Acoustic Identification of Flat Spots On Wheels Using Different Machine Learning Techniques

Gabriel Dernbach[1], Athanasios Lykartsis[1], Leon Sievers[2], Stefan Weinzierl[1]
[1] *TU Berlin, Fachgebiet Audiokommunikation, 10587 Berlin, Deutschland*
[2] *Railwatch GmbH, 53177 Bonn, Deutschland*
stefan.weinzierl@tu-berlin.de

Introduction

The continuous, non-invasive monitoring of machines and machine-related services can help ensure trouble-free operation and relieve the provider of unnecessary, manual routine controls. Audio recordings in particular require no further modifications of the devices or installations to be monitored and are thus especially convenient to acquire. We consider the case of the acoustic detection of flat spots, a sign of wear on the wheels of rail vehicles.

The task can be considered as a special case of acoustic scene classification, assigning one of several acoustic event categories to a given audio recording. Acoustic scene classification has traditionally been addressed by extraction of hand-crafted audio features and forwarded to a general classification algorithm such as a support vector machine (SVM) [1]. Other classical approaches are based on the slightly more general mel-frequency cepstral coefficients (MFCCs) combined with a clustering method (e.g. Gaussian mixture models) to facilitate multi class classification [2]. Recent progress in the field has been stimulated by public data sets and open contests, such as the widely acknowledged DCASE challenge [3]. Over the past years convolutional neural networks in conjunction with log-mel-spectrograms have proven to be promising building blocks in addressing acoustic recognition tasks [4].

We have adapted these methods to the specific requirements of the acoustic detection of flat spots. This damage to the shape of railroad wheels can be caused by slip and slide conditions that causes wheels to lock up while the train is still moving, by faulty brakes or wheelset bearings. It is noticeable acoustically through periodic knocking noises, the frequency of which is determined both by the speed of the train and the diameter of the wheels.

We have compared different feature representation such as raw audio data, MFCCs and log-mel-spectrograms, as well as different classifiers, from a SVM classifier, a standard convolutional network architecture (CNN) to encoder-decoder segmentation networks (U-Net). We have further identified desirable feature invariances and implement the corresponding acoustic transformations.

Our findings suggest that the task is facilitated by resampling the audio to a virtually constant flat spot beating frequency. Furthermore, convolutional encoder-decoder architectures employing spectrogram representations outperform other methods with comparable number of parameters.

Dataset

The data set was provided by Railwatch GmbH, a company responsible for monitoring and reporting faulty train wagons. The data has been recorded at three different sites in close proximity to the rail tracks and displays minor variations in recording distance and large variations in ambient noise. Each recording contains the sound of one full train passing by the recording spot. The duration of a recording varies from 20 seconds up to several minutes, depending on the train speed and the number of wagons it carries. Each individual sample consists of the raw audio file, as well as measurements of the train speed and the radii of individual wheels at their corresponding timestamps. Estimation of speed and wheel radii were based on video recordings and were provided with the dataset. The respective labels have been annotated by experts as they listened to the recordings, indicating a flat spot by marking the corresponding region of time. The data set contains 566 train passings, summing to a joint duration of 7.9 hours (see figure 1).

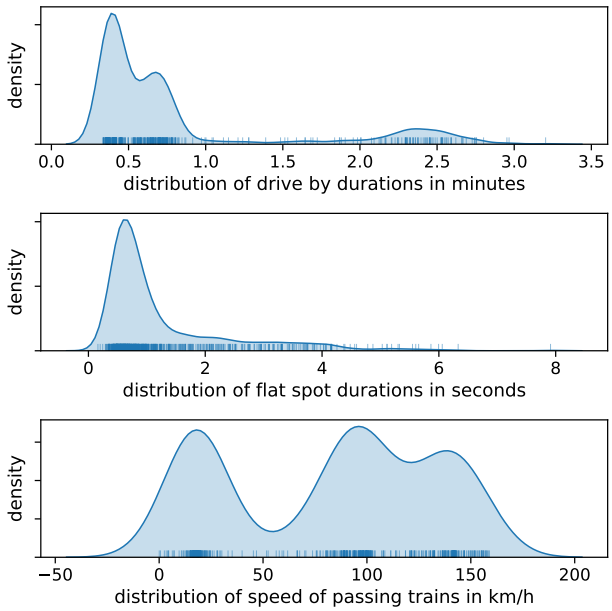

Figure 1: Statistics of the audio data set, indicating the duration of the train passings, the duration of the annotated flat spots, and the speed of the passing trains

279 train passings exhibit at least one flat spot annotation. A total of 765 flat spot regions have been marked, summing to a joint duration of 16.2 minutes. We note the pronounced imbalance of marked to unmarked regions of

3.5%. Most of the marked regions are of short duration, typically between 0.5 to 2 seconds. We observe a wide variability of the speeds per train passing (figure 1) and note that the perceptual quality of a flat spot also varies substantially with speed. For low speeds, the "beating" sound is clearly noticeable as separate, countable hits. For very high speeds the beating resembles an amplitude modulation of a wide-band turbulent noise. In terms of label precision we note that the flat spot labels have considerable variations in how much environmental sound was included before and after the actual sound of the flat spot.

