Deep Feature Learning for Acoustics-based Terrain Classification

Abhinav Valada, Luciano Spinello, and Wolfram Burgard

Abstract In order for robots to efficiently navigate in real-world environments, they need to be able to classify and characterize terrain for safe navigation. The majority of techniques for terrain classification is predominantly based on using visual features. However, as vision-based approaches are severely affected by appearance variations and occlusions, relying solely on them incapacitates the ability to function robustly in all conditions. In this paper, we propose an approach that uses sound from vehicle-terrain interactions for terrain classification. We present a new convolutional neural network architecture that learns deep features from spectrograms of extensive audio signals, gathered from interactions with various indoor and outdoor terrains. Using exhaustive experiments, we demonstrate that our network significantly outperforms classification approaches using traditional audio features by achieving state of the art performance. Additional experiments reveal the robustness of the network in situations corrupted with varying amounts of white Gaussian noise and that fine-tuning with noise-augmented samples significantly boosts the classification rate. Furthermore, we demonstrate that our network performs exceptionally well even with samples recorded with a low-quality mobile phone microphone that adds substantial amount of environmental noise.

1 Introduction

Robots are increasingly being used for tasks in unstructured real-world environments and thus have to be able to deal with a huge variety of different terrains. As every terrain has a distinct physical property, it necessitates an appropriate navigation strategy to maximize the performance of the robot. Therefore, terrain classification is paramount to determine the corresponding trafficability. However, it is a highly challenging task to robustly classify terrain. Especially, the predomi-

All authors are with the Department of Computer Science, University of Freiburg, Germany · Corresponding author's e-mail: valada@informatik.uni-freiburg.de

nantly used vision-based approaches suffer from rapid appearance changes due to various factors including illumination variations, changes in weather, damping due to rain and camouflaging with leaves. Accordingly, researchers have also explored the utilization of alternative modalities such as ladars or vibrations measured using accelerometers. Each of these approaches have their own advantages and disadvantages. For example, optical sensors are quintessential when there is good illumination and distinct visual features, while accelerometer-based approaches are ideal to classify terrains with varying degrees of coarseness. However, the use of sound to classify terrains in the past has not been studied in a comparable depth, even though sound produced from vehicle-terrain interactions have distinct audio signatures even utilizable for fine-grained classification. Most importantly, the disturbances that affect other light-based or active sensors do not affect microphones, hence they can even be used as a complementary modality to increase robustness. We believe that utilization of a complementary set of sensing modalities is geared towards long-term autonomy.

In this paper, we present a novel multiclass terrain classification approach that uses only audio from the vehicle-terrain interaction to robustly classify a wide range of indoor and outdoor terrains. As in any pattern recognition task, the choice of features significantly dictates the classification performance. Vehicle-terrain interaction sounds are very unstructured in nature as several dynamic factors contribute to the signal. Instead of using handcrafted domain specific features, our approach employs a deep convolutional neural network (DCNN) to learn them. DCNNs have recently been achieving state of the art performance on several pattern recognition tasks [13, 14, 18]. They learn unsupervised hierarchical feature representations of their input by exploiting spatial correlations. The additional advantage of this is that the features learned from this approach generalize effectively as DCNNs are relatively insensitive to certain input variations.

The convolutional neural network architecture we introduce is built upon recent advances in deep learning. Our network consisting of six convolution layers and six cascaded cross channel parametric pooling layers is depicted in Fig. 1. In order to make the learned feature representations invariant to certain signal variations and also to increase the number of training samples, we performed a number of transformations on the original signal to augment the data. We experimented with several hyperparameters for our network and show that it significantly outperforms classification methods using popular baseline audio features. To the best of our knowledge, this is the widest range of terrain classes successfully classified using any proprioceptive terrain classification system. Additionally, our method achieves state of the art performance in classification using a proprioceptive sensor. Audio classification is susceptible to background noise to a great extent. We stress test our network with additive white Gaussian noise (WGN) at varying signal to noise ratios (SNR). We also perform noise aware fine-tuning to increase the robustness and show that our network performs exceptionally well even on audio data collected by the robot with a low quality mobile phone microphone which adds significant environmental noise.

