



Visual Transformers for Primates Classification and Covid Detection

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

Mobile and Distributed Systems Group,
LMU Munich

{steffen.illum, robert.mueller, andreas.sedlmeier, linnhoff}@ifi.lmu.de

Abstract

We apply the vision transformer, a deep machine learning model build around the attention mechanism, on mel-spectrogram representations of raw audio recordings. When adding mel-based data augmentation techniques and sample-weighting, we achieve comparable performance on both (PRS and CCS challenge) tasks of ComParE21, outperforming most single model baselines. We further introduce overlapping vertical patching and evaluate the influence of parameter configurations.

Index Terms: audio classification, attention, mel-spectrogram, unbalanced data-sets, computational paralinguistics

1. Introduction

As raw audio can meaningful be represented by mel-spectrograms, the field of audio classification has seen a lot of influences from the field of computer vision. Such algorithm work mostly out of the box (once a suitable hyper-parameter combination is found) and build a competitive baseline for many problems. Currently the most used architecture from the range of deep-learning algorithms is the convolutional neural network (CNN) [1]. When additional written text is provided alongside raw audio, algorithms from the field of natural language processing (NLP) take over. Resulting embeddings from both types of algorithms can subsequently be combined before entering a joint classifier. Most recently, transformers, employing deep-learning attention mechanism, are taking over the field of NLP. The VisionTransformer (ViT) [2] recently fused both of the fields by interpreting images as sequences of patches and thus utilising what can be called dynamic convolution. We relate to this idea by training the ViT on mel-spectrogram representations (thus single channel images) of raw audio data, which is a novelty. We further observe the performance impact by varying the shape and order of patches to match the natural temporal dependency in spectrograms. Additionally, we incorporated data-augmentation techniques while over-sampling the datasets to overcome limitations in two of the challenge datasets of ComParE21 [3].

2. Dataset and Task Description

We focus on two of the provided datasets of ComParE21 [3], one binary and one multi-class classification task:

Primates Sub-Challenge (PRS): Zwerts et al. [4] describes the Primate Vocalisations Corpus, which consists of non-invasive acoustic recordings from Central Africa Mefou Primate Sanctuary to identify and count species. This is a multi-class classification task including the classes: ['background', 'chimpanze', 'geunon', 'mandrille', 'redcap'], which are either primate species or background noises from the natural primate habitat. The *background* class is ambiguous as there is the possibility of faint primate scream mixing. For further introduc-

tory explanation please refer to [3, 4]. When working with the dataset, we noticed an imbalance across the classes, in the train- (Figure 1a) as well as in the provided *devel*-dataset. The counts per sample and class are similar in *train* & *devel* (c.p. Figure 1a), but we noticed slight variations in audio-sample length (number of frames, c.p. Figure 1b & 1c). Distributions for *train* and *devel* are comparable while the test dataset has variations in sample length regarding the number of very small (≤ 0.3 seconds). Class *geunon* e.g., not only is underrepresented but also comes at small audio sample lengths.

The COVID-19 Cough Sub-Challenge (CCS): Brown et al. [5] used an application (Android and Web) to gather (crowd-source) coughs and breathing from *negative* and *positive* tested humans *COVID-19* (binary classification task). For further introductory explanation please refer to [3, 5]. This binary dataset not only is imbalanced, also the number of samples is rather small (215p vs. 71n, c.p. Figure 1d), especially in respect to usual deep learning datasets. Please notice that samples are five times as long as compared to PRS-challenge (up to 16 minutes). When comparing the sample length distribution (c.p. Figure 1e & 1f) we notice slight differences in comparison of *devel* and *test*.

Scores are measured by the *unweighted average recall* (UAR, also known as the *Balanced Error Rate* (BER) [6]):

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

3. Related Work

3.1. Audio Classification

Research in the field of audio classification can be categorized based on the respective goal which is to be achieved, e.g., genre prediction [7] or classification [8] (multi-class classification approaches), as well as binary classification or anomaly detection.

