# A Convolutional Neural Approach for Audio Classification in Construction Sites

Alessandro Maccagno[1], Andrea Mastropietro[1], Umberto Mazziotta[1], Michele Scarpiniti[2], and Aurelio Uncini[2]

[1] Sapienza University of Rome, Department of Computer, Control and Management Engineering, Italy,
{maccagno.1653200,mastropietro.1652886,mazziotta.1647818}@studenti.uniroma1.it,
[2] Sapienza University of Rome, Department of Information Engineering, Electronics and Telecommunications, Italy
michele.scarpiniti@uniroma1.it, aurel@ieee.org

**Abstract.** Convolutional neural networks have been widely used in the field of audio recognition and classification, often with extremely positive results. Due to the success of said approach, it is our belief that application using neural techniques may be developed to tackle increasingly more complex and specific problems; to this end, we developed an application to classify different types and brands of construction vehicles and tools, which operates on the emitted audio through a stack of convolutional layers. The proposed network works on the mel-spectrogram representation of each audio excerpt; this approach is usually employed in automatic speech recognition but recent results have demonstrated its effectiveness even in environmental sound classification (ESC). Our contribution will be to show that techniques employed in general ESC can be successfully adapted to work in a more specific environmental sound classification task, such as construction sites event recognition.

**Keywords:** deep learning, convolutional neural networks, audio processing, environmental sound classification

## 1 Related Work

In the literature it possible to find several instances of successful applications in the field of environmental sound classification which make use of deep learning. For example, in the work of Piczak [3], the author exploits a 2-layered CNN working on the spectrogram of the data to perform ESC, reaching an average accuracy of 70% over different datasets. Other approaches, instead of using handcrafted features such as the spectrogram, perform end-to-end environmental sound classification obtaining higher results with respect to the previous ones [5]. Getting into the more specific field of environmental audio classification in construction site, the closest attempts have been performed by Cheng et al. [7] who used support vector machines (SVM) to analyze activity of construction tools and equipment. The contribution of this paper will be to adapt preexisting models which have proven to be effective in general environmental sound classification to a way more specific and novel domain such as construction yard equipment type and brand classification, obtaining remarkable results.

## 2 Overview

As aforementioned, the aim of the work presented in this paper is to create an application able to recognize vehicles and tools used in construction sites, and classify them in terms of type and brand. This task will be tackled with a neural approach, that will involve the use of a deep convolutional neural network (CNN), which will be fed with the mel spectrogram of the audio source as input. Our classification will be carried on 5 classes extracted from the Utah Audio Data dataset, containing in situ recordings of multiple vehicles and tools. Our study is different from the previous ones since we are focused on a very specific domain, and we work on a dataset constituted of real world examples, not on data built on purpose. We will demonstrate that neural approaches for ESC can be adapted with good results (average accuracy of 97%) to a very specific domain as the one of construction sites. In particular, the architecture we are proposing is motivated by the MelNet architecture described by Li et al. [1], which has been proven to be remarkably effective in environmental sound classification.

## 3 Dataset

The dataset we worked on is the `Utah Audio Data`, and it is composed of real audio tracks recorded in construction sites in Utah, USA. Unlike artificially built datasets, when working with real data different problems arise, such as noise due to weather conditions and/or workers talking among themselves. Thus, we focused our work on the classification of a reduced number of classes, that are *Backhoe JD50D Compact*, *Compactor Ingersoll Rand*, *Concrete Mixer*, *Excavator Cat 320E*, *Excavator Hitachi 50U*, all of which having approximately 15 minutes of audio. Classes which did not have enough usable audio (too short, excessive noise, low quality of the audio) were ignored for this work.

### 3.1 Data Preprocessing

In order to feed the network with enough and proper data, each audio file for each class is segmented into fixed length subsamples (the choice of the best sample size is described in the experiment section). As first step we split the original audio files into two parts, training samples (70% of the original length) and test samples (30% of the original length); this is done to avoid testing the network on data used previously to train the network, as this would cause the network to overfit and give misleading results.

Then we perform data augmentation by splitting the files into smaller segments of 30ms, each of which overlaps the subsequent one by 15ms. We then compute

the RMS (Root Mean Square) of every signal of these smaller segments, and drop the ones with too small power w.r.t the average RMS of the different segments, in order to remove the segments which contain mostly silence.
After that, the dataset is balanced by taking N samples for each class, where N is the number of elements contained in the class with the least amount of samples. This way, we avoided the problem of having certain classes with an abnormal number of usable audio segments being potentially either overrepresented or underrepresented and negatively impacting the training of the model, especially due to the presence of multiple models of the same vehicle.
Using the the Python library `librosa`[3] we extracted the waveform of the audio track from the audio subsamples and, using the same library, we generated the log-scaled mel spectrogram[2] of the signal that will be the input to the network.

