Robustness of Acoustic Scene Classification using a CNN in a Real-World Scenario

Bjarke B. Madsen, Gergely Kiss, Magnus Borresen and Michail Kampitakis
Email: [bmadse18, gkiss21, mborre15, mkampi21]@student.aau.dk
Department of Electronic Systems, Signal Processing, Aalborg University

Introduction

Acoustic Scene Classification (ASC) can be used for a wide variety of applications regarding intelligent sensing technology [1]. In this case a Convolutional Neural Network (CNN) [2] is used for classifying acoustic scenes in a construction site scenario with high noise level, where it's assumed that the sound is captured through a microphone and is transmitted wirelessly. The output of this process is a class label, information that could be utilized for further process, such as activating specific noise-cancelling algorithms. Previous work has shown that a CNN is the state-of-the-art for ASC [3]. The goal of the project is the following:

- Data acquisition and preprocessing of different acoustic scenes
- Implementation of a CNN model that can classify acquired data
- Testing robustness of model when imposed to different types of noise

Dataset

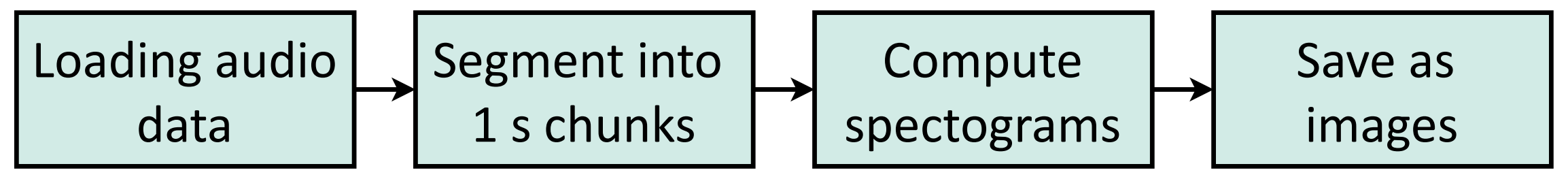

Loading audio data
Segment into 1 s chunks
Compute spectograms
Save as images

Fig. 1: Preprocessing chain

- Dataset recorded at Nyt Aalborg Universitetshospital (NAU)
- Equipment used: *Zoom H4n recorder, Presonus PRM1 microphone, and t.bone MM-1 microphone.*
- Classes recorded:

➤ *Inside* ➤ *Inside vehicle* ➤ *Office* ➤ *Semi outside* ➤ *Outside*

Sample rate of the recordings is 44,1 kHz. The classes represent different types of locations at NAU, where semi-outside is an unfinished building without external walls. Acquired data is preprocessed into images of linear-frequency spectrograms as shown in the preprocessing chain in **fig. 1.** Supplementary audio data is also found in open source databases.