Representation

We consider the three most common representations in audio event detection analysis community, namely raw audio, log-mel spectrograms and MFCCs. These show an increasing degree of compression, and therefore increasing associated inductive bias. All extraction was based on the original audio with a 48000 Hz sampling rate, being subsampled to 8192 Hz and cut into non overlapping frames of 2 or 5 seconds. For the raw audio data, no further processing is applied. For the log mel spectrograms, we apply a short-time Fourier transform (STFT) with a window size of 512 and a hop size of 128 samples, followed by a mel filterbank with 40 filters. Finally each element is being log-compressed with a factor of 7. For the MFCCs we extracted the first 13 coefficients.

During feature extraction, we also include the train speed estimations provided. By standardization to a virtual identical speed we aim to diminish the variance introduced by the train speed to our input representation, so that the detection task becomes more homogeneous and thus easier to solve. Normalization to a common virtual pass-by speed s_c can be performed by scaling the audio playback rate, which is essentially an audio resampling operation. For the speed s_i of a train i, the scaling factor t_i is then given by:

$$t_i = \frac{s_c}{s_i} \tag{1}$$

In order to keep the amount of resampling small we choose s_c to be the median over all s_i. To further refine the normalization with the information about the speed of the train s_i and the radius of the wheel r_i, we can compute the expected frequency of the beating and standardize the individual playback speeds to a common beating frequency b_c, using a scaling factor b_i given by

$$b_i = b_c \frac{2\pi r_i}{s_i} \tag{2}$$

Models

We considered three machine learning models: First, we applied **support vector machines with Gaussian kernel**, a standard method well documented for acoustic scene classification and providing a baseline for small to medium-sized data sets.

Secondly, we used a classical **convolutional neural network** [5]. For the log mel spectrogram as an input to the algorithm, we take five convolutional (2D) and two fully connected blocks applying batch normalization throughout. When working with raw audio we use eight convolutional (1D) inverted residual blocks [6] with squeeze excitation, followed by one fully connected layer. Each convolution block ends with a pooling operation of kernel size three and stride three as presented in SampleCNN [7]. The choice of the inverted residual blocks was based on memory considerations, as we anticipated the use of the network as the encoder backend to the following U-Net architecture. Finally, we employed a **U-Net like convolutional network** [8], i.e., a encoder decoder network of convolutional blocks with additional skip connections between encoder to decoder layers of matching sizes. For the mel-spectrogram representation we took the original 2D design and trimmed its number of filters and depth. The encoder part of the network is then identical to the feature extractor of the mel-CNN. The base U-Net outputs a 2D segmentation mask, we therefore appended a convolutional layer of 1D over frequencies and then average pool. The model thus outputs a 1D segmentation mask corresponding to the flat spot regions to be predicted. For the raw audio representation we employed the SampleCNN as an encoder and built the corresponding mirrored decoder similar to the mel-filtered variant. In literature, the architecture closest to ours is found in Stoller et al. [9].

Model regularization is achieved by weight decay and drop out as well as data augmentations. In particular we apply mixup augmentation [10], which is performed by taking a weighted sum of two randomly selected data points as

$$\tilde{x} = \lambda x_i + (1-\lambda) x_j$$
$$\tilde{y} = \lambda y_i + (1-\lambda) y_j$$

where x_i are the features of item i, y_i its respective targets and $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ the random variable describing the distribution over weighting factors. We choose $\alpha \in [0.1, 0.4]$, as most augmentations are then only slight perturbations of the original samples and 50/50 overlaps are especially rare. We also consider random shifts in pitch [11] but this did not lead to considerable performance gains, as was the case with random addition of low variance Gaussian noise. [1] For the mel spectrogram representation we additionally apply random cut outs of the mel axis (2 %) as well as the time axis (10 %) [12].

All models were trained on a cross entropy loss. A frame had to reported to contain a flat spot when at least 5 % of it's content overlaps with an annotated flat spot region. For two second frames this corresponds to at least 100 ms of flat spot annotation, whereas for five second frames at least 250 ms must be annotated. 250 ms is also the

[1] The signals at hand are rarely of clear pitched harmonic content but more of stochastic and percussive nature. Pitch shift augmentation, although highly regarded in other domains is of limited use here, as tested experimentally.

shortest duration of flat spot annotations and therefore is the hard limit which should not be exceeded. For the segmentation models, we add the criterion of a per sample classification to support the creation of accurate flat spot location masks.

