INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Surgical Mask Detection with Convolutional Neural Networks and Data Augmentations on Spectrograms

Steffen Illium, Robert Müller, Andreas Sedlmeier and Claudia Linnhoff-Popien

Mobile and Distributed Systems Group, LMU Munich
{steffen.illium,robert.mueller,andreas.sedlmeier,linnhoff}@ifi.lmu.de

Abstract

In many fields of research, labeled data-sets are hard to acquire. This is where data augmentation promises to overcome the lack of training data in the context of neural network engineering and classification tasks. The idea here is to reduce model over-fitting to the feature distribution of a small under-descriptive training data-set. We try to evaluate such data augmentation techniques to gather insights in the performance boost they provide for several convolutional neural networks on mel-spectrogram representations of audio data. We show the impact of data augmentation on the binary classification task of surgical mask detection in samples of human voice (*ComParE Challenge 2020*). Also we consider four varying architectures to account for augmentation robustness. Results show that most of the baselines given by *ComParE* are outperformed.

Index Terms: Binary Classification, Data Augmentation, Audio Processing, Mel-Spectrograms, Machine Learning, Convolutional Networks

1. Introduction

In relation to the wide variation of the human voice as well as its differences in expression through loudness, speed and tone, one can image the vast amount of variation present in this data distribution. In comparison, the data-set provided in context of *ComParE Challenge 2020* of about *10H* of spoken human voice, is a rather small sample out of this total distribution. We investigate methods of data augmentation to overcome this disadvantage by enhancing the performance of four neural network architectures. Our motivation to evaluate convolutional models, is the advantage of available choices in data augmentations (raw audio + image representation) and their demonstrated performance of recent years. Recent research on audio classification has shown the legitimacy of processing mel spectrograms extracted from raw audio signals.

The structure of this paper is as follows: Related work (section 2) presents automatic classification of audio segments by neural networks and strategies of data augmentation. Then our methods of augmentation strategies and model architectures are described in 3. We then picture the challenge data-set of *ComParE* in section 4 and describe our experiment implementation in section 5. Results are presented in section 6, which is followed by a discussion in section 7.

2. Related Work

2.1. Audio Classification

The work at hand can be placed in the field of audio classification which comprises a wide variety of different tasks like multi-label classification to predict notes in musical recordings [1], predicting genre tags for songs [2], environmental sound classification [3], or acoustic scene classification [4], in which a single label has to be predicted for an entire audio clip.

In recent years, research tackling such audio classification problems has largely shifted from using manually constructed features, e.g. based on mel-frequency cepstral coefficients (MFCC), to using deep neural networks (DNN) in order to automatically learn task-relevant features. In such approaches of audio classification (when using DNN), it is possible to differentiate further based on the type of input data that is used: i) learning from spectrogram features or ii) direct end-to-end learning from raw audio, i.e. wave-forms.

Learning from spectrogram features, i.e. representing the audio input using time-frequency representations, it is possible to directly leverage architectures that were initially developed for image data processing, such as convolutional neural networks (CNN). The performance of such for classifying environmental and urban sound clips using log-scaled mel-spectrograms as input is for example investigated in [3]. The authors show that a deep convolutional model outperforms classical methods that rely on manually constructed features. [5] also compute log-scaled mel-spectrograms before evaluating the performance of different DNN-architectures in a video soundtrack classification task.

By contrast, an investigation into the ability of CNN to learn useful features directly from raw audio signals, for an automatic tagging task is presented in [2]. Results show that while it is possible to learn directly from raw audio, networks using spectrogram-based input outperform those using raw audio as input. Although [6] show that it is possible for end-to-end learning approaches to match the performance of CNNs using log-mel spectrograms, the authors only achieve this by using very deep networks, which consequently require a lot of computational power.

An interesting argument that it is not only the training procedure, but the architecture of CNNs by itself that is an important element in the quest for learning good feature representations is presented in [7]. By evaluating the performance of untrained, i.e. randomly weighted networks, the authors are able to show that excluding any optimization steps, these networks are able to reach impressive accuracy on different audio classification tasks, outperforming an MFCC baseline.

In the work at hand, we chose to follow the route of first extracting mel-spectrograms and using these as input for CNN model architectures.

2.2. Audio Data Augmentation

The idea of data augmentation in the field of audio processing and more specific classification tasks is far from new. [8] for example introduce a combination of vocal tract length perturbation (VTLP), speed perturbation as well as a shift in tempo. Perturbations in speed were found to be the most helpful aug-

http://dx.doi.org/10.21437/Interspeech.2020-1692

mentation approach.