CNN Model

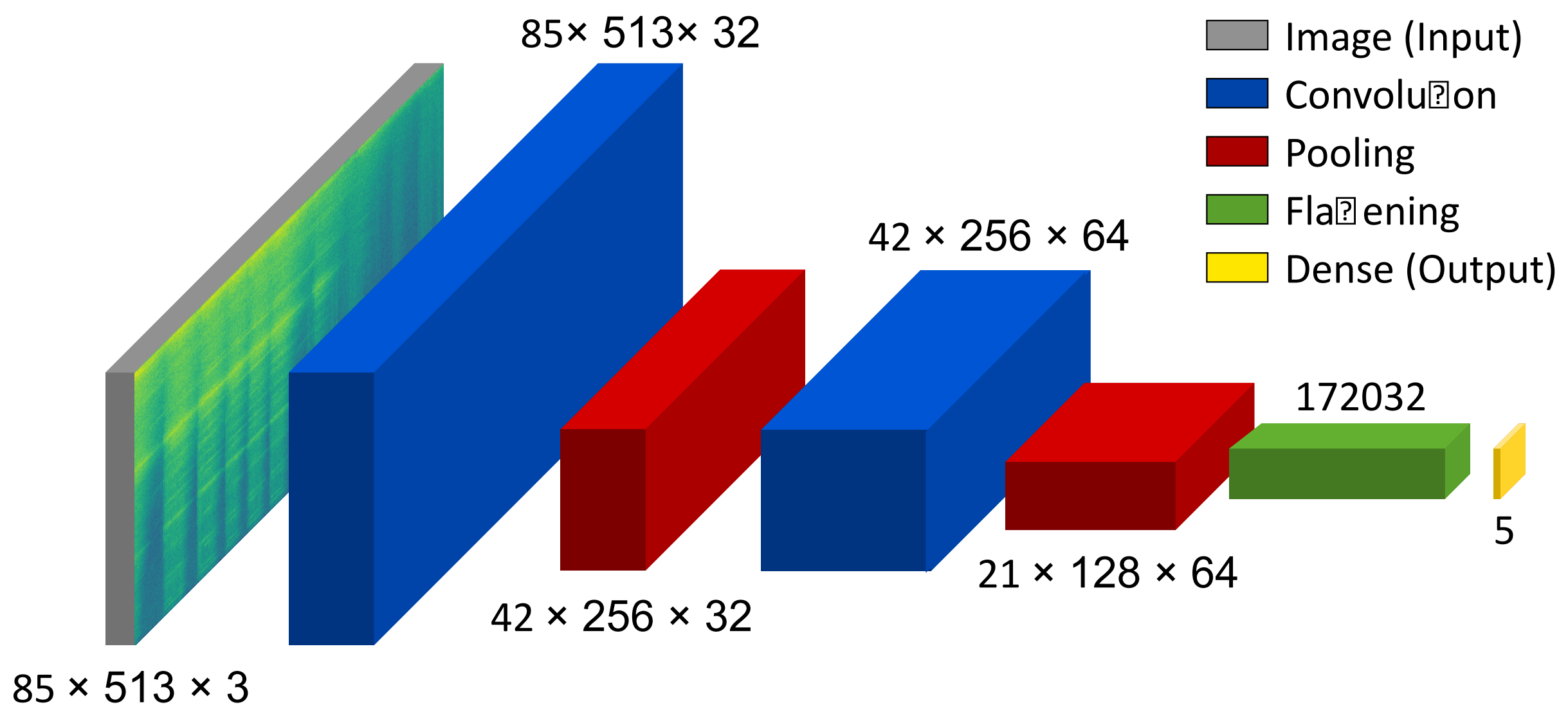


Fig. 2: Implemented CNN model for acoustic scene classification. Input is a spectrogram, output is a label probability distribution of the five classes.

The CNN model is implemented in Python using Keras library [2]. The preprocessed dataset consisting of 54.844 spectral images is split up into training (80%) for the CNN learning process, validation (10%) to evaluate model loss during learning, and test (10%) for final model evaluation. The model input is a spectrogram image. The two convolutional layers are responsible for feature extraction, both utilizing 3 × 3 kernels where the first convolution consist of 32 filters and the second of 64. The convolutional outputs is padded to have the same size as the input, and is followed by max pooling layers that decrease the image size to half. Rectified linear unit (ReLu) is chosen as activation for convolutions and pooling layers. The dense layer output five classes activated by softmax, meaning that the output is a probability distribution of the classes. CNN can be seen on **fig. 2**, trained with categorical cross entropy loss and Adam optimiser [2].

