy error using mini-batches of 32 randomly chosen examples, Nesterov momentum of 0.95, and a learning rate of 0.01 scaled by 0.85 every 2000 updates. Weights are initialized from random orthogonal matrices [22].

For regularization, we set the target values to 0.02 and 0.98 instead of 0 and 1. This avoids driving the output layer weights to larger and larger magnitudes while the network attempts to have the sigmoid output reach its asymptotes for training examples it already got correct [14]. We found this to be a more effective measure against overfitting than L2 weight regularization. As a complementary measure, we apply 50% dropout [7] to the inputs of all dense layers.

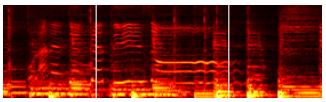
All parameters were determined in initial experiments by monitoring classification accuracy at optimal threshold on validation data, which proved much more reliable than cross-entropy loss or accuracy at a fixed threshold of 0.5.

4. DATA AUGMENTATION

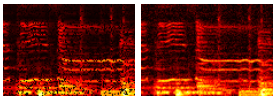
We devised a range of augmentation methods that can be efficiently implemented to work on spectrograms or mel spectrograms: Two are data-independent, four are specific to audio data and one is specific to binary sequence labelling. All of them can be cheaply applied on-the-fly during training (some before, some after the mel-scaling stage) while collecting excerpts for the next mini-batch, and all of them have a single parameter modifying the effect strength we will vary in our experiments.



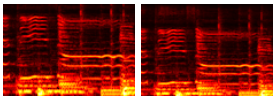
(a) Linear-frequency spectrogram excerpt of 4 sec. The framed part will be mel-scaled and serve as network input.



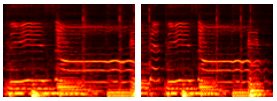
(b) Corresponding mel spectrogram.



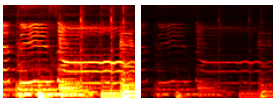
(c) Dropout and Gaussian noise.



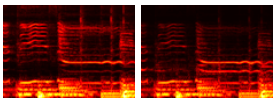
(d) Pitch shift of +/-20%.



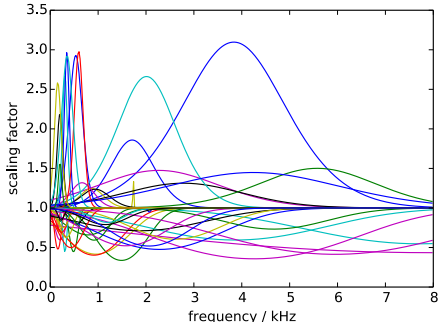
(e) Time stretch of +/-20%.



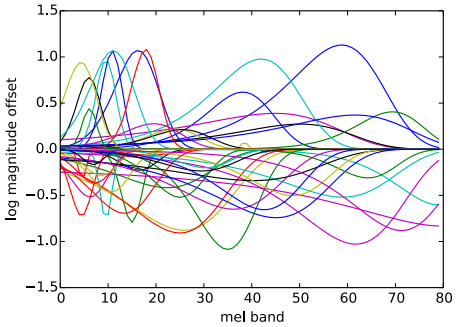
(f) Loudness of +/-10 dB.



(g) Random frequency filters.



(h) Random filter responses of up to 10 dB.



(i) Same filter responses mapped to mel scale.

Figure 1: Illustration of data augmentation methods on spectrograms (0:23–0:27 of "Bucle Paranoideal" by LaBarcaDeSua)

4.1 Data-independent Methods

An obvious way to increase a model's robustness is to corrupt training examples with random noise. We consider **dropout** – setting inputs to zero with a given probability – and additive **Gaussian noise** with a given standard deviation. This is fully independent of the kind of data we have, and we apply it directly to the mel spectrograms fed into the network. Figure 1c shows an example spectrogram excerpt corrupted with 20% dropout and Gaussian noise of $\sigma = 0.2$, respectively.

4.2 Audio-specific Methods

Just like in speech recognition, **pitch shifting** and **time stretching** the audio data by moderate amounts does not change the label for a lot of MIR tasks. We implemented this by scaling linear-frequency spectrogram excerpts vertically (for pitch shifting) or horizontally (for time stretching), then retaining the (fixed-size) bottom central part, so the bottom is always aligned with 0 Hz, and the center is always aligned with the label. Finally, the warped and cropped spectrogram excerpt is mel-scaled, normalized and fed to the network. Figure 1a shows a linear spectrogram excerpt along with the cropping borders, and Figures 1d–e show the resulting mel spectrogram excerpt with different amounts of shifting or stretching. During training, the factor for each example is chosen uniformly at random [1] in a given range such as 80% to 120%, and the width of the range defines the effect strength we can vary.