### 3.2 Spectrogram Extraction

The technique used to extract the spectrogram from the sample is the same used by Piczak[3]. The samples were re-sampled to 22050Hz, then we used a window of size 1024 with hop-size of 512 and 60 mel bands. A mel band represents an interval of frequencies which are perceived to have the same pitch by human listeners. They have been found to be performing in speech recognition.
With this parameters, and the chosen length of 30ms for the samples (see next sections), we obtain a small spectrogram of 60 rows (bands) and 2 columns (frames). Then, using again `librosa`, we compute the derivative of the spectrogram and we overlap the two matrices, obtaining a dual channel input which is fed into the network.

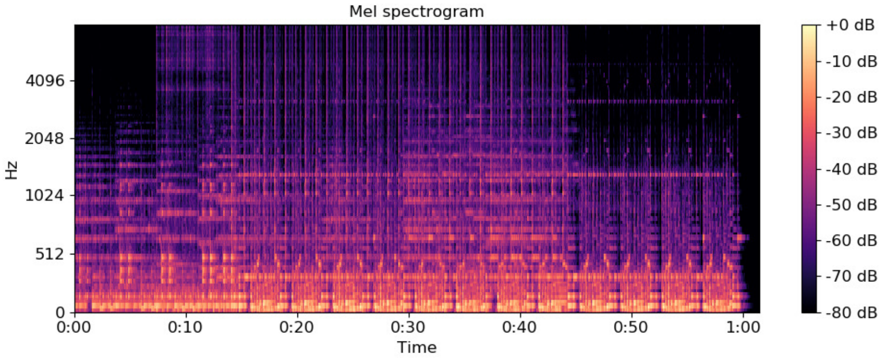

**Fig. 1.** Example of log-mel spectrogram extracted from an audio source.

[3] https://librosa.github.io/

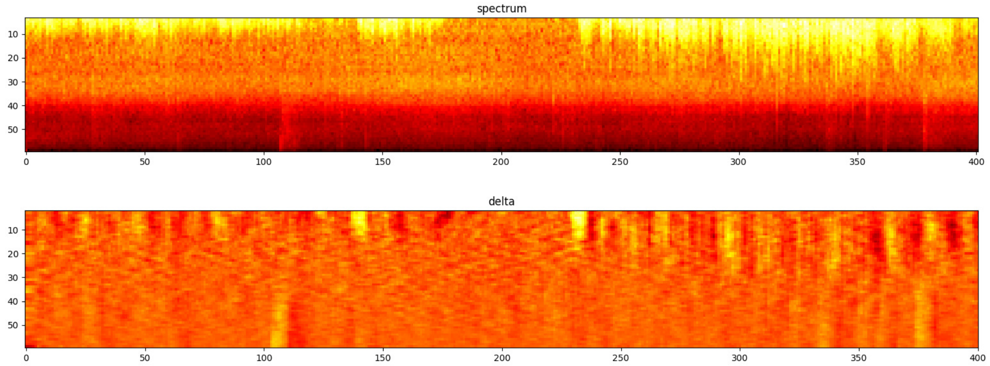

**Fig. 2.** Our case: log-mel spectrogram extracted from a fragment along with its derivative. On the abscissae we find the time buckets, each of which representing a sample about 23ms long, while on the ordinates the log-mel bands. Since our fragments are 30ms long, the spectrogram we extract will contain 2 buckets.

# 4 Convolutional Neural Networks

## 4.1 Motivation

# 5 Architecture

The CNN is fed with the spectrogram of a sample stacked with its time derivative: thus, we end up having a 60x2 input image with 2 channels, where 60 is the number of mel bands, 2 is the number of time buckets and the channels represent respectively the spectrogram and its time derivative.
The structure of the network is the following, and it can be graphically appreciated in Fig. 3:

1. Input layer: size of (60, 2, 2).
2. Convolutional layer: 24 filters, kernel size of (6, 2), strides (1,1), ReLu activation function.
3. Convolutional layer: 24 filters, kernel size of (6, 2), strides (1,1), ReLu activation function.
4. Convolutional layer: 48 filters, kernel size of (5, 1), strides (2,2), ReLu activation function.
5. Convolutional layer: 48 filters, kernel size of (5, 1), strides (2,2), ReLu activation function.
6. Convolutional layer: 64 filters, kernel size of (4, 1), strides (2,2), ReLu activation function.
7. Dense layer: 200 units, ReLu activation function.
8. Dropout: dropout rate of 0.3.
9. Output layer: dense layer with 5 units (one for each class) with softmax activation function.

The optimizer chosen for the network is an Adam Optimizer [6], with the a learning rate of 0.0005. Such value was chosen by performing several runs trying different learning rates (such as 0.001, 0.0001, 0.00001 and more) and ending up having the best results with 0.0005.