Noise Types

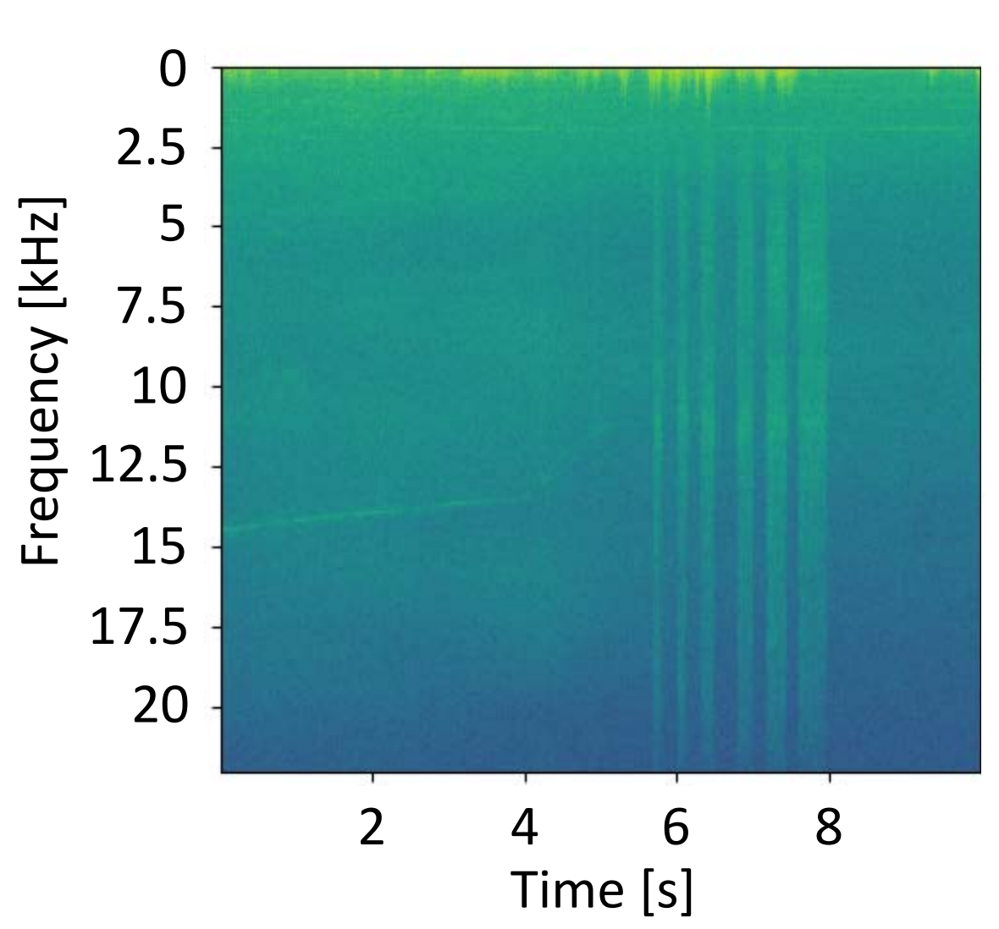


Fig. 3: Wind noise, located at the lowest frequencies.

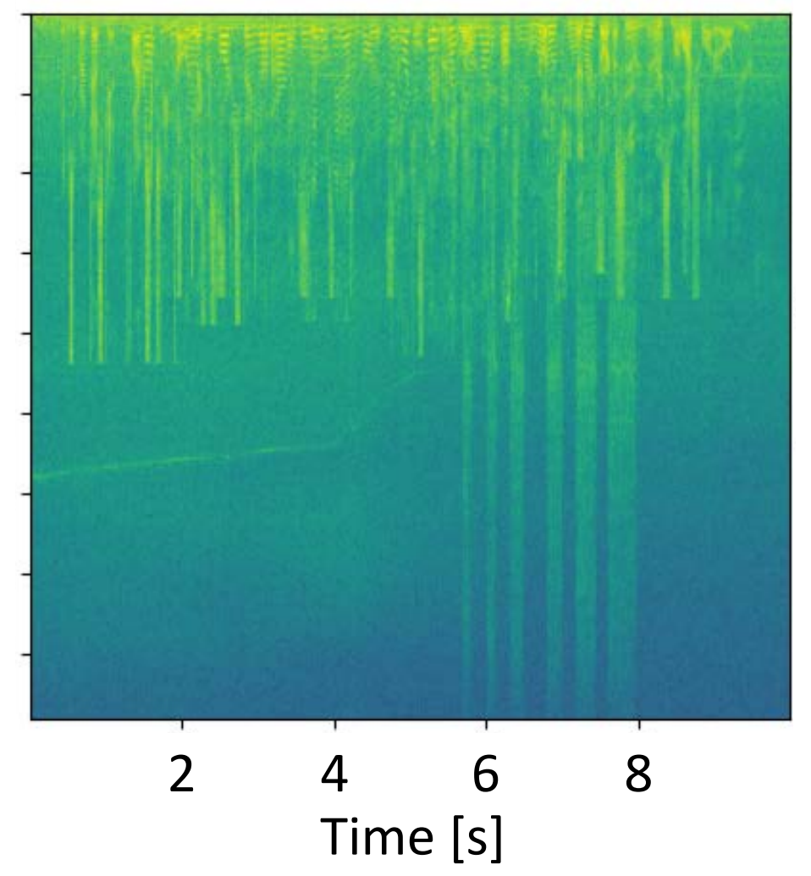


Fig. 4: Speech noise, located from low to around 8 kHz.