[1] Choosing factors on a logarithmic scale did not improve results.

A much simpler idea focuses on invariance to **loudness**: We scale linear spectrograms by a random factor in a given decibel range, or, equivalently, add a random offset to log-magnitude mel spectrograms (Figure 1f). Effect strength is controlled by the allowed factor (or offset) range.

As a fourth method, we apply random **frequency filters** to the linear spectrogram. Specifically, we create a filter response as a Gaussian function $f(x) = s \cdot \exp(0.5 \cdot (x - \mu)^2/\sigma^2)$, with μ randomly chosen on a logarithmic scale from 150 Hz to 8 kHz, σ randomly chosen between 5 and 7 semitones, and s randomly chosen in a given range such as -10 dB to 10 dB, the width of the range being varied in our experiments. Figure 1h displays 50 of such filter responses, Figure 1g shows two resulting excerpts. When using this method alone, we map responses to the mel scale, logarithmize them (Figure 1i) and add them to the mel spectrograms to avoid the need for mel-scaling on the fly.

4.3 Task-specific Method

For the detection task considered here, we can easily create additional training examples with known labels by **mixing** two music excerpts together. For simplicity, we only regard the case of blending a given training example A with a randomly chosen negative example B, such that the resulting mix will inherit A's label. Mixes are created from linear spectrograms as $C = (1 - f) \cdot A + f \cdot B$, with f chosen uniformly at random between 0 and 0.5, prior to mel-scaling and normalization, but after any other augmentations. We control the effect strength via the probability of the augmentation being applied to any given example.

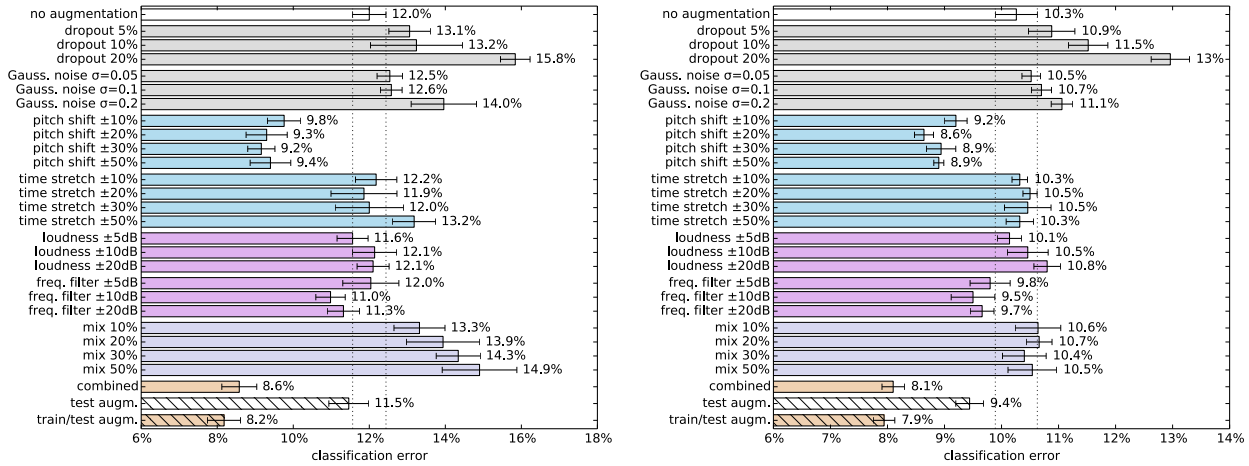


Figure 2: Classification error for different augmentation methods on internal datasets (left: *In-House A*, right: *In-House B*) Bars and whiskers indicate the mean and its 95% confidence interval computed from five repetitions of each experiment.

5. EXPERIMENTAL RESULTS

We first compare the different augmentation methods in isolation at different augmentation strengths on two internal development datasets, to determine how helpful they are and how to parameterize them, and then combine the best methods. In a second set of experiments, we assess the use of augmentation at test time, both for networks trained without and with data augmentation. Finally, we evaluate the best system on two public datasets, comparing against our base system and the state of the art.

5.1 Datasets

In total, we work with four datasets, two of them public:
– *In-House A:* 188 30-second preview snippets from an online music store, covering a very wide range of genres and origins. We use 100 files for training, the remaining ones for evaluation.
– *In-House B:* 149 full-length rock songs. While being far less diverse, this dataset features a lot of electric guitars that share characteristics with singing voice. We use 65 files for training, 10 for validation and 74 for testing.
– *Jamendo:* 93 full-length Creative Commons songs collected and annotated by Ramona et al. [21]. For comparison to existing results, we follow the official split of 61 files for training and only 16 files each for validation and testing.
– *RWC:* The RWC-Pop collection by Goto et al. [4] contains 100 pop songs, with singing voice annotations by Mauch et al. [19]. To compare results to Lehner et al. [16], we use the same 5-fold cross-validation split (personal communication).