Further techniques such as masking or *SpecAugmentation* [9]), which, at first, seem to introduce random shapes of zeros in mel-spectrograms, mask certain areas of a given maximal size of frequency bands and temporal bins. In combination with time warping, it was shown that is is possible to overcome model over-fitting by this kind of data augmentation.

In [10] data augmentation is added to enhance music genre classification: Loudness, noise introduction time stretching and pitch shifting are applied before CNN models learn the classification tasks.

Research has shown that data augmentation is a valid strategy in the field of machine learning. We follow this route by implementing and evaluating some promising augmentation techniques to overcome the over-fitting problem of the given dataset of *ComParE 2019 challenge* [11].

3. Methods

3.1. Data Augmentation

There are five different augmentation approaches we consider for audio data augmentation to enhance prediction performance of the binary classification task given:

Speed augmentation
This augmentation processes either speeds up or slows down an audio recording [8]. First we randomly select a starting point $s \sim \mathcal{U}(0, T)$ where T is the number of samples in the recording. Then we sample a window length $w \sim \mathcal{U}(0, 0.4)$. The speed of the samples in the range $[s, \lfloor w * T \rfloor]$ is subsequently adjusted according to $a \sim \mathcal{U}(0.7, 1.7)$. While implementations that interpolate data-points in mel-spectrograms exist, we apply our implementation to raw audio data.

Loudness Augmentation We further diversify the data-set by adjusting the loudness (intensity) of recordings. A loudness factor $l \sim \mathcal{U}(0, 0.4)$ is sampled. This determines how much of the original signal is added back to the original sample ($S + S \times l$, where S is a sample). We want loud signals to get even louder, not just to raise all values of a training sample.

Shift Augmentation
To ensure temporal offset, the mel-spectrogram is shifted to either the left or right by $shift \sim \mathcal{U}(0, 0.4)$ percent of its length in the time dimension. The direction of the shift is determined by $direction \sim Bernoulli(0.5)$. Resulting empty data-points s are filled with $zero$. There is also the possibility for Gaussian noise as fill value.

Noise Augmentation
Random Gaussian noise, $noise \sim \mathcal{N}(0, 0.4)$, is added ($S + S \times noise$) to the mel-spectrogram as it was shown to improve the robustness of end-to-end trained neural networks models. Research has shown the benefits not only in audio classification (where recordings can be noise) but also in image processing (were photos can be noisy in low light environments, too).

SpecAugment
This procedure describes the *masking* of vertical and horizontal windows with $zero$. Similar to [9], we first determine a random starting point, $s \sim U(0, T)$, then we determine the size of the window by $w \sim \mathcal{U}(0, 0.2)$. All data-points $sinw$ are then filled with $zero$. This procedure is applied for both axes (time and frequency). Again, there is the possibility for Gaussian noise as fill value.

We analyze these five data augmentation methods as presented in Figure 1g, individually and in combination.

3.2. Model Architectures

For our experiments we determined five promising convolution based deep neural network models as classifiers. For the sake of comparison, we choose the same default parameter settings for all the models, if not specified otherwise. Please note the varying size and introduction of additional parameters through changes in network architecture.

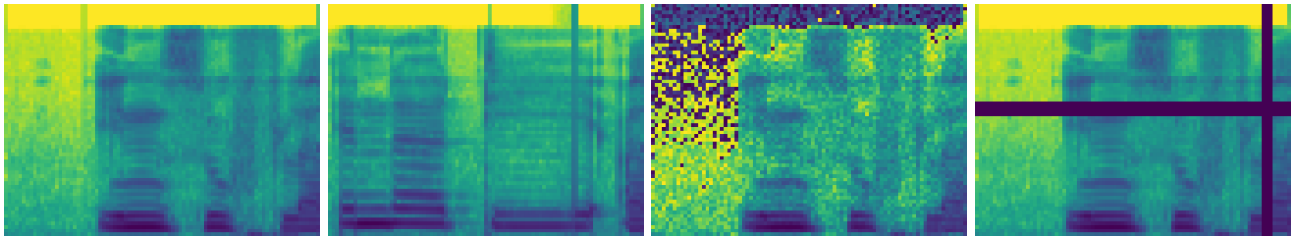
DefaultNetworkConfiguration
For our experiments we compose four 3×3 (*kernel_size*) subsequent convolution operations with [32, 64, 128, 64] *filters*, respectively. A striding of two is applied instead of the commonly used max-pooling. Additional *zero-padding* of 2×2 helps to keep the last tensors' shape at sufficient size. The last convolution is followed by four linear layers of different sizes [128, 256, 128, 1], respectively. There is a *dropout rate* of 0.2 before all operations with trainable parameters (convolutions, fully connected linear layer). Every layer is *leaky-ReLU* [12] activated while tensors are *batch-normalized* [13]. The last single neuron layer represents an exception as it is *Sigmoid* activated for the binary classification task. The classifier output is then evaluated against given training labels by a *BinaryCrossEntropy-Loss* (BCE).