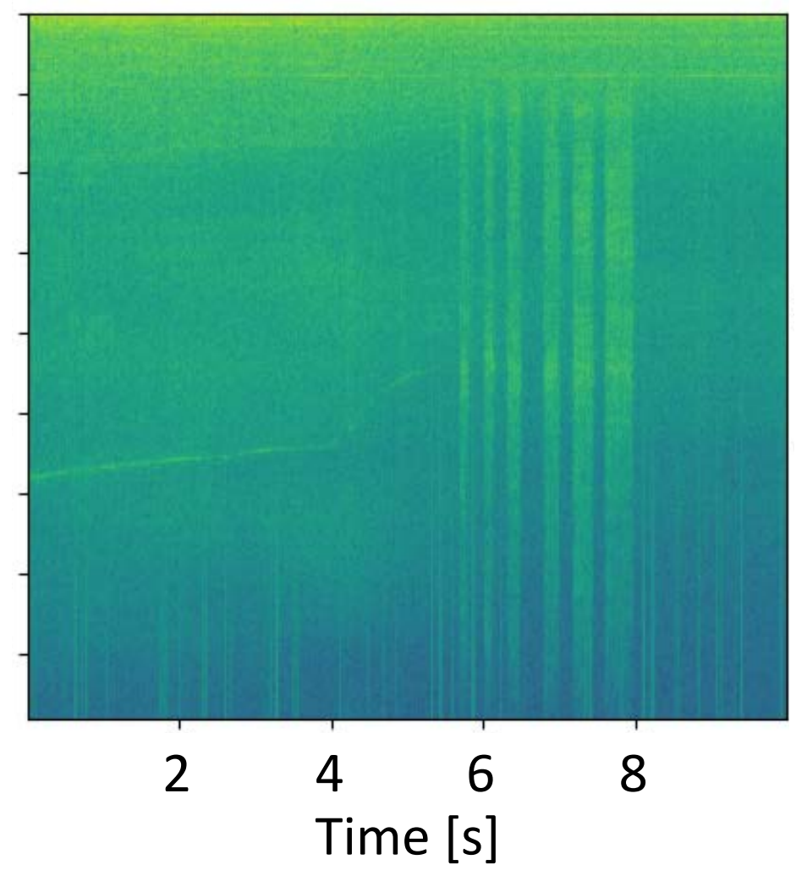


Fig. 5: AWGN and random packet loss noise.