2 Related Work

The use of sound as a modality for classifying vehicle-terrain interactions has very sparsely been explored. The following are the only related works using acoustics for terrain classification. Ojeda *et al.* [17], used a feedforward neural network and a suite of sensors for terrain classification, including a microphone, gyroscopes, accelerometers, motor current and voltage sensors, infrared, ultrasonics and encoders. They had five terrain classes and their classifier achieved an average classification accuracy of 60.3% using the microphone. They found that using the entire spectrum gave them the same performance as using only 0 to 50 Hz components of the discrete fourier transform. The authors concluded that overall the performance was poor using the microphone, other than for classifying grass.

More recently, Libby and Stentz [15] trained a multiclass sound-based terrain classifier that uses Support Vector Machines (SVMs). They evaluated the performance of various features using extraction techniques derived from the literature survey as input to the SVM. Their multidimensional feature vectors consists of spectral coefficients, moments and various other temporal as well as spectral characteristics. Their classifier achieves an average accuracy of 78% over three terrain classes and three hazardous vehicle-terrain interaction classes. They further increase the accuracy to 92% by smoothing over a window of 2 seconds.

A patent by Hardsell *et al.* [8] describes an approach to terrain classification where a classifier is trained on fused audio and video data. They extract scale invariant transformation features from the video data and use Gaussian mixture models with a time-delay neural network to represent the audio data. The classifier is then built using expectation-maximization.

The use of contact microphones for terrain classification has also been explored. Unlike air microphones that we use in our work, contact microphones pick up only structure-borne sound. Brooks and Iagnemma [2] use a contact microphone mounted on their analog rover's wheel frame to classify terrain. They extract the log-scaled Power Spectral Density (PSD) of the recorded vibrations and used them to train a pairwise classifier. Their classifier with three classes, achieves an average accuracy of 74% on a wheel-terrain testbed and 85.3% on the test bed rover. They also present a self-supervised classifier that was first trained on vibration data, which then provided the labels for training a visual classifier [3].

A number of methods have been developed for using accelerometer data to classify terrain [17, 19, 21]. Weiss *et al.* [21] use vibrations induced in the vehicles body during traversal to classify the terrain. They train a seven class SVM with features extracted from log-scaled PSD, discrete fourier transform and other statistical measures. Their classifier produced an average accuracy of 91.8% over all the classes. However, such approaches report a significant number of false positives for finer terrains such as asphalt and carpet. For another similar application, Eriksson *et al.* [5] employ a mobile sensor network system that uses hand selected features from accelerometer data to identify potholes and other road anomalies. Their system detects the anomalies over 90% of the time in real-world experiments.

There is a considerable amount of specialized audio features developed for speech recognition and music classification, but it remains unclear which of these features performs well for our application. We evaluated several traditional audio features from our literature survey and compared them as baseline approaches. Libby and Stentz [15] show that a combination of Ginna and Shape features perform the best for classification of vehicle-terrain interactions. Gina features, based on the work by Giannakopoulos *et al.* [6] is a 6D feature vector consisting of zero crossing rate (ZCR), short time energy (STE), spectral centroid, spectral rolloff and spectral flux. Shape features, based on the work by Wellman *et al.* [22], characterize the distribution of moments of the spectrum. It is a 4D feature vector consisting of spectral centroid, standard deviation, skewness and kurtosis.

Ellis [4] use a combination of mel-frequency cepstral coefficients (MFCCs) and chroma features. MFCCs are the most widely used features for audio classification and Chroma features are strongly related to the harmonic progression of audio signals. We use a combination of twelve bin MFCC's and twelve bin Chroma features for comparison. Trimbral features have been a popular set of features for various audio classification applications. Tzanetakis and Cook [20] use a 19D feature representation consisting of means and variances of spectral centroid, rolloff, flux, ZCR, low energy and means and variances of the first 5 MFCCs. For our final feature set comparison, we use a combination of 13 bin MFCC's, line spectral pair (LSP) and linear prediction cepstral coefficients (LPCCs) [1]. We call this Cepstral feature set in the later discussions.

3 Deep Convolutional Neural Network For Acoustic Based Terrain Classification

One of the main objectives of our work is to develop a new deep convolutional neural network architecture tailored to classifying unstructured vehicle-terrain interaction sounds. In this section, we detail the various stages of our classification pipeline shown in Fig. 1. Our approach can be split into two main stages. The first stage involves processing the raw audio samples into short windowed clips, augmenting the samples and spectrogram transformation. The second involves training our deep convolutional neural network with this data.