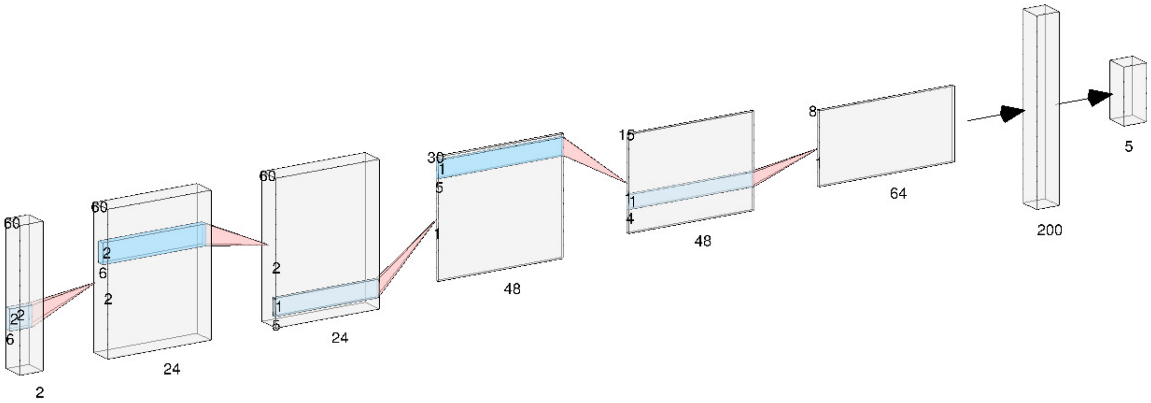

**Fig. 3.** Graphical representation of the network architecture.

## 6 Experiments

A sizeable amount of time was spent into finding the proper length for the audio segments. This is of crucial importance since, if the length is not adequate the network will not be able to learn proper features that clearly characterize the input. So, in order to select the most suitable length, we generated different dataset variants by splitting the audio using different lengths; then, we trained different models, which we subsequently tested each of them on one of the differently-sized datasets. The testing results are show in Fig 4.

As we can see, with smaller sample sizes better results are obtained, while we notice a drop as the size increases. It is also interesting to observe that with very large sample sizes the accuracy tends to slightly improve again; but the usage of long samples does not lead to anything interesting since the network may tend to learn an ensemble of the signal that is not significant and useful if working with small audio fragments, that are needed by fast-response applications (hazard detection, activity monitoring, etc.). Finally, the sample size we ended up choosing is 30ms, since it led not only to having a high accuracy but also a larger number of samples. In order to properly test the network we performed a k-fold

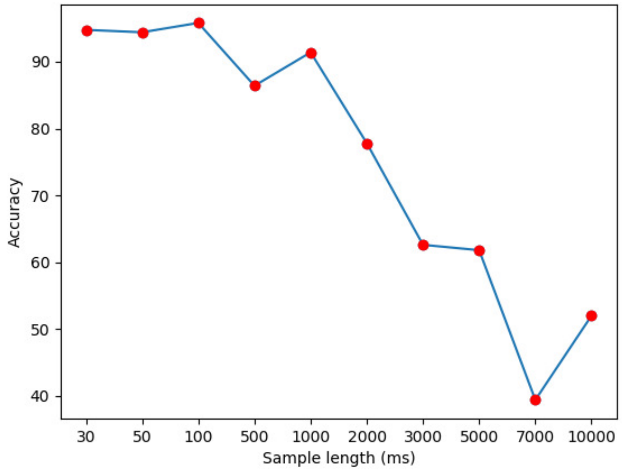

**Fig. 4.** Classification accuracy according to different sample sizes of the audio subsamples.

cross validation, with $k = 5$. The results of the classification are shown in the next subsection.

### 6.1 Classification Results

As just stated, a 5-fold cross validation was performed and the results are shown in Table 6.1. The dataset was split into training set and validation set (80%-20%) for each fold.

| Class | Accuracy | Recall | Precision | F1 |
|---|---|---|---|---|
| Backhoe JD50D Compact | 98.52 | 97.23 | 95.54 | 96.34 |
| Compactor Ingersoll Rand | 98.73 | 97.89 | 95.71 | 96.76 |
| Concrete Mixer | 99.21 | 98.49 | 97.58 | 98.03 |
| Excavator Cat 320E | 99.19 | 97.34 | 98.60 | 97.96 |
| Excavator Hitachi 50U | 98.99 | 97.82 | 97.16 | 97.49 |
| **All classes** | **97.08** | **97.34** | **97.30** | **97.32** |

**Table 1.** K-Fold cross validation classification results

As we can notice, the network achieves very high results in all the metrics, demonstrating its effectiveness in this particular domain. Even though our classes include also vehicles of the same type (we have two excavators and a backhoe is

a kind of excavator as well) such classes are discriminated in a very clear and accurate way as the net recognizes also the brand of the machine.
After having performed the cross validation, we trained the network again on the original version of the dataset (training set 70% and test set 30%) and tested it. The accuracy results obtained are shown in the confusion matrix in Fig. 6. The way the network learns can be seen in Fig. 5; the learning is actually really fast as we see that high accuracy values are reached within few epochs and thus the convergence is rapid.