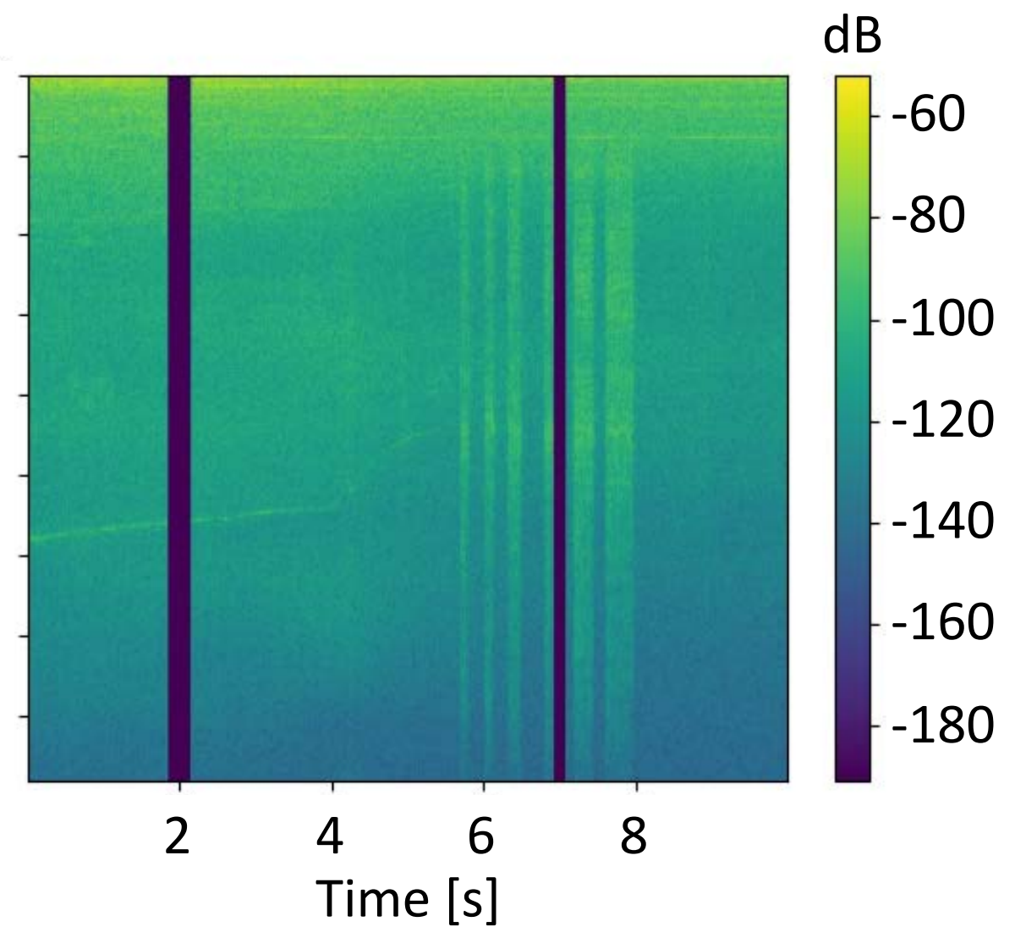


Fig. 6: AWGN and burst packet loss noise.

Noise can alter the quality of a prediction and is therefore imposed to the test data to test the robustness of the model. Different types of noise can occur either during signal recording (Environmental noise), or during wireless transmission (System and channel noise). Spectrograms visualize 10 s of the 'outside' class.

Environmental noise

- **Wind noise, fig. 3:** Loaded sample of wind noise (random started position) added to the signal. This process simulates the altering of recorded audio quality in outdoor situation.
- **Speech noise, fig. 4:** Loaded random sample of recorded voice is added to the signal. Simulates the effect that a talking person near the microphone has to the signal.

System and Channel noise