As is the case for most research fields concerned with classification tasks, progress in recent years was mostly achieved using deep neural networks (DNN) and gradient based learning techniques. By contrast to earlier research, features are no longer manually constructed (as was the case e.g., for work based on mel-frequency cepstral coefficients (MFCC)) but instead automatically inferred (e.g. VGG [9]). Consequently, these approaches can be differentiated based on how the input data is structured: i) direct usage of the mel-spectrograms, i.e., the time-frequency representation or ii) end-to-end learning from the audio's wave-forms.

When using spectrogram features, it is possible to apply and build upon network architectures that were conceived to tackle problems related to image data. Research has shown that these approaches outperform classical methods relying on manually constructed features [8, 10]. While [7] has shown that it is possible to construct end-to-end learning approaches directly using the audio's wave-form, their method's performance is worse

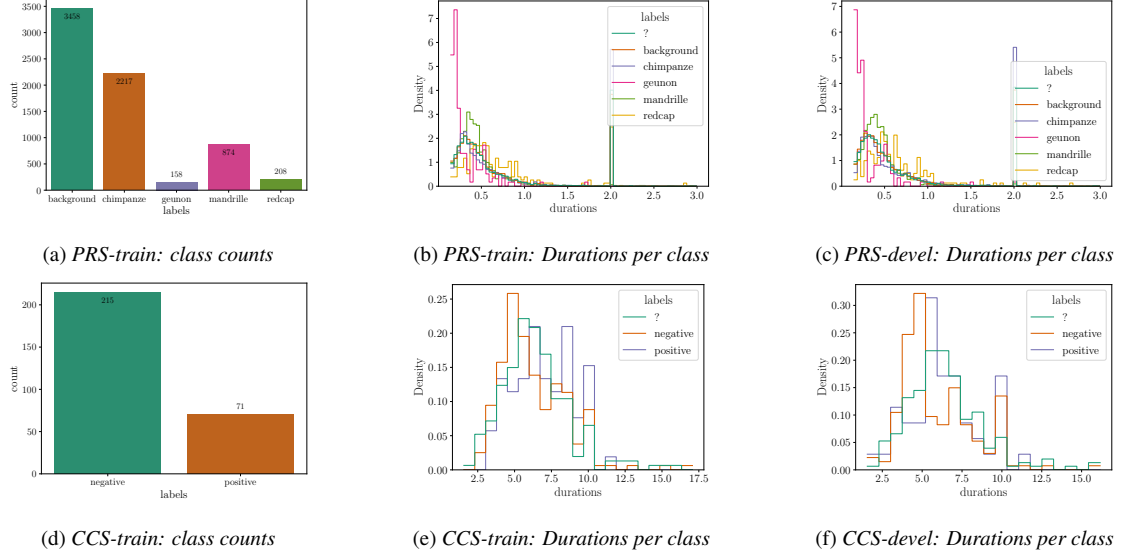


Figure 1: Statistics for 'primates' and 'CCS' datasets. 1a and 1d show the counts per class for each training set, 1b, 1c 1e & 1f show the distribution of file-durations per class. For comparison, the test set duration (label: ?) is also shown.

compared to spectrogram-based approaches. [11] was finally able to match the performance of CNN based architectures using log-mel spectrograms by applying very deep networks.

Only recently, it has been shown that transformer architectures [12] can be a competitive approach to CNNs in the image domain. Initially developed in natural language processing (NLP), the transformer architecture aimed to improve on complex sequence models like recurrent convolutional neural networks. Still, the transformer follows the encoder-decoder structure as is common for state-of-the-art sequence models based around the attention mechanism by mapping a sequence of input features x to a sequence of representations z . The so-called Vision Transformer (ViT) [2] showed that the architecture is able to achieve competitive results to convolutional neural networks by interpreting an image as a sequence of patches.

In the work at hand, we analyze the performance of such a ViT architecture when applied to audio data by using the spectrogram representation of the data as input features as well as the importance of its hyper-parameters.

3.2. Audio Data Augmentation

To improve results in audio classification tasks, data augmentation techniques are commonly applied. One such approach is masking or *SpecAugmentation* [13], which masks areas (which can be restricted by a maximum size) of certain frequency bands and temporal bins by replacing them with random shapes of zeros. When combined with time warping, research has shown that these techniques can reduce over-fitting in training. Results in [14] show that these and other approaches like loudness modifications, time shifting, or the addition of noise improve classification results of CNN based models. In the work at hand, we build upon these results by implementing and evaluating augmentation techniques on the given dataset of the ComParE21 [3].