Each dataset includes annotations indicating the presence of vocals with sub-second granularity. Except for RWC, datasets do not contain duplicate artists.

5.2 Evaluation

At test time, for each spectrogram excerpt, the network outputs a value between 0 and 1 indicating the probability of voice being present at the center of the excerpt. Feeding maximally overlapping excerpts, we obtain a sequence of 70 predictions per second. Following Lehner et al. [17], we apply a sliding median filter of 800 ms to smoothen the output, then apply a threshold to obtain binary predictions. We compare these predictions to the ground truth labels to obtain the number of true and false positives and negatives, accumulated over all songs in the test set.

While several authors use the F-Score to summarize results, we follow Mauch et al.'s [19] argument that a task with over 50% positive examples is not well-suited for a document retrieval evaluation measure. Instead, we focus on classification error, and also report recall and specifity (recall of the negative class).

5.3 Results on Internal Datasets

In our first set of experiments, we train our network with each of the seven different augmentation methods on each of our two internal datasets, and evaluate it on the (unmodified) test sets. We compare classification errors at the optimal binarization threshold to enable a fair comparison of augmentation methods unaffected by threshold estimation.

Figure 2 depicts our results. The first line gives the result of the base system without any data augmentation. All other lines except for the last three show results with a single data augmentation method at a particular strength.

Corrupting the inputs even with small amounts of noise clearly just diminishes accuracy. Possibly, its regularizing effects [2] only apply to simpler models, as it is not used in recent object recognition systems either [5, 11, 12]. Pitch shifting in a range of $\pm 20\%$ or $\pm 30\%$ gives a significant reduction in classification error of up to 25% relative. It seems to appropriately fill in some gaps in vocal range uncovered by our small training sets. Time stretching does not have a strong effect, indicating that the cues the network picked up are not sensitive to tempo. Similarly, random loudness change does not affect performance. Random frequency filters give a modest improvement, with the

Method	Error	Recall	Spec.
Lehner et al. [16]	10.6%	90.6%	–
Leglaive et al. [15]	8.5%	92.6%	–
Ours w/o augmentation	9.4%	90.8%	90.5%
train augmentation	8.0%	91.4%	92.5%
test augmentation	9.0%	92.0%	90.1%
train/test augmentation	7.7%	90.3%	94.1%

Table 1: Results on Jamendo

Method	Error	Recall	Spec.
Lehner et al. [16]	7.7%	93.4%	–
Ours w/o augmentation	8.2%	92.4%	90.8%
train augmentation	7.4%	93.6%	91.0%
test augmentation	8.2%	93.4%	89.4%
train/test augmentation	7.3%	93.5%	91.6%

Table 2: Results on RWC

best setting at a maximum strength of 10 dB. Mixing in negative examples clearly hurts, but a lot less severely on the second dataset. Presumably this is because the second dataset is a lot more homogeneous, and two rock songs mixed together still form a somewhat realistic example, while excerpts randomly mixed from the first dataset are far from anything in the test set. We hoped this would drive the network to recognize voice irrespectively of the background, but apparently this is too hard or besides the task.

The third from last row in Figure 2 shows performance for combining pitch shifting of $\pm 30\%$, time stretching of $\pm 30\%$ and filtering of ± 10 dB. While error reductions do not add up linearly as in [10], we do observe an additional ~6% relative improvement over pitch shifting alone.

5.4 Test-time Augmentation

In object recognition systems, it is customary to also apply a set of augmentations at test time and aggregate predictions over the different variants [5, 11, 12]. Here, we average network predictions (before temporal smoothing and thresholding) over the original input and pitch-shifted input of -20%, -10%, $+10\%$ and $+20\%$. Unsurprisingly, other augmentations were not helpful at test time: Tempo and loudness changes hardly affected training either, and all remaining methods corrupt data.

The last two rows in Figure 2 show results with this measure when training without data augmentation and our chosen combination, respectively. Test-time augmentation is beneficial independently of train-time augmentation, but increases computational costs of doing predictions.

5.5 Final Results on Public Datasets

To set our results in perspective, we evaluate the base system on the two public datasets, adding our combined train-time augmentation, test-time pitch-shifting, or both. For Jamendo, we optimize the classification threshold on the validation set. For RWC, we simply use the optimal threshold determined on the first internal dataset.