3.1 Preprocessing and Spectrogram Extraction

We first split the audio signals from each class into small "clips" of t_w seconds. We experimentally determine the shortest clip length that gives the best classification performance. Feature responses from each of these clips are then extracted and added as a new sample for classification.

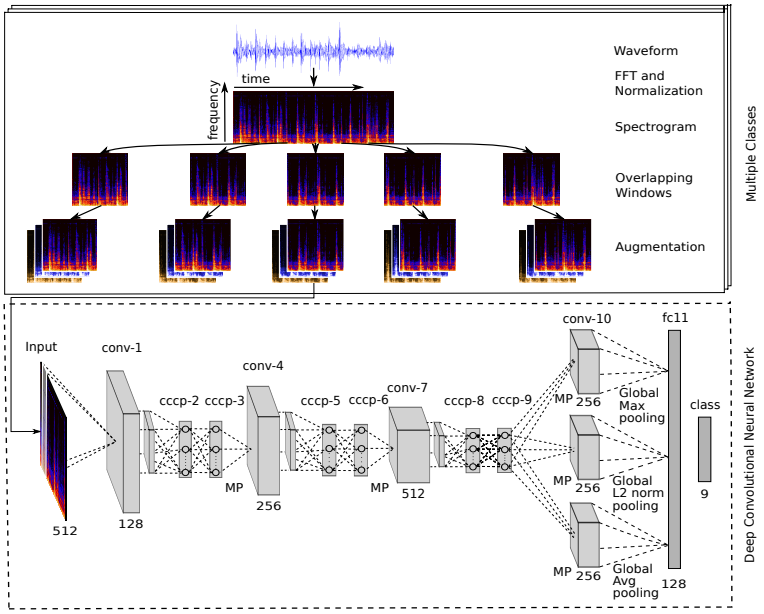

Fig. 1 Overview of our terrain classification pipeline. Raw audio signal of the terrain interaction is first transformed into its spectrogam representation and then piped into a DCNN for feature learning and classification. MP refers to max pooling.

Features derived from spectrogram representations of audio signals have been shown to outperform other standard features for environmental sound classification applications [12]. Therefore in our approach, we extract the Short Time Fourier Transform (STFT) based spectrogram of each clip in our dataset. We first block each audio clip into M samples with 75% overlap between each frame. Let $x[n]$ be the recorded raw audio signal with duration of N_f samples, f_s the sampling frequency, $S(i, j)$ be the spectrogram representation of the 1-D audio signal and $f(k) = kf_s/N_f$. By applying STFT on length M windowed frame of signal, we get

$$X(i,j) = \sum_{p=0}^{N_f-1} x[n]\, w[n-j] \exp\left(-p\frac{2\pi k}{N_f}n\right), \quad p = 0,\ldots,N_f-1 \tag{1}$$

A Hamming window function $w[n]$ is used to compensate for Gibbs effect while computing STFT by smoothing the discontinuities at the beginning and end of the audio signal.

$$w[n] = 0.54 - 0.46\cos\left(2\pi\frac{n}{M-1}\right), \quad n = 0,\ldots,M-1 \tag{2}$$

We then compute the log of the power spectrum as

$$S_{log}(i,j) = 20\log_{10}(|X(i,j)|) \tag{3}$$

We chose N_f as 2,048 samples, therefore the spectrogram contains 1,024 Fourier coefficients. By analyzing the spectrum, we found that most of the spectral energy is concentrated below 512 coefficients, hence we only use the lower 512 coefficients to reduce the computational complexity. The noise and intensity levels vary a fair amount in the entire dataset as we collected data in different environments. Therefore, we normalized the spectrograms by dividing by the maximum amplitude. We compute the normalized spectrogram as $S(i,j) = S_{\log}(i,j)/max_{i,j}S_{\log}(i,j)$. We then compute the mean spectrum over the entire dataset and subtract it from the normalized spectrogram to remove any temporal artifacts.