4. Methods

Based on the observations from previous section, we decide to apply data augmentation as presented in [14] while oversampling the datasets to re-balance the deficits in data distribution per class and length. Oversampling was assumed to work best at equally distributed classes, so that the total number of samples (S_{tot}) $S_{tot} = \max(\text{len}(S_{class})) \times n_{classes}$, we then sampled randomly with replacements given the probability $p = 1/\text{len}(S_{class})$ for each sample and class respectively. This is a critical point. Without equal sampling, no training was possible for both challenges with the algorithm introduced in Subsection 4.2. The audio samples were further cropped to smaller length. We let the HP-search find the best matching sample length for each dataset and algorithm. This results in shorter sample length (~ 0.4) for PRS and longer samples (~ 1.2) for CCS dataset. This chapter now briefly revisits the data-augmentation techniques, then introducing the deep learning models.

4.1. Mel-Spectrogram Augmentation

Named parameters in this section are treated as model hyper-parameter and further tuned by the tree-structured Parzen Estimator-algorithm (TPE) hyper-parameter search [15]. We searched a meaningful parameter-space in 400runs per algorithm (200training-epochs) and dataset. For all augmentation every draw from a random distribution is performed per sample draw.

Shift Augmentation: Temporal offset is provided, to account for positional variations within the the mel-spectrograms. Data-samples are shifted either left or right (*direction* $\sim \text{Bernoulli}(0.5)$) by *shift* $\sim \mathcal{U}(0, \text{shift.ratio})$ percent of axis length in the temporal dimension. Resulting zero-value data-points s are left empty.

Noise Augmentation: Random Gaussian noise, *noise* $\sim \mathcal{N}(0, \text{noise.ratio})$, is added ($S + S \times \text{noise}$) to the mel-spectrogram to improve robustness and generalisation capabili-

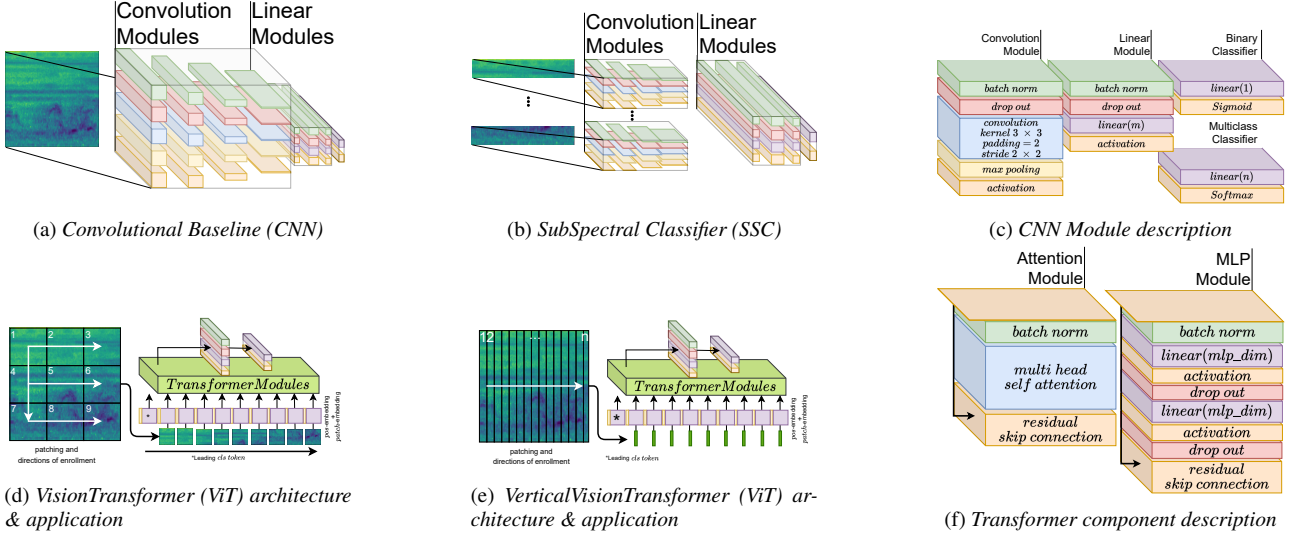


Figure 2: Model architectures as introduced and described in section 4. Hyper-parameters such as number of neurons or layer depth are determined per model and dataset in a broad hyper-parameter search.

ties of end-to-end trained neural networks models. This method has already been successfully applied in tasks of audio classification (where recordings can be noise) and image processing.

