# NEURAL NETWORKS WITH DILATED CONVOLUTIONS FOR SOUND EVENT RECOGNITION

**Stepan Miklanek**
Doctoral Degree Programme (2), FEEC BUT
E-mail: xmikla12@stud.feec.vutbr.cz

Supervised by: Jiri Schimmel
E-mail: schimmel@feec.vutbr.cz

**Abstract**: Convolutional neural networks, most commonly deployed in image classification tasks, typically use square-shaped convolutional kernels, which are well suited for feature extraction from two-dimensional data. This study explores the effect of utilizing spectrally aware dilated convolutions specialized for sound event recognition. By extending the base kernels in the time or the frequency dimension, the features extracted from the spectral audio representations should, in theory, better capture the temporal and timbral information of different sound events. The baseline neural network model with squared kernels was compared against three models, which used an increasing dilation factor in the subsequent convolutional layers. The three models were purposefully tuned to focus towards the frequency and time feature extraction. The results have shown that the models with dilated convolutions performed noticeably better in comparison with the baseline model.

**Keywords**: sound event recognition; convolutional neural networks; dilated convolution

## 1 INTRODUCTION

In recent years, numerous studies presented exceptional results of utilizing convolutional neural networks (CNNs) in image classification or object detection tasks. The neural network models specifically tailored for these problems can also be used when dealing with audio [1]. The main drawback of using these models is that they are usually overly complex for tasks such as sound event recognition or music auto-tagging. The CNNs used in image processing are designed to extract spatial features of various objects present in images. Analogically, the majority of audio-motivated CNN architectures are using spectral transformations, which can also be thought of as image-like representations [2]. In image processing, squared convolutional filters of 3×3 or 12×12 are widely used [3]. Note that the audio processing filter dimensions do not correspond to spatial information. The spectrogram-based filter dimensions, in fact, correspond to time and frequency. In the audio realm, wider filters are capable of learning longer temporal dependencies, while filters with larger height are capable of learning timbral features as shown in [4]. The larger frequency and time receptive fields also result in more complex models as the convolutional kernels get bigger. The main focus of this paper is to build lightweight classification models that make use of dilated convolutions to capture spectral properties and dependencies of varying sound events. The second question is, whether the models with spectrally aware layers can outperform the baseline model with common squared kernels.

### 1.1 RELATED WORK

The CNNs are well established tool for sound event recognition. The results presented in the previous papers show that various neural network architectures can perform well on different sound event datasets [4, 5, 6]. Usually, these neural network models have up to tens of millions of parameters. Conversely, all the models presented in this study have less than 100k parameters and achieve similar classification accuracy.

# 2 METHODS

## 2.1 Dilated convolutions

Dilated convolutional kernels were used instead of rectangular or one-dimensional kernels proposed in earlier literature [4]. The basic square-shaped kernel can be dilated in the time or the frequency axis by inserting zeros between the kernel values as shown in Figure 1.

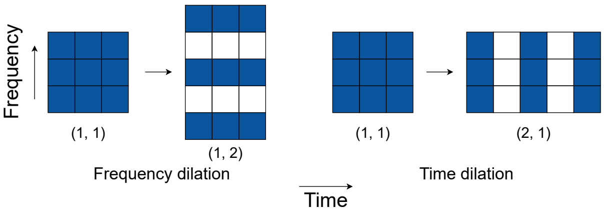

**Figure 1:** Frequency and time dilated convolutional kernels.

The main advantage of the dilated convolutions is that the number of trainable weights remains constant when increasing the dilation factor. Larger dilation factor results in extended receptive field in the selected spectrogram dimension. This should adapt the conventional convolutional neural networks towards the timbral and temporal feature extraction needed for accurate sound event recognition.

## 2.2 Dataset

The *UrbanSound8K* dataset was chosen for the model performance comparison [6]. This dataset contains 8732 labeled sound excerpts, most of which are 4 seconds long. The dataset includes manual annotations of 10 low-level classes—air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren and street music. The number of audio clips is evenly distributed across all the categories, except for the car horn and the gun shot class.