ConvClassifier (CC)
This rather classic model Figure 2a combines state of the art strategies in image classification in the field of machine learning. Four convolution stages reduce the image's spatial resolution while learning to activate feature maps that are growing in size. Additional fully connected layers (or linear) learn the main features within the training samples. This models follows the architecture that was proposed as our *default network configuration*.

SubSpectralNet (SSN)
In reference to [14] we train a combination of four different models based on our *default architecture*. Each model is applied to a different non-overlapping number of mel-bands $= 64/4$, depicted in Figure 2d. Those concatenated band-wise predictions (Sigmoid activated prediction, not the logit) are then processed by a global classifier sub-network (four fully-connected layers with [128, 256, 128, 1] neurons, respectively). Further parameters are the same as in our *default configuration*, this includes activation, batch-normalization and position and rate of dropout. All Sigmoid classifier outputs (band-wise predictions + sub-network prediction) are individually evaluated, each by a BCE-loss. Gradients for model training are finally calculated based on the mean of these losses.

SubSpectralClassifier (SSC)
In addition to the SSN model architecture we define another sub-spectral network. This time the global classifier sub-network has to learn which features to use. The default convolution stack is applied in a band-wise fashion (like before) while the global classification sub-network processes the concatenated output of all last convolution operation, pictured in Figure 2e. Neurons for the fully-connected classifier are as before [128,256,128,1] while the very last layer is Sigmoid activated. BCE-Loss is used for classifier output evaluation.

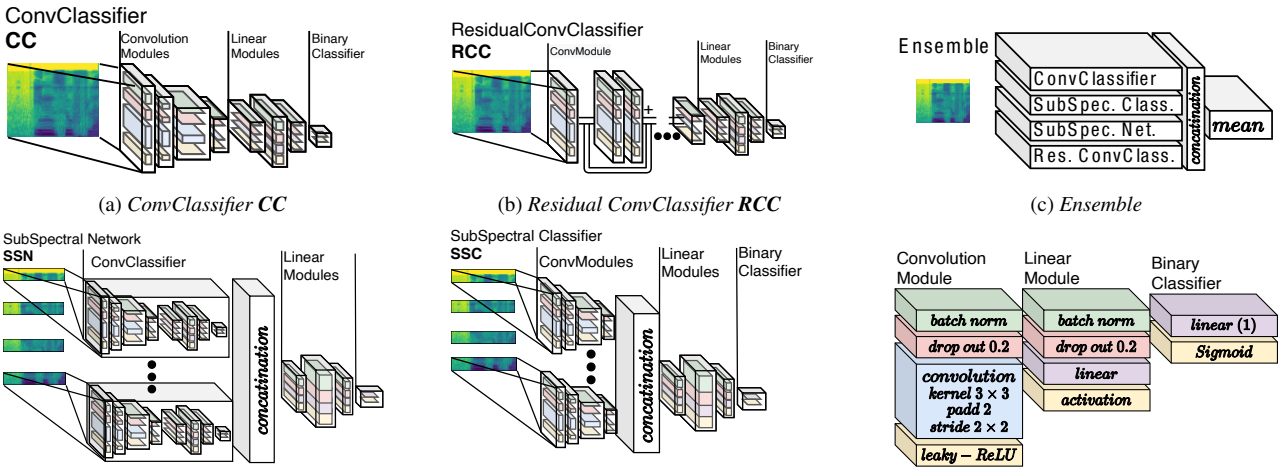
ResidualConvClassifier (RCC)
Similar to [15, 16] we combine residual skip connections around blocks of convolutional operations (Figure 2b). For this, we modify the *default architecture* by introducing a residual block of two similar shaped convolution operations in between every single convolution operations respectively. The fully-connected classifier follows our *default architecture*.