We created additional training samples by applying a set of augmentation strategies A_t on the audio signal in the frequency domain. Offsets in time and frequency was used to perform shifting to transform the spectrogram. The transformations were applied using 2D affine transform and warping, keeping the shape constant. Furthermore we created more samples using time stretching, modulating the tempo, using random equalization augmentation and by increasing as well as decreasing the volume gain. We also experimented with frequency and time normalization with a sliding window and local contrast normalization.

3.2 Network Architecture and Training

The extracted spectrograms in our training set are of the form $S = \{s^1, \ldots, s^M\}$ with $s^i \in \mathbb{R}^N$. Each of them are of size $v \times w$ and number of channels d ($d = 1$ in our case). We assume M to be the number of samples and y^i as the class label in one-hot encoding, $y^i \in \mathbb{R}^C$, where C is the number of classes. We then train the DCNN by minimizing the negative log likelihood of the training data. Our network shown in Fig. 1 has six Convolution layers, six Cascaded Cross Channel Parametric Pooling (CCCP) layers, two Fully-Connected (FC) layers and a Softmax layer. All the convolution layers are one dimensional with a kernel size of three and convolve along the time dimension. We use a fixed convolutional stride of one. CCCP layers follow the first, second and third convolution layers. CCCP layers was proposed by Lin *et al.* [16] to enhance discriminability for local patches within the receptive fields. CCCP layers are effectively employ 1×1 convolutions over the feature maps and the filters learnt are a better non-linear function approximator. A max-pooling layer with a kernel of 2, then follows the second and fourth CCCP layers. Max-pooling adds some invariance by only taking the high activations from adjacent hidden units that share the same weight, thereby providing invariance to small phase shifts in the signal.

DCNNs that are used for feature learning with images are designed to preserve the spatial information of objects in context, however for our application we are not interested to localize features in the frame, rather we are only interested to identify the presence or absence of features in the entire frame. Therefore, we added three different global pooling layers after CCCP-9 to compute the statistics across time. This global pooling approach is similar to that used for content based music recom-

mendation by Oord *et al.* [18]. For global pooling layers, we use max pooling, L2 norm pooling and average pooling. We experimented with just one global pooling layer and combinations of two global pooling layers and the accuracy dropped over 3% while compared to using all three global pooling layers. We also investigated the effect of global stochastic pooling with the other three pooling combinations, but the network did not show any significant improvement. Finally, a fully connected layer is then used to combine outputs of all the global pooling layers.

Rectified linear units (ReLUs) have significantly helped in overcoming the vanishing gradient problem. They have been shown to considerably accelerate the training compared to *tanh* units. We use ReLUs $f(x) = \max(0,x)$, after the convolution layers and dropout regularization [10] on fully connected layers except the softmax layer. We used a dropout probability of 0.5. We also experimented with Parameterized Rectified Linear Units (PReLU) [9], which has shown to improve model fitting but it drastically affected our performance compared to ReLUs.

We used Xavier weight initialization [7] for the Convolution, CCCP and FC layers. The Xavier weight filler initializes weights by drawing from a zero mean uniform distribution from $[-a,a]$ and a variance as a function of the number of input neurons, where $a = \sqrt{3/n_{in}}$ and n_{in} is the number of input neurons. Using this strategy enables us to move away from the traditional layer by layer generative pretraining. Let $f_j(s^i;\theta)$ be the activation value for spectrogram s^i and class j, θ be the parameters of the network (weights W and biases b). The softmax function and the loss is computed as

$$P(y = j \mid s^i;\theta) = \text{softmax}(f(s^i;\theta)) = \frac{\exp(f_j(s^i;\theta))}{\sum_{k=1}^{K} \exp(f_k(s^i;\theta))} \tag{4}$$

where $P(y = j \mid s^i;\theta)$ is the probability of the j^{th} class and the loss can be computed as $\mathcal{L}(u,y) = -\sum_k y_k \log u_k$. Using stochastic gradient decent (SGD), we then solve

$$\min_{\theta} \sum_{i=1}^{N} \mathcal{L}(\text{softmax}(f(s^i;\theta)), y^i) \tag{5}$$

We use minibatch SGD with a momentum of 0.9 and a batch size of 128. Minibatch SGD refers to a more efficient way of computing the derivatives before updating the weights in proportion to the gradient, especially in large datasets such as ours. We improve the efficiency by computing the derivative on a random small minibatch of training samples, rather than the entire training set which would be computationally exhaustive. Furthermore, we optimize SGD by smoothing the gradient computation for minibatch t using a momentum coefficient α as $0 < \alpha < 1$. The update rule can then be written as