- **Additive white gaussian noise (AWGN)**: Zero mean normal AWGN is generated and added to the signal. This method will mimic noise imposed from a real-world random process.
- **AWGN and random packet loss, fig. 5:** Is generated with a Bernoulli random process with 5% probability of packet loss. The lost packets are distributed along the spectrogram. This process simulates minor losses that can occur.
- **AWGN and burst packet loss, fig. 6:** Loss of 5% of total packets at one or two random positions. The positions are chosen randomly along the spectrogram and neighbouring packets on these positions are lost. This process simulates longer losses on channels.

Results

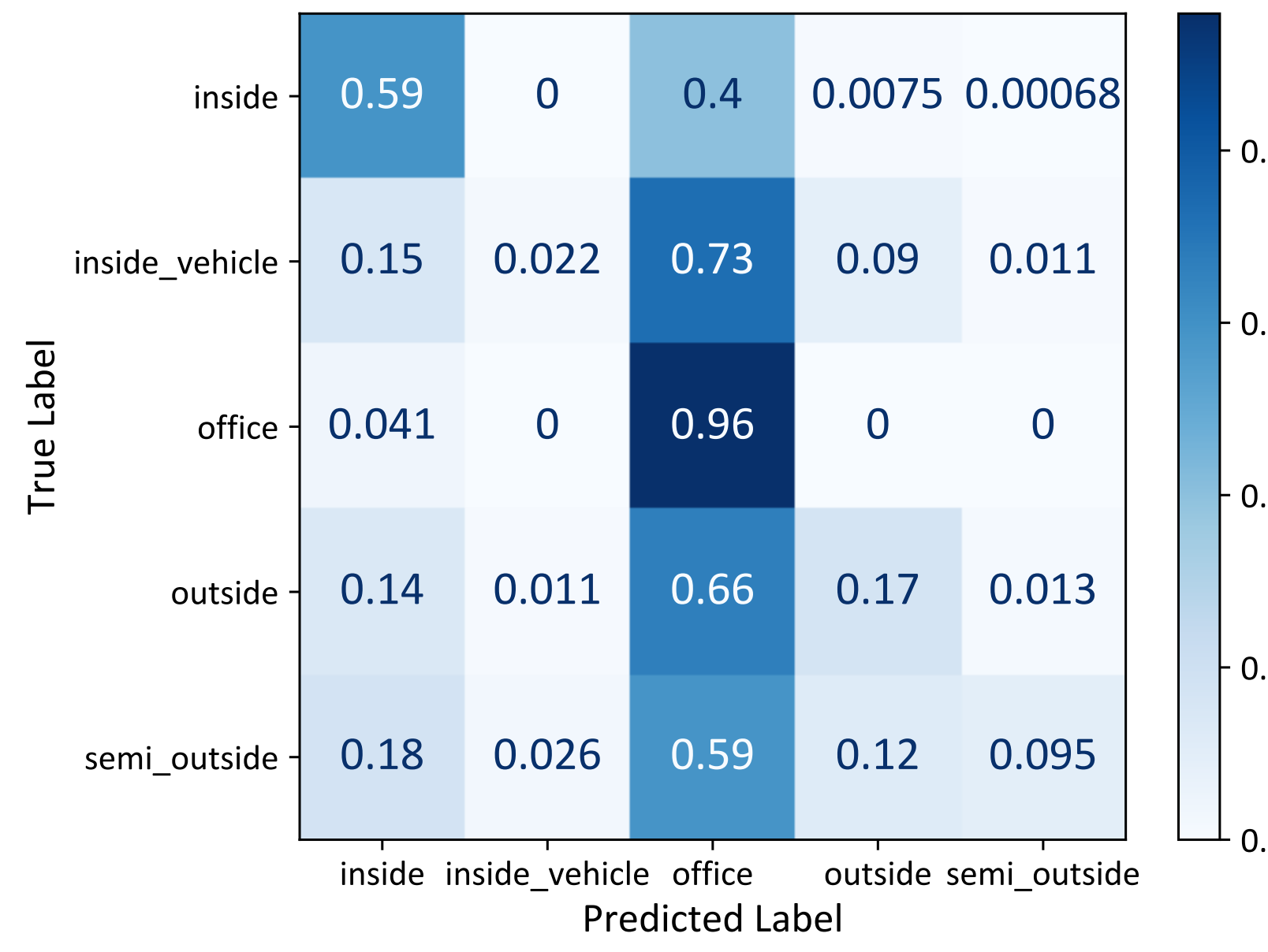
Model	Original	S&C	Wind	Speech	All
5 epochs	94%	67% (89%)	74%	46%	42% (44%)
10 epochs	95%	70% (92%)	78%	49%	44% (47%)