All the neural network models had input layers that accepted 88200 samples of audio, which is equal to 4 seconds at a sampling rate of 22050 Hz. In the case of shorter excerpts, the rest of the audio arrays was padded with zeros to match the required input size of the models. The dataset was divided into three subsets, where 80% of the data was used for training and the remaining 20% was split in half for validation and testing. The partitioning was done in pseudo-random fashion so that the classification accuracy of all the models could be directly compared.

## 2.3 Neural network models

The raw input waveforms were transformed to mel-spectrogram representations by a custom layer from the Kapre module for Python [7]. The mel-spectrograms were normalized across the last dimension by a batch normalization layer. Each model consisted of four convolutional layers with 3×3 kernels activated by a rectified linear (ReLU) function, followed by a dropout layers to prevent overfitting. The max pooling layers were used to downsample the results of the convolutions. The number of convolutional filters was set to 4, 8, 16, and 32 for the last layer. The feature maps of the last convolutional layers were flattened to vectors and followed by a final dropout layer. The feature vectors were fed into fully connected layer with 100 neurons. At the end of each model, there was a second fully connected layer with 10 neurons activated by a softmax function to get the final predictions. The structure of the models is shown in Table 1.

The models were built so that they had the same number of layers and roughly equal amount of trainable variables. The baseline Conv2D model had square-shaped kernels in every convolutional layer.

**Table 1:** Neural network architectures.

| | Conv2D | | ConvDF | | ConvDT | | ConvDB | |
|---|---|---|---|---|---|---|---|---|
| Layer Type | Shape | Dilation | Shape | Dilation | Shape | Dilation | Shape | Dilation |
| Input Waveform | (88200, 1) | | (88200, 1) | | (88200, 1) | | (88200, 1) | |
| Mel-spectrogram | (345, 120, 1) | | (345, 120, 1) | | (345, 120, 1) | | (345, 120, 1) | |
| Batch Normalization | (345, 120, 1) | | (345, 120, 1) | | (345, 120, 1) | | (345, 120, 1) | |
| Convolution | (343, 118, 4) | (1, 1) | (343, 118, 4) | (1, 1) | (343, 118, 4) | (1, 1) | (343, 118, 4) | (1, 1) |
| Dropout | (343, 118, 4) | | (343, 118, 4) | | (343, 118, 4) | | (343, 118, 4) | |
| Max Pooling | (171, 59, 4) | | (171, 59, 4) | | (171, 59, 4) | | (171, 118, 4) | |
| Convolution | (169, 57, 8) | (1, 1) | (169, 55, 8) | (1, 2) | (167, 57, 8) | (2, 1) | (167, 114, 8) | (2, 2) |
| Dropout | (169, 57, 8) | | (169, 55, 8) | | (167, 57, 8) | | (167, 114, 8) | |
| Max Pooling | (84, 28, 8) | | (84, 27, 8) | | (83, 28, 8) | | (83, 57, 8) | |
| Convolution | (82, 26, 16) | (1, 1) | (82, 19, 16) | (1, 4) | (75, 26, 16) | (4, 1) | (75, 49, 16) | (4, 4) |
| Dropout | (82, 26, 16) | | (82, 19, 16) | | (75, 26, 16) | | (75, 49, 16) | |
| Max Pooling | (41, 13, 16) | | (41, 19, 16) | | (37, 13, 16) | | (37, 24, 16) | |
| Convolution | (39, 11, 32) | (1, 1) | (39, 3, 32) | (1, 8) | (21, 11, 32) | (8, 1) | (21, 8, 32) | (8, 8) |
| Dropout | (39, 11, 32) | | (39, 3, 32) | | (21, 11, 32) | | (21, 8, 32) | |
| Max Pooling | (9, 3, 32) | | (9, 3, 32) | | (5, 5, 32) | | (7, 4, 32) | |
| Flatten | 864 | | 864 | | 800 | | 896 | |
| Dropout | 864 | | 864 | | 800 | | 896 | |
| Fully Connected | 100 | | 100 | | 100 | | 100 | |
| Fully Connected | 10 | | 10 | | 10 | | 10 | |
| Trainable Parameters | 93,656 | | 93,656 | | 87,258 | | 96,856 | |

The ConvDF and ConvDT models used dilated convolutions in the frequency and the time dimension, respectively. The ConvDB utilized dilated convolutions in both spectrogram dimensions. The dilated convolutions were described in more detail in Section 2.1. The exact dilation patterns are present in Table 1.