SpecAugment: As presented by [16] we randomly *mask* vertical and horizontal windows (mel- and temporal-axis) with *zero* value (all data-points *sinw*). First a random starting point, $s \sim U(0, T)$, is determined. Then we draw the size of the window by $w \sim U(0, \text{mask_ratio})$.

Loudness Augmentation: We adjust the loudness (intensity) of recording by a loudness factor. $l \sim U(0, \text{loudness_ratio})$ determines how much of the original signal is added to the original sample ($S + S \times l$). Loud signals get even louder instead just raising all values (which would be normalised anyway).

4.2. Model Architectures

In our experiments we trained and applied three different architectures in an end to end fashion, by *AdamW* [17] as optimiser for gradient based back-propagation. For comparison, we choose a similar parameter range where possible settings for all the models, if not specified otherwise. Please note the varying size and introduction of additional parameters through changes in network architecture. We further tuned the model hyper-parameters as described above by TPE [15].

For both tasks (PRS and CCS) we define default parameters as follows, PRS: $n_logits = 5$, $loss = ce_loss$, CCS: $n_logits = 1$, $loss = bce_loss$. Further model parameters are the same to reduce the influence of a wide hyper-parameter space. Those are dropout = 0.2, *GELU* [18]

CNNBaseline (CNN) As deep-learning baseline, we implement a quite regular convolutional neural network (CNN Figure 2a). Four blocks of batch-normalisation [19], dropout, convolution max-pooling and a activation function are stacked. Filter sizes are $\{32, 64, 128, 64\}$ while convolutional-kernel sizes are 3×3 . with zero-padding = 1 we kept the shape of each activation matrix. Pooling kernel sizes are 2×2 , cutting the resolution in half, finally. The last convolution is followed by a linear module composed of batch-normalisation, dropout, fully-connected layer (128 neurons) and an activation function. The final fully

connected classifier is implemented as described above and dependent on the task.

SubSpectralClassifier (SSC) We consider another deep-learning baseline as comparison for the ViT algorithms. The SubSpectralClassifier (SSC) is implemented as proposed by [14]. In short; four small convolutional neural networks are trained on different non-overlapping of mel-bands $n_mels = 128/8$, depicted in Figure 2b. Those concatenated band-wise embeddings are then processed by a classifier sub-network (three fully-connected layers with $[128, 64, n]$ neurons, respectively). We switched the ReLU activation function in favour of the more recent GELU activation function. Further parameters are the same as in [14], this includes batch-normalisation and position and rate of dropout.

VisionTransformer (ViT) In reference to [2] we implement the ViT as stack of blocks of multi-head self-attention (scaled dot-product attention) followed by double layered mlps (GELU activated fully-connected layers with dropout, c.p. Figure 2d & 2f), both wrapped with residual skip-connections [20]. Multi-head attention runs multiple self-attention operations in parallel, then subsequently fusing the individual embeddings into a single d/n_heads sized embedding. This style of architecture was first seen in form of the *Transformer* neural-network [21] which relates a single sentence (sequences of word embedding) to it self.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{n}}\right)V$$

$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \dots; \text{head}_n]W^O$$

$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

An additional linear module is attached before evaluating the resulting embedding by the subsequent classifier (c.p. Figure 2c).

VerticalVisionTransformer (ViT) We strictly follow the ViT design described above but realise overlapping full-height (mel-dimension) patches to analyse the mel-spectrogram in an more intuitive fashion. Since mel-spectrograms posses a temporal relation, we move the kernel ($n_mels \times \text{patch_size}, \text{stride} = 1$) horizontally along the temporal axis rather then in a rows-then-column pattern (c.p. Fig-

ure 2d vs. 2e). The width of the overlapping patches was empirically found to work best between 5 and 9 pixels. This procedure results in more and larger patches (in comparison to the original ViT) while respecting the natural order of the audio/mel-spectrogram domain.