As can be seen in Tables 1–2, on both datasets we slightly improve upon the state of the art. This shows that augmentation did not only help because our base system was a weak starting point, but actually managed to raise the bar. We assume that the methods we compared to would also benefit from data augmentation, possibly surpassing ours.

6. DISCUSSION

We evaluated seven label-preserving audio transformations for their utility as data augmentation methods on music data, using singing voice detection as the benchmark task. Results were mixed: Pitch shifting and random frequency filters brought a considerable improvement, time stretching did not change a lot, but did not seem harmful either, loudness changes were ineffective and the remaining methods even reduced accuracy.

The strong influence of augmentation by pitch shifting, both in training and at test-time, indicates that it would be worthwhile to design the classifier to be more robust to pitch shifting in the first place. For example, this could be achieved by using log-frequency spectrograms and inserting a convolutional layer in the end that spans most of the frequency dimension, but still allows filters to be shifted in a limited range.

Frequency filtering as the second best method deserves closer attention. The scheme we devised is just one of many possibilities, and probably far from optimal. A closer investigation of why it helped might lead to more effective schemes. An open question relating to this is whether augmentation methods should generate (a) realistic examples akin to the test data, (b) variations that are missing from the training and test set, but easy to classify by humans, or (c) corrupted versions that rule out inrobust solutions. For example, it is imaginable that narrow-band filters removing frequency components at random would force a classifier to always take all harmonics into account.

Regarding the task of singing voice detection, better solutions would be reached by training larger CNNs or bagging multiple networks, and faster solutions by extracting the knowledge into smaller models [6]. In addition, adding recurrent connections to the hidden layers might help the network to take into account more context in a light-weight way, allowing to reduce the input (and thus, the dense layer) size by a large margin.

Finally, we expect that data augmentation would prove beneficial for a range of other MIR tasks, especially those operating on a low level.

7. ACKNOWLEDGMENTS

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF): TRP 307-N23, and the Vienna Science and Technology Fund (WWTF): MA14-018. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40 GPU used for this research. Last but not least, we thank Bernhard Lehner for fruitful discussions on singing voice detection.

8. REFERENCES

[1] A. Jannun A. Maas and A. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Int. Conf. on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[2] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural Comput.*, 8(3):643–674, April 1996.

[3] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data Augmentation for Deep Neural Network Acoustic Modeling. In *Proc. of the 2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014.

[4] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical, and jazz music databases. In *Proc. of the 3rd Int. Conf. on Music Information Retrieval (ISMIR)*, pages 287–288, October 2002.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *CoRR*, abs/1502.01852, February 2015.

[6] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. ArXiv:1503.02531, March 2015.

[7] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, July 2012.

[8] Eric J. Humphrey and Juan P. Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Proc. of the 11th Int. Conf. on Machine Learning and Applications (ICMLA)*, 2012.

[9] Navdeep Jaitly and Geoffrey E. Hinton. Vocal tract length perturbation (VTLP) improves speech recognition. In *Int. Conf. on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[10] Naoyuki Kanda, Ryu Takeda, and Yasunari Obuchi. Elastic spectral distortion for low resource speech recognition with deep neural networks. In *Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, 2013.

[11] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the 3rd Int. Conf. on Learning Representations (ICLR)*, May 2015.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, November 1998.

[14] Y. LeCun, L. Bottou, G. Orr, and K. Müller. Efficient BackProp. In G. Orr and Müller K., editors, *Neural Networks: Tricks of the trade*. Springer, 1998.

[15] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *Proc. of the 2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, Brisbane, Australia, April 2015.

[16] Bernhard Lehner, Gerhard Widmer, and Sebastian Böck. A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks. In *Proc. of the 23th European Signal Processing Conf. (EUSIPCO)*, Nice, France, 2015.

[17] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *Proc. of the 2014 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7530–7534, 2014.

[18] Tom LH. Li and Antoni B. Chan. Genre classification and the invariance of MFCC features to key and tempo. In *Proc. of the 17th Int. Conf. on MultiMedia Modeling (MMM)*, Taipei, Taiwan, 2011.

[19] Matthias Mauch, Hiromasa Fujihara, Kazuyoshi Yoshii, and Masataka Goto. Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proc. of the 12th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2011.

[20] Anton Ragni, Kate M. Knill, Shakti P. Rath, and Mark J. F. Gales. Data augmentation for low resource languages. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors, *Proc. of the 15th Annual Conf. of the Int. Speech Communication Association (INTERSPEECH)*, pages 810–814, Singapore, 2014. ISCA.

[21] Mathieu Ramona, Gaël Richard, and Bertrand David. Vocal detection in music with support vector machines. In *Proc. of the 2008 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1885–1888, 2008.

[22] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Int. Conf. on Learning Representations (ICLR)*, 2014.