Tab. 1: Classification accuracy with different noise types. Results with random packet loss is shown in parantheses. All includes every noise type.

The model is tested with our own acquired data supplemented by online data. A total of 4648 samples unevenly distributed over the five classes, is tested.

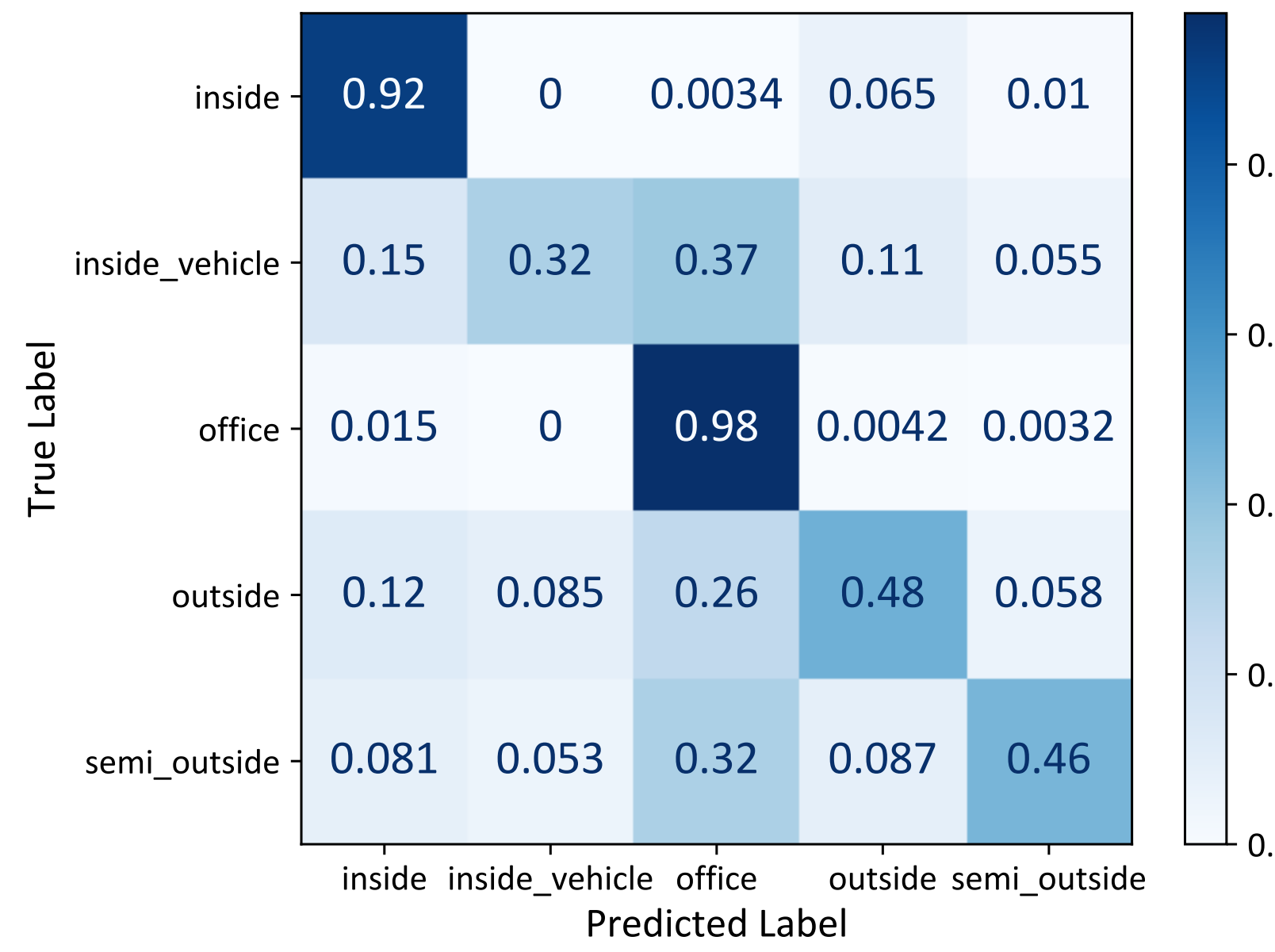
The overall test results are shown in **tab. 1**, where the **original** test data without noise have a descent accuracy. AWGN and a packet loss of 5% has been added in both cases of random and burst method. The model performs fairly well with **system & channel** and **wind**, but it is found that the accuracy decreases with **speech**, or **all** types present. By comparing results of **5** and **10 epochs**, it is found that a training the network with a greater number than 5 epochs does not drastically increase the accuracy.