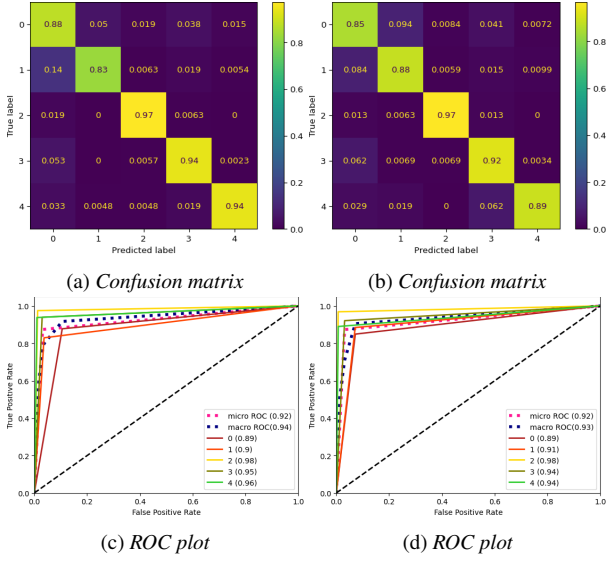


Figure 3: Additional Evaluation Metrics for ViT (3a,3c) & VViT (3b,3d) on PRS-dataset. For confusion matrices, rows sum up to 1. Class assignments are: {0: background, 1: chimpanze, 2: geunon, 3: mandrille, 4: redcap}

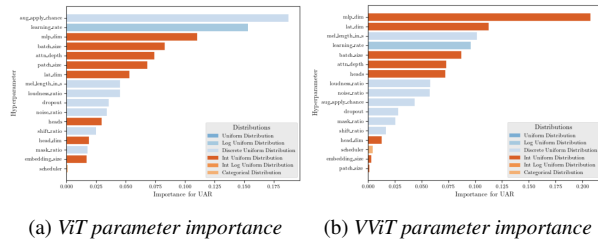


Figure 4: ViT (4a) & VViT (4b) hyper-parameter importance on PRS dataset measured by fANOVA[22]

5. Results

We train the described algorithms for 200 epochs on the over-sampled and augmented *train* datasets. No sample from *devel* or *test* is touched in training while seeds (0, ..., 19) are fixed along models and runs. Hyper-parameters are tuned by TPE as stated above per network architecture (CNN, SSC, ViT & VViT) (c.p.section 4).

Evaluations based on *devel* show, that most of the provided baselines of this years classification tasks of *ComParE 2021* are mostly out-performed by even the simplest CNN architectures (CNN & SSC), with *OpenXBOW* and *AUDeep* beeing the strongest competitors. Depending on the task, the *Transformer*-style approaches (ViT & VViT) then overtake both (ComParE and our CNN baselines) by a margin (Figure 5), outperforming the *devel* dataset performance measured in UAR (c.p., Eq. 2). Model performance on *test* seemed to stagnate in most cases.

For PRS we observe high rate of confusion between *background* (0) and *chimpanze* (1) on both attention based models (c.p. Figure 3). We noticed the same behaviour for

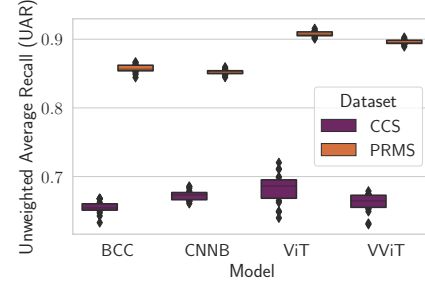


Figure 5: Model performance measured in UAR over 20 seeds

CNN & SSC. Presumably more chimpanzees are mixed faintly into the background class. We therefore suggest further steps of data-processing. For CCS we noticed a high false negative rate (up to 60%) along all models, which is very bad in respect to a usage of medical applicability. Usually a high specificity over a high sensitivity is required.