(a) *Raw Mel-Spectrogram* (b) *Speed Variation* (c) *Gaussian Noise Injection* (d) *Masking (SpecAugment)*

(e) *Temporal Shift* (f) *Loudness* (g) *All Combined Augmentations*

Figure 1: *Examples for transformed mel-spectrograms, that are used in model training as given in section 3.*



(a) *ConvClassifier* ***CC*** (b) *Residual ConvClassifier* ***RCC*** (c) *Ensemble*

(d) *SubSpectral Network* ***SSN*** (e) *SubSpectral Classifier* ***SSC*** (f) *Module Block Description*

Figure 2: *Model architectures as introduced and described in section 3. If not stated otherwise, model*

4. Dataset

We focused on the augmentation of audio recordings, included in the Mask Augsburg Speech Corpus (MASC) as given as part of *ComParE 2020 Challenge* [11]. The original dataset consists of 10 h 9 min 14 sec of audio recordings. There are a total of 32 speakers (German native, 16 f, 16 m, age from 20 to 41 years, mean age 25.6 years ± 4.5), which are performing different tasks with and without wearing a surgical mask. It is stated that all audio samples were recorded in a sound-proof audio studio with proper equipment at 44kHz which have been down-sampled and converted to 16 kHz and mono/16 bit. The recordings were pre-segmented into chunks of 1 sec duration without overlap. As motivated in section 1 and proposed in section 3 we transform the raw audio data regarding speed of various intensity and length. Hence, the amount of training data samples quadrupled, whereas the validation and test data-sets both stay untouched. Please note that at least one raw (not transformed) sample of the initial training data is always maintained.

5. Experiments

Our experiments are conducted under controlled settings. Models have been trained on the same training data as we used only fixed seed random operations (along python, numpy, pytorch). Data augmentation is performed as proposed in section 3. We are especially inspired by the implementation of *nlpaug* [17]. Parameters for data augmentation are chosen through the experience of various experiments, as follows: speed factor $= 0.7$, speed ratio $= 0.3$, masking ratio $= 0.2$, noise ratio $= 0.4$, shift ratio $= 0.3$, loudness ratio $= 0.4$. They stay the same throughout all combination of models and seeds in training (no transformation is applied in validation). Mel-spectrograms are extracted at a window hop-length=256 at n-fft=512 transformations for n-mels=64 bands. This results in a sample shape of 64×87. All samples are then locally normalized (zero mean, unit variance).

Data processing and augmentation is performed as follows: First, we transform the train dataset by applying the *speed aug-*

Table 1: *Result comparison: We present model performance under the influence of a variety of data augmentation strategies.*

	ConvClassifier (CC)	SubSpectral Network (SSN)	SubSpectral Classifier (SSC)	Residual Conv-Classifier (RCC)	**Max.**
Raw	64.71 ± 0.30	63.54 ± 0.26	65.82 ± 0.39	64.53 ± 0.53	65.82
Speed	64.71 ± 0.40	63.54 ± 0.91	66.40 ± 0.34	64.07 ± 0.77	66.40
Noise	63.62 ± 0.22	61.35 ± 0.67	64.48 ± 0.46	63.38 ± 0.41	64.48
Loudness	64.31 ± 0.43	62.87 ± 0.64	65.65 ± 0.50	64.14 ± 0.37	65.65
Shift	65.40 ± 0.55	66.31 ± 0.70	**68.20** ± 0.44	64.38 ± 0.78	**68.20**
Masking	64.27 ± 0.34	62.27 ± 0.33	65.10 ± 0.32	64.30 ± 0.45	65.10
Combined	65.03 ± 0.41	63.49 ± 0.79	66.35 ± 0.34	64.12 ± 0.42	66.35
Max.	65.40	66.31	**68.20**	64.38	
Model Parameters	633,219	1,801,621	1,533,321	1,095,171	

mentation as it needs to be applied on the raw audio source. We then extract log mel-spectrogram from the original audio sample (by *librosa*[1]) for training as well as for validation data, which is computationally expensive. For this reason, computation of spectrograms is performed once per training (seed), rather than per epoch. Spectrograms are then transformed into log scale, inverted (dark=low energy) and stored as single channel 8-bit grey-scale image (values between 0-255). All further augmentations are applied in real-time, which is done before local normalization takes place. The dataset is randomly sampled and batched for the models.

Training procedure is applied as follows: We trained each model on five different seeds by back-propagation through *Adam* optimizer [18] at a learning-rate of $lr = 1e-4$, weight-decay $=$ for 51 epochs (146 batches per epoch, batch-size of 200). Losses are calculated as given in section 3 by BCE as a binary classification task. All scores are then measured by the *unweighted average recall* (UAR) (cf. Equation 1 as required by the rules of *ComParE 2020 Challenge* [11]. It is also known as the *Balanced Error Rate* (BER) [19].

$$UAR = 0.5 \times \left(\frac{TP}{FN + TP} + \frac{TN}{TN + FP} \right) \qquad (1)$$

6. Results

Table 2: *Comparison between different models. Performance is measured in terms of the UAR.*