Normalized confusion matrices has been generated from 10 epoch model tests. In **fig. 7**, the test data are imposed with AWGN, burst packet loss, wind, and speech resulting in 44% accuracy. An example without burst and speech is shown in **fig. 8**, where the test data are imposed with AWGN, random packet loss, and wind resulting in 70% accuracy.



True Label \ Predicted Label	inside	inside_vehicle	office	outside	semi_outside
inside	0.59	0	0.4	0.0075	0.00068
inside_vehicle	0.15	0.022	0.73	0.09	0.011
office	0.041	0	0.96	0	0
outside	0.14	0.011	0.66	0.17	0.013
semi_outside	0.18	0.026	0.59	0.12	0.095

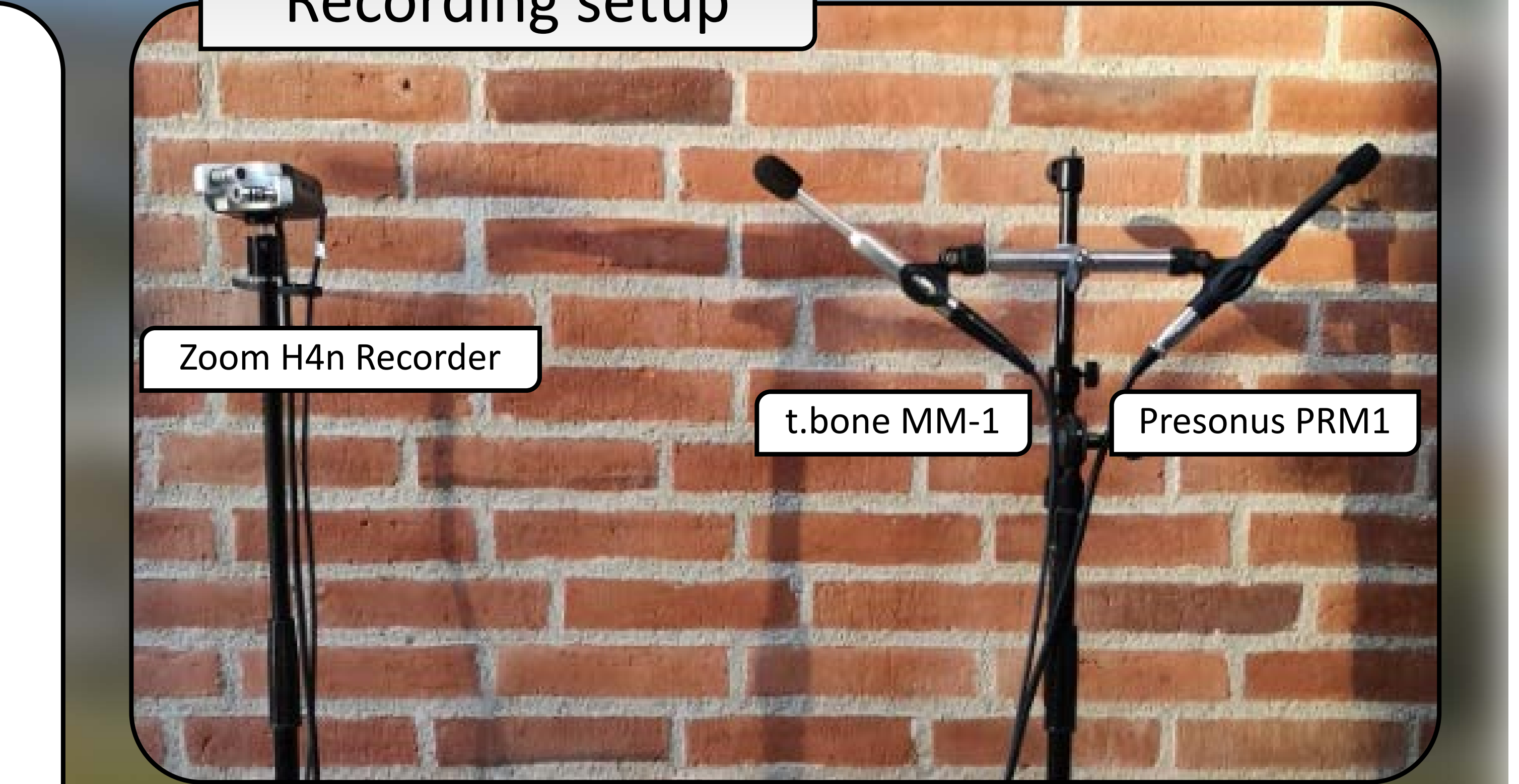
Fig. 7: Normalized confusion matrix of test data with wind, speech, AWGN, and burst packet loss imposed to the signal.



True Label \ Predicted Label	inside	inside_vehicle	office	outside	semi_outside
inside	0.92	0	0.0034	0.065	0.01
inside_vehicle	0.15	0.32	0.37	0.11	0.055
office	0.015	0	0.98	0.0042	0.0032
outside	0.12	0.085	0.26	0.48	0.058
semi_outside	0.081	0.053	0.32	0.087	0.46

Fig. 8: Normalized confusion matrix of test data with wind and random packet loss imposed to the signal.

Recording setup



Discussion

The trained model shows promising results in predicting the acoustic scenes, even with most added noise types. AWGN does not seem to have any effect on predictions. Burst packet loss and wind noise make the prediction accuracy decrease and with wind noise the model predictions are still usable. The model seems to have found features that resembles the characteristics of the scenes, therefore, the training data create limitations on the prediction accuracy when introducing noise. Since only the office scenes had recordings of human speech, a constraint is applied on the model when trying to add speech to all scenes. The method of adding wind noise is debatable since wind noise is not an additive noise type. To correctly represent the effect of wind noise, it should be present at the time of recording. Likewise, power tools seem to show up as features as well, as scenes are often miss-classified with these noise types introduced. To alleviate this issue, more diverse training data should be gathered.

Conclusion

The presented convolutional neural network method of ASC had high classification accuracy when no noise was imposed on the acoustic field. In noisy environment the accuracy decreased, however it still seems to be usable. Data augmentation and higher amount of training data can further improve the model accuracy. The CNN model was specifically built for classifying construction scenes. Further testing of the CNN structure is needed to determine whether the model can be used for other environments.

Acknowledgement

We would like to thank RTX for the project proposal and supplementary project supervision. We would also like to thank Jorton A/S for access to the construction site NAU, and Thomas Vestergaard for a private donation of equipment for this purpose.

References

[1] Tuomas Virtanen, Mark D. Plumbley, and Dan Ellis. *Computational Analysis of Sound Scenes and Events.* ISBN: 3319634496

[2] Francois Chollet. *Deep Learning with Python.* ISBN: 97816127294433

[3] Jacob Abeßer. *"A review of deep learning based methods for acoustic scene classification".* eng. In: *Applied sciences* 10.6 (2020), pp. 2020 - ISSN: 2076-3417

Background photo: Region Nordjylland