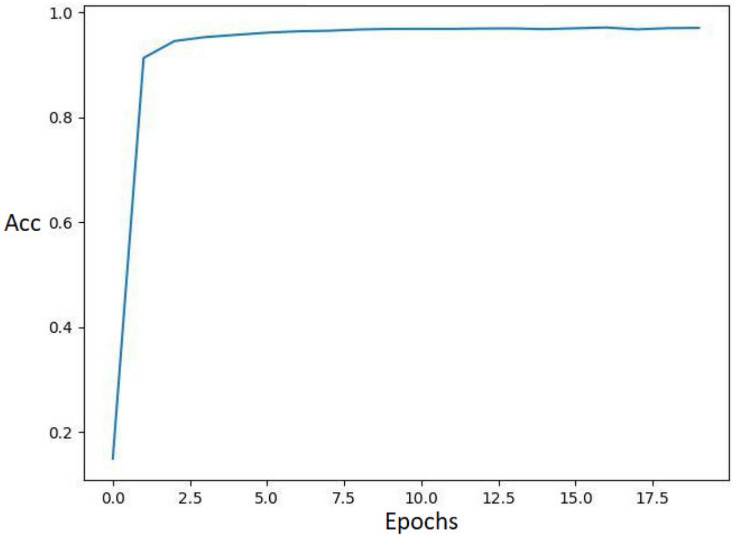

**Fig. 5.** Accuracy on test set.

### 6.2 Prediction

In order to predict a new sample in input, such audio file is split into segments as described above; every fragment will be classified as belonging to one of the classes and the audio track will be labelled according to the majority of the labels among all the fragments; in this way we can also see what is the probability for the input track to belong to each of the classes.

## 7 Conclusions and Future Work

In this paper we demonstrated that it is possible to apply a neural approach already tested in environmental sound classification to a more specific domain,

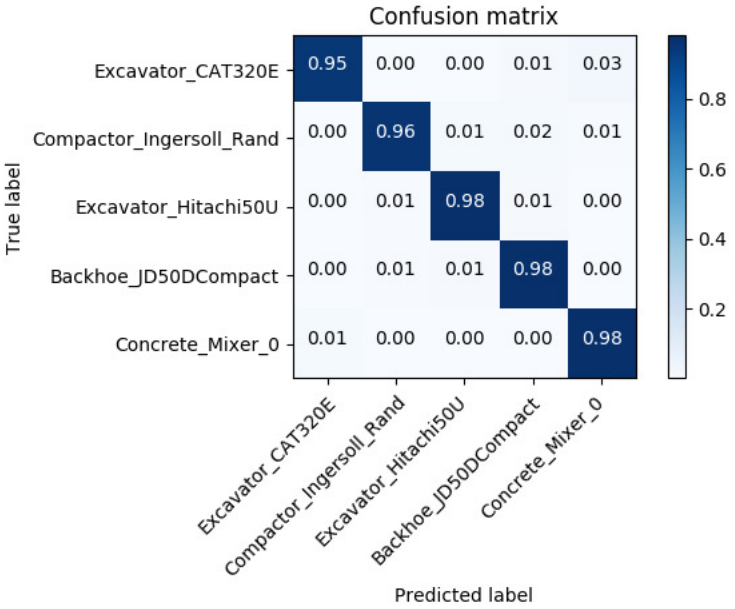

**Fig. 6.** Confusion matrix.

that is the one of construction sites, with rather high results. Such architecture works with small audio fragments and, for practical applications, the ability to perform a classification using very short samples can lead to the possibility to use such network in time-critical applications in construction sites that require fast responses, such as hazard detection and activity monitoring.

Up to now, the network was tested on 5 classes; the idea is to try to increase the number of classes to include more tools and vehicles employed in building sites in order to lead in the future to a more reliable and useful system. Moreover, the most interesting way to extend the work would be to try to combine more architectures to see to which extent different kind of neural networks can help in such kind of audio classification.

## References

1. S. Li, Y. Yao, J. Hu, G. Liu, X. Yao, and J. Hu, “An ensemble stacked convolutional neural network model for environmental event sound recognition,” *Applied Sciences*, vol. 8, no. 7, 2018.
2. E. B. Stevens, Stanley Smith; Volkmann; John Newman, “A scale for the measurement of the psychological magnitude pitch,” 1937.
3. K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Sep. 2015.

4. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.
5. Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2721–2725, March 2017.
6. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.
7. CHENG