$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) - \epsilon \frac{\partial E}{\partial w_{ij}(t)} \tag{6}$$

We employ a weight decay of $\lambda = 5 \cdot 10^{-4}$ to regularize the network. We begin the training with an initial learning rate of λ_0 and reduced it every iteration by an inverse learning rate policy as $\lambda_n = \lambda_0 * (1 + \gamma * N)^{-c}$. Where λ_0 is the base learning rate, N is the number of iterations and c is the power. We use $c = 0.75$ and $\gamma = 0.1$. We determine the hyperparameter λ_0 by experimenting with different rates in an initial trial. The best performing rate of 10^{-2} was then ascertained. The entire training of 350K iterations ($\sim$ 135 epochs) took about 4 days on a single GPU.

3.3 Noise Aware Fine-Tuning

Classification performance is often strongly affected by noise from the environment. Since the microphone is mounted on the robot and used in real-world environments, it is inevitable that the recorded signals include the robot's motor noise in addition to environmental noise. Fortunately deep networks have good generalization to real-world scenarios if they are trained with noisy samples. In order to quantify the performance in the presence of noise, we added WGN to training samples at various SNR's and measured the classification accuracy. WGN adds a very similar effect as various physical and environmental disturbances including wind and water sources.

From experiments detailed in Sect. 5.4, it can be seen that the classification performance of our network quickly drops below SNRs of 40 dB. As a solution to this problem, we augmented raw audio signals with additive WGN at SNRs ranging from 50 dB to -10 dB, in steps of 10 dB. We then performed noise adaptive fine-tuning of all the layers in our network with the training set containing both noised and original samples. The weights and biases are initialized by coping from our original model trained as described in Sect. 3.2. The new model is then trained by minimizing the negative log likelihood as shown in Eq. (5). We again use minibatch SGD with a learning rate $1/10th$ of the initial rate use for training the network, 10^{-3}. The learning rate was further reduced by a factor of 10, every 20,000 iterations.

4 Data Collection and Labeling

As we are particularly interested in analyzing the sounds produced from the vehicle-terrain interaction on both indoor and outdoor terrains, we use the Pioneer P3-DX platform which has a small footprint and feeble motor noise. Interference from nearby sound sources in the environment can drastically influence the classification. It can even augment the vehicle-terrain interaction data by adding its own attributes from each environment. In order to prevent such biases in the data being collected, we use a shotgun microphone that has a supercardioid polar pattern which helps in rejecting off-axis ambient sounds. We chose the Rode VideoMic Pro and mounted it near the right wheel of the robot as shown in Fig. 3. The integrated shock mount in the microphone prevents any unwanted vibrations from being picked up.

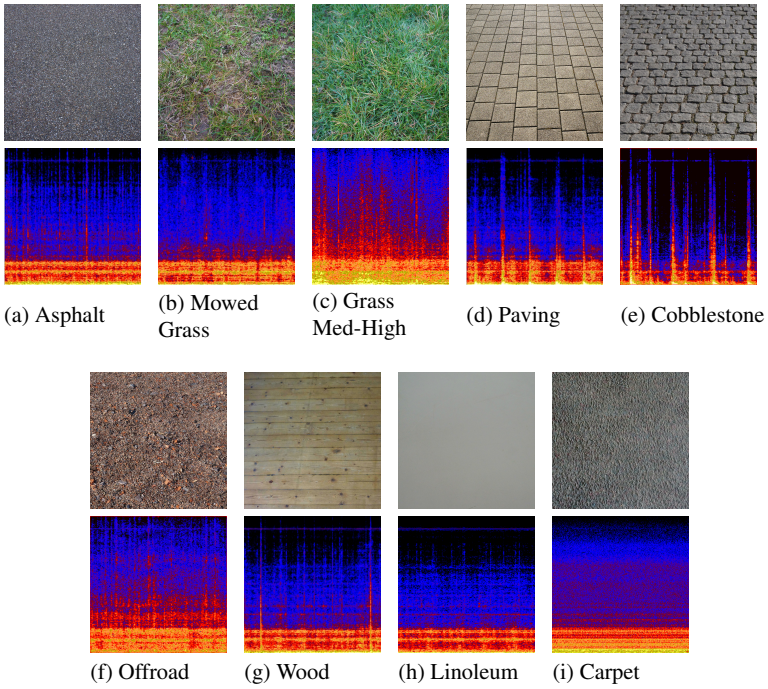

Fig. 2 Terrain classes and an example spectrogram of a 2,000 ms clip (colorized spectrograms are only shown for better visualization, spectrograms used for training are in gray scale).