We report the F1 score, i.e., the geometric mean of precision and recall, for classification performance, since classification accuracy is less insightful for settings of high class imbalance.

Results

The data set was set up for three fold cross validation with splits retaining grouping of individual train passings. That is, frames extracted from one passing are only allowed to appear in one and only one of train, develop or test set. Training hyper-parameters such as learning rate, weight decay and drop out were determined during a pre run of randomized search. The training data set was balanced by oversampling the minority class.

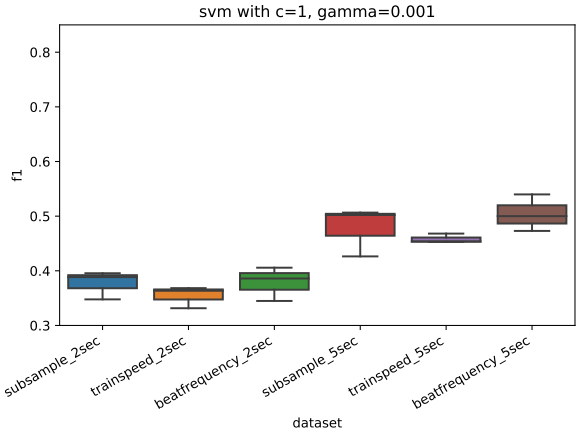

Figure 2: SVM model with MFCCs as input.

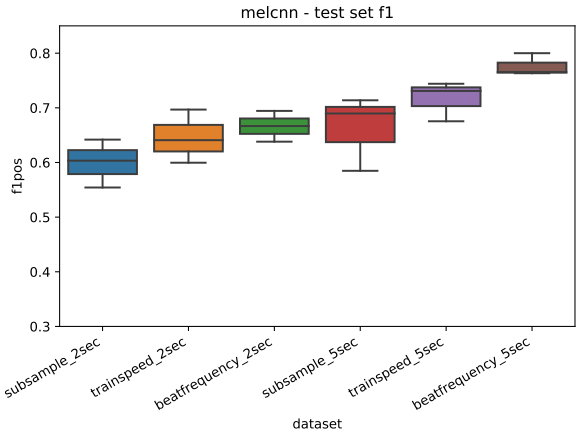

Figure 3: CNN model with mel-spectrogram as input.

The performance of the baseline SVM can be seen in Fig. 2. Its best score of $F1 = 0.50$ was achieved with a beat frequency standardization on frames of five seconds. The best results in total were achieved with models based on the log mel spectrogram reported in Figs. 3 and 4. For the models learned on raw audio we had to increase the capacity significantly to obtain comparable results. The models were scaled up to 23 M parameters, but still remained slightly behind their mel-spectrogram counterparts of 76 K parameters, probably also because of missing corresponding augmentation and regularization techniques, since there is no direct correspondence of spectral cut out augmentations for raw audio data. The SampleCNN achieved its best scores of median 0.63 F1, with a maximum of 0.68 F1. The Sample-UNet achieved it's best scores of median 0.73, with a maximum of 0.75 F1. While the mel-U-Net can be easily trained on a commodity CPU found in most laptops, the sample-U-Net is only feasible to train on modern GPUs.

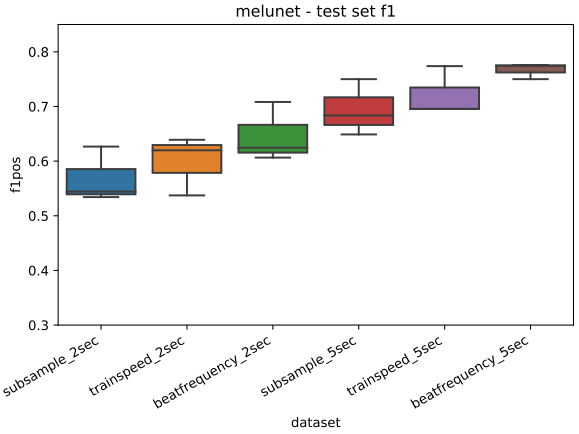

Figure 4: U-Net model with mel-spectrogram as input.

tp	fp	fn	tn	f1	cv split	filters
90	23	37	616	0.75	0	[16, 32, 64]
91	23	30	662	0.77	1	[16, 32, 64]
116	36	31	603	0.78	2	[16, 32, 64]

Table 1: Performance metrics of mel spectrogram unet trained with beat frequency normalization on five seconds frames. The model size amounts to only 72k parameters

There is a strong tendency in 5 s frames providing better results than the 2 s frames, which might be due to the increased context necessary for the solution of the task. However, one should be aware of the possibility of broader windows subsuming several flat spots. A case hard to detect could then hide in a frame with a more prevalent one that triggers classification and would then stop to contribute to the error. A quick inspection of the detailed segmentation masks provided by the melUnet, however, did not indicate such cases. The detailed segmentation mask of the melUnet is a key advantage for the usage of this architecture in practice, since predicting the precise location of a flat spot is more valuable than just detecting it's mere presence.