Baseline Models	**UAR**	
	Dev	**Test**
DeepSpectrum + SVM (ResNet50)	63.4	70.8
S2SAE + SVM (Fused)	64.4	66.6
ComParE functionals + SVM (C=10^{-3})	62.3	67.8
ComParE BoAW + SVM (N=2k)	64.2	67.7
Fusion	-	71.8
Our Models		
ConvClassifier + Shift	65.4	-
SpectralNet Network + Shift	66.3	-
SpectralNet Classifier + Shift	68.2	71.5
Residual Conv. Classifier + Shift	64.5	-
Ensemble over best Models; Mean Vote	68.0	-

[1]https://librosa.github.io/ - 10.5281/zenodo.3606573

In Table 1 we present the influence of different augmentation techniques as well as their combinations in contrast to model training on raw, non-augmented data-sets. We also look at different model results for those augmentation methods, at varying trainable parameter sizes. The maximum reached *UAR-scores* (cf. section 5) over all model instances as well as its variance are reported to a total of 100. We found it surprising to see that a conventional CNN (3×3 kernel applied non-overlapping $stride = 2$) on normalized mel-spectrograms already reached a score which could compete with the baseline UAR score (devel.). Data augmentation just minimally enhanced these results by about 1%. Temporal Shift was found to be the overall best and most reliable augmentation strategy, compared to the other domain specific augmentation methods implemented. The proposed SubSpectral Classifier (SSC) (inspired by SSN [14]) achieved promising results on raw samples, which could be further amplified through augmentation by temporal shift. All models are robust to the changes we applied to training data and stayed within their range of performance (varying up to 5%). The wild combination of available augmentations did not prevail, which is rather not surprising. The Residual ConvClassifier (RCC) seems to be more stable as a model itself as data augmentation did not influence the model performance noticeably.

7. Conclusion

In this work, we demonstrated the influence of data-augmentation on the binary classification task of surgical-mask detection in context of the *ComParE Challenge 2020* [11]. We not only implemented and evaluated different augmentation methods, but also showed their influence on different model architectures. We found the combination of temporal shift with standard CNN architectures to be a competitive strategy. Further, our proposed SubSpectral Classifier (SSC) achieved better results, while performing at a similar variance in evaluation. The implemented residual network (RCC) was the overall most robust architecture, as any influence of data-augmentation was not measurable. Not even when combining all available perturbations.

In the future, we aim to evaluate a combination of models, especially with SubSpectral variants. Model ensembles (depicted in Figure 2c) showed promising results compared to the given baselines (cf. Table 2), but were not able to prevail against the SSC model in mean or majority voting.

8. References

[1] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning features of music from scratch," 2016.

[2] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6964–6968.

[3] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2015, pp. 1–6.

[4] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "Cp-jku submissions for dcase-2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[5] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," 2016.

[6] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," 2016.

[7] J. Pons and X. Serra, "Randomly weighted cnns for (music) audio classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 336–340.

[8] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] D. S. Park, Y. Zhang, C. Chiu, Y. Chen, B. Li, W. Chan, Q. V. Le, and Y. Wu, "Specaugment on large scale datasets," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6879–6883.

[10] R. L. Aguiar, Y. M. G. Costa, and C. N. Silla, "Exploring data augmentation to improve music genre classification with convnets," in *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018, pp. 1–8.

[11] B. W. Schuller, A. Batliner, C. Bergler, E.-M. Messner, A. Hamilton, S. Amiriparian, A. Baird, G. Rizos, M. Schmitt, L. Stappen, H. Baumeister, A. D. MacIntyre, and S. Hantke, "The INTERSPEECH 2020 Computational Paralinguistics Challenge: Elderly emotion, Breathing & Masks," in *Proceedings of Interspeech*, Shanghai, China, September 2020, p. 5 pages, to appear.

[12] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1, 2013, p. 3.

[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[14] S. S. R. Phaye, E. Benetos, and Y. Wang, "Subspectralnet–using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 825–829.

[15] L. Ford, H. Tang, F. Grondin, and J. Glass, "A deep residual network for large-scale acoustic scene analysis," *Proc. Interspeech 2019*, pp. 2568–2572, 2019.

[16] Y. Ren, D. Liu, Q. Xiong, J. Fu, and L. Wang, "Spec-resnet: a general audio steganalysis scheme based on deep residual network of spectrogram," *arXiv preprint arXiv:1901.06838*, 2019.

[17] E. Ma, "Nlp augmentation," https://github.com/makcedward/nlpaug, 2019.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] A. Rosenberg, "Classifying skewed data: Importance weighting to optimize average recall," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.