### 2.4 Evaluation metrics

To evaluate the performance of each model, following objective metrics were chosen. The accuracy (*ACC*) is defined as the fraction of correct predictions over *n* samples. Precision and recall metrics were also used. Precision is the ability of the classifier not to label as positive a sample that is negative, and recall is the ability of the classifier to find all the positive samples. Another used metric is the *F-score*, sometimes also called the *F-measure*. This metric can be defined as a weighted harmonic mean of the precision and recall. The binary metrics can be extended to multi-class problems by treating the data as a collection of binary problems, one for each class. In order to compute the multi-class metrics, macro-, or micro-averaging must be used. Macro-averaging is computed as a mean of the binary metrics, given equal weight to each class. On the other hand, micro-averaging gives each sample-class pair an equal contribution to the overall metric.

## 3 RESULTS

Note that the purpose of this study was not to create a perfect classifier, but rather investigate on the advantages of using dilated convolutions. Although related works already used dilated convolutions for audio classification [8], the models proposed in this study are substantially less complex with comparable results.

### 3.1 Training

The models were trained by minimizing the categorical cross-entropy loss, which is the most common choice when dealing with multi-class problems. The training process ran for 200 epochs on the same training and validation sets. At the end of every epoch, the training set was randomly shuffled.

It has been found that the models with dilated convolutions outperform the baseline Conv2D model by a significant margin. The ConvDF model with dilations in the frequency dimension had the second highest validation loss and the second lowest validation accuracy. The ConvDB model had the lowest validation loss and highest accuracy, making it the best performing out of the four trained models. The ConvDT was the intermediate model of the models with dilated convolutions.

### 3.2 Objective results

The trained models with the lowest validation loss were further validated on a separate subset of 874 audio examples that were not used in the process of training. This was done to check if the models are good at generalizing of the prediction on unseen data. The metrics presented in Section 2.4 were used to compare the models. The evaluation metrics computed on a test set are shown in Table 2.

**Table 2:** Performance of the models on a test set.

| Model | *ACC* | $P_{\text{macro}}$ | $R_{\text{macro}}$ | $F\text{-}score_{\text{macro}}$ |
|---|---|---|---|---|
| **Conv2D** | 0.847 | 0.874 | 0.858 | 0.857 |
| **ConvDF** | 0.882 | 0.897 | 0.889 | 0.888 |
| **ConvDT** | 0.891 | 0.906 | 0.897 | 0.896 |
| **ConvDB** | **0.914** | **0.921** | **0.917** | **0.917** |

The results presented in the previous section were confirmed by computing the test *ACC* and additional classification metrics. The models with dilated convolutions performed better than the baseline model across all the computed metrics. The *ACC* can be misleading when dealing with unbalanced datasets. Fortunately, this was not the case because the *UrbanSound8K* is a fairly balanced dataset. Nevertheless, the results were also backed by computing the *P*, *R*, and *F-score*, which verified the proposals made in the previous sections of the paper. The fact that the additional metrics did not significantly deviate from the *ACC* values means that all three models generalized well. The test *ACC* of the ConvDB model was 6.7% higher than the baseline Conv2D model. The ConvDT model performed slightly worse with *ACC* 2.3% lower than the ConvDB model. The ConvDF model had the worst performance out of the models with dilated convolutions. Still, its *ACC* was 3.5% higher than the baseline model. The normalized confusion matrix of the best performing model is shown in Figure 2.