The normalization to a virtually constant beating frequency turned out useful. With each combination of representation and network it constantly increased performance significantly.

We notice a tendency for the false positives and false negatives to be balanced (Tab. 1). In many applications, however, the economic cost of false positives and false negatives may not be symmetric. This imbalance is best taken into consideration by weighting the respective loss of the classes, but staying with the balanced sampling strategy. This allows untangling modelling of costs (re-weighted loss) and providing proper gradients for optimization (training with balanced sampling).

Small changes in the amount of frequency cut off and log compression showed no considerable variations in F1 Score. Learning rate, dropout and weight decay can vary over wide range for the melccn without degrading performance. The unet architectures are more sensitive to optimal training parameterization. Replacing the transposed convolution in the upsampling paths of the melUnet with a simple linear upsampling did not degrade performance although being much cheaper to compute. Modest increases in the width and depth of the melUnet did not improve performance. Already small models can fit the training data set very well and thus are in need of strong regularization. One might interpret this as suggesting a necessary expansion of the training data set. However, when training the models on subsets of varying size we notice a saturation in performance form usage of 70% of the data set onwards. The combination of sufficient model complexity to overfit and the lack of improvement by taking more data into consideration suggests to test for label noise. There is indeed some arbitrary variation to the exact starting and stop times of the marked flat spot regions as well as the difficult per case decision of whether the prevalence of a wheel beating suffices to report a flat spot.

Conclusion

In this study, we presented a system for the identification of wheel flat spots of passing trains based on audio recordings. Different preprocessing techniques and machine learning models have been evaluated. We could show that convolutional segmentation architectures (U-Net) employing mel spectrogram representations outperform other methods with comparable number of parameters. Further our findings suggest that the task is facilitated by resampling the recording to a standardized expected flat spot beating frequency.

Improved performance could be achieved by further investments in labelling. That is, some flat spots are more or less pronounced suggesting the use of soft labels. To speed up labelling and finding relevant unlabeled sections quickly we suggest using hard negative mining. Also, more elaborate data augmentation could be applied, such as including and overlaying samples with characteristic environmental noises.

For future work, we are planning the direct prediction of the faulty wagon axles by aligning the audio and other metadata. A data set that contains annotations of which axle was in fact defective would side step lossy intermediate representations as well as the label noise of manual annotation and could be trained end to end.

Gabriel Dernbach acknowledges partly support by the German Ministry of Education and Research (BMBF) in the project ALICE III (01IS18049B).

References

[1] Geiger, J. T., B. Schuller, and G. Rigoll: Large-scale audio feature extraction and SVM for acoustic scene classification. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 1–4, IEEE, 2013.

[2] Mesaros, A., T. Heittola, and T. Virtanen. TUT database for acoustic scene classification and sound event detection. 24th European Signal Processing Conference (EUSIPCO), pp. 1128–1132, IEEE, 2016.

[3] Mesaros, A., T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley. Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(2), pp. 379–393, 2017.

[4] Valenti, M., M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen. DCASE 2016 acoustic scene classification using convolutional neural networks. Proc. Workshop Detection Classif. Acoust. Scenes Events, pp. 95–99, 2016.

[5] Simonyan, K., and A. Zisserman. Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems. In Advances in neural information processing systems, pp. 568–576, 2014.

[6] Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520, 2018.

[7] Kim, T., Lee, J., and J. Nam. Sample-level CNN architectures for music auto-tagging using raw waveforms. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 366–370. IEEE, 2018.

[8] Ronneberger, O., P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. Springer, Cham, 2015.

[9] Stoller, D., E. Sebastian, and S. Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. arXiv preprint arXiv:1806.03185, 2018.

[10] Zhang, H., M. Cisse, Y. N. Dauphin, and Lopez-Paz, D. Mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412, 2017.

[11] Schlüter, J., and T. Grill. Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In ISMIR, pp. 121–126, 2015.

[12] Park, D. S., W. Chan, Y. Zhang, , C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. arXiv preprint arXiv:1904.08779, 2019.