We further analysed the hyper-parameter importance for both attention based models (ViT & VViT, c.p. Figure 4). While both models do not care whether a learning rate scheduler (*lr_scheduler* in [None, $lr = n^{epoch}, n \sim \mathcal{U}(0.88, 1)$]) is present, the initial given *lr* and total available parameters of fully connected layers (*mlp_dim* & *lat_dim*) are important for both models. Usually transformer models are known to work best, employing a specific training sequence, including warm-up-phases and sophisticated schedulers. In both tasks, we could not confirm these practices.

On one hand, we are surprised by the low importance of both *embedding_size* & *head_dim* for both models, as these parameters control how much of information can be evaluated at every single layer in parallel (the lower the initial projection dimension, the higher the initial information compression rate). On the other hand, we find it interesting to see how the importance of data augmentation not only depends on the used datasets, but on the model architecture as well. Even when they differ very little.

We further observe a reduced overall performance of VViT (at lower variance) which boils down to the assumption, that spatial, non overlapping regions are more important for both tasks, than a bigger picture, which processes information on all mel bands at once.

In future we plan to extend our view to more recent models featuring attention, e.g. the *Performer*[23] which approximates the result of the regular attention mechanism, without explicitly constructing the quadratic-sized attention matrix.

Table 1: Our experimental results compared to provided baselines from *ComParE21* [3].

Dataset	primates		coughing	
Model	Devel	Test	Devel	Test
Deep Spectrum	81.3	78.8	63.3	64.1
End2You	72.70	70.8	61.8	64.7
OpenSMILE	82.4	82.2	61.4	65.5
OpenXBOW	83.3	83.9	64.7	72.9
AUDeep	84.6	86.1	67.6	67.6
Fusion	-	87.5	-	73.9
ours CNN	85.9	-	68.6	67.4
ours SSC	86.6	84.2	66.8	72.0
ours ViT	91.5	88.3	72.0	69.9
ours VViT	90.3	87.2	67.8	68.6

6. References

- [1] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.
- [2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.
- [3] B. W. Schuller, A. Batliner, C. Bergler, C. Mascolo, J. Han, I. Lefter, H. Kaya, S. Amiriparian, A. Baird, L. Stappen, S. Ottl, M. Gerczuk, P. Tzirakis, C. Brown, J. Chauhan, A. Grammenos, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, M. R. Leon J. J. Zwerts, J. Treep, and C. Kaandorp, "The INTERSPEECH 2021 Computational Paralinguistics Challenge: COVID-19 Cough, COVID-19 Speech, Escalation & Primates," in *Proceedings INTERSPEECH 2021, 22nd Annual Conference of the International Speech Communication Association*. Brno, Czechia: ISCA, September 2021, to appear.
- [4] J. A. Zwerts, J. Treep, C. S. Kaandorp, F. Meewis, A. C. Koot, and H. Kaya, "Introducing a central african primate vocalisation dataset for automated species classification," *arXiv preprint arXiv:2101.10390*, 2021.
- [5] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo, "Exploring automatic diagnosis of covid-19 from crowdsourced respiratory sound data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3474–3484.
- [6] A. Rosenberg, "Classifying skewed data: Importance weighting to optimize average recall," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.
- [7] 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.
- [8] 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.
- [9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [10] 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.
- [11] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," 2016.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.
- [13] 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.
- [14] S. Illium, R. Müller, A. Sedlmeier, and C. Linnhoff-Popien, "Surgical mask detection with convolutional neural networks and data augmentations on spectrograms," *Proc. Interspeech 2020*, pp. 2052–2056, 2020.
- [15] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.
- [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] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.
- [18] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.
- [19] 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.
- [20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.
- [22] F. Hutter, H. Hoos, and K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance," in *Proceedings of the 31st International Conference on Machine Learning*, ser. *Proceedings of Machine Learning Research*, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Beijing, China: PMLR, 22–24 Jun 2014, pp. 754–762. [Online]. Available: <http://proceedings.mlr.press/v32/hutter14.html>
- [23] K. Choromanski, V. Likhoshesterov, D. Dohan, X. Song, A. Kane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser *et al.*, "Rethinking attention with performers," *arXiv e-prints*, pp. arXiv–2009, 2020.