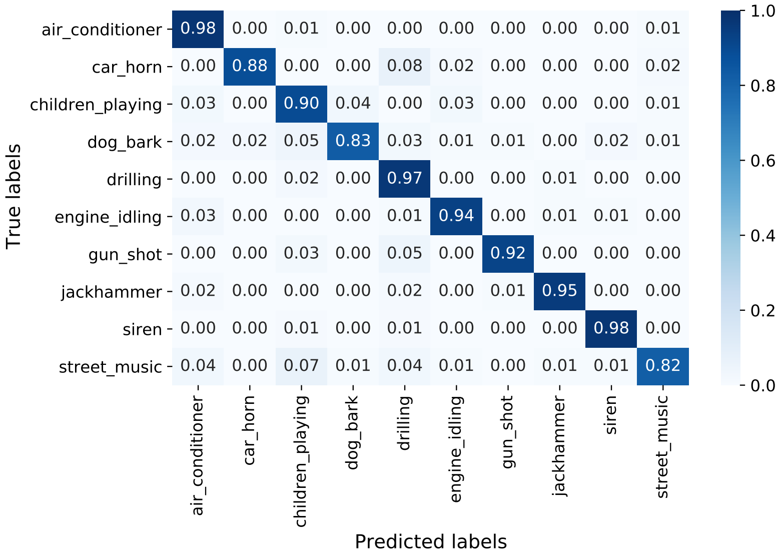

**Figure 2:** Normalized confusion matrix of the ConvDB model.

According to the confusion matrix, the car horn, the dog bark, and the street music classes were the hardest to predict. The confusions were probably caused by the prevailing background noise, which is present in some of the sound excerpts. Furthermore, as stated in the Section 2.2, numerous sound samples were shorter than the 4 second input of the neural network models. This could also contribute to the number of false predictions made by the models.

## 4 CONCLUSION

This study presented relevant architectural choices of building deep learning models for sound event recognition. The design strategy of utilizing spectrally aware layers for audio classification purposes can further improve the results obtained with the common feature extraction techniques used in image processing. The deep learning approaches are often criticized for the difficulty in understanding the hidden relationships that neural networks are learning. Many authors are using unnecessary brute force methods without rationalizing the elementary principles behind the data they are trying to classify. Of course, this work is not the universal answer for successful audio classification, but at least it tries to pinpoint the possible future directions. The proposed methods are yet to be further validated on other datasets. Also, the presented neural network models can be further modified and improved.

## REFERENCES

[1] PONS, Jordi, Oriol NIETO, Matthew PROCKUP, Erik SCHMIDT, Andreas EHMANN and Xavier SERRA. *End-to-end learning for music audio tagging at scale.* In: *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR2018).* Paris, France, 2018, pp. 637–643. ISBN 978-2-9540351-2-3.

[2] CHOI, Keunwoo, George FAZEKAS and Mark SANDLER. *Automatic tagging using deep convolutional neural networks.* In: *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR2016).* New York, USA, 2016, pp. 805–825. ISBN 978-0-692-75506-8.

[3] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN and Jian SUN. *Deep Residual Learning for Image Recognition.* In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Las Vegas, USA, 2016, pp. 770–778 [retrieved 2021-03-08]. ISBN 978-1-4673-8851-1. Available: doi:10.1109/CVPR.2016.90

[4] PONS, Jordi. *Deep Neural Networks for Music and Audio Tagging.* Barcelona, Spain, 2019. Dissertation. Universitat Pompeu Fabra. Supervised by Xavier Serra.

[5] LUZ, Jederson S., Myllena C. OLIVIERA, Flavio H.D. ARAUJO and Deborah M.V. MAGALHAES. Ensemble of handcrafted and deep features for urban sound classification. *Applied Acoustics.* 2021, **2021**(175).

[6] SALAMON, Justin, Christopher JACOBY and Juan Pablo BELLO. *A Dataset and Taxonomy for Urban Sound Research.* In: *22nd ACM International Conference on Multimedia.* Orlando, USA, 2014.

[7] CHOI, Keunwoo, Deokjin JOO and Juho KIM. *Kapre: On-GPU Audio Preprocessing Layers for a Quick Implementation of Deep Neural Network Models with Keras.* In: *Machine Learning for Music Discovery Workshop at 34th International Conference on Machine Learning.* Sydney, Australia, 2017.

[8] CHEN, Yan, Qian GUO, Xinyan LIANG, Jiang WANG and Yuhua QIAN. Environmental sound classification with dilated convolutions. *Applied Acoustics.* 2019, **2019**(148), 123–132.