We collected over 15 hours of audio data from a total of 9 different indoor and outdoor terrains. We particularly choose our terrain classes such that some of them have similar visual features (Fig. 2(a), 2(h), 2(i)) and hence pose a challenge to vision based approaches. The data was collected at several different locations to have enough generalizability, therefore even signals in each class have varying temporal and spectral characteristics. The robots speed was varied from $0.1\,\mathrm{m\,s^{-1}}$ to $1.0\,\mathrm{m\,s^{-1}}$ during the data collection runs. The data was recorded in the lossless 16-bit WAV format at 44.1 kHz to avoid any recording artifacts. Experiments were conducted by recording at various preamp levels and microphone mounting locations. There was no software level boost added during the final recordings as they also tended to amplify the ambient noise significantly, instead the microphones 20 dB hardware level boost was turned on.

All the data was manually labeled by looking at live tags with timestamps that were made during the recordings. A waveform analyzer tool was used to crop out any significant disturbances. The data from each class was then split into overlapping time windows, where each window is then used separately as a new data sample for feature extraction. As Libby *et al.* mention in [15], choosing an appropriate length for the time window is critical, as too short of a window might cut off a potential feature and by having too large of a window we will loose the classification

Fig. 3 The Pioneer P3-DX platform showing the shotgun microphone with the shock mount, mounted close to the wheel.

resolution. We also analyzed the effect of different window sizes in our experiments. In order to train the classifier to be generalizable to different locations with the same terrain, a ten-fold cross validation approach was adopted. Furthermore, we ensured that all the sets and classes have approximately the same number of samples to prevent any bias towards a specific class.

5 Experimental Results

We performed the implementation and evaluations using the publicly available, Caffe [11] deep learning toolbox and ran all our experiments on a system with an Intel i7-4790K processor and a NVIDIA GTX 980M GPU. We used the cuDNN library for GPU acceleration. For all the baseline comparisons and noise robustness tests, we chose a clip window length of 300 ms and performed ten-fold cross-validation. The results from our experiments are described in the following sections.

5.1 Baseline Comparison

We chose two benchmark classifiers, k-Nearest Neighbors (kNNs) and SVMs. SVMs perform well in high dimensional spaces and kNNs perform well when there are very irregular decision boundaries. As a preprocessing step we first normalize the data to have zero mean. We use the one-vs-rest voting scheme with SVM to handle multiple classes and experimented with Linear and Radial Basis Function (RBF) kernels as decision functions. We used inverse distance weighting for kNNs and optimized the hyperparameters for both the classifiers by a grid-search using cross-validation. We empirically evaluated six popular feature combinations described in Sect. 2, with SVM and kNN. We used scikit-learn and LibSVM for the implementation. It was ensured that the training and validation sets do not contain the same audio split or the augmented clip. The results from this comparison are shown in Table 1.

Table 1 Classification accuracy of several baseline feature extraction approaches on our dataset

Features	SVM Linear	SVM RBF	k-NN
Ginna	44.87 ± 0.70	37.51 ± 0.74	57.26 ± 0.60
Spectral	84.48 ± 0.36	78.65 ± 0.45	76.02 ± 0.43
Ginna & Shape	85.50 ± 0.34	80.37 ± 0.55	78.17 ± 0.37
MFCC & Chroma	88.95 ± 0.21	$\mathbf{88.55 \pm 0.20}$	88.43 ± 0.15
Trimbral	89.07 ± 0.12	86.74 ± 0.25	84.82 ± 0.54
Cepstral	$\mathbf{89.93 \pm 0.21}$	78.93 ± 0.62	$\mathbf{88.63 \pm 0.06}$
DCNN (ours)		$\mathbf{97.36 \pm 0.12}$	

The best performing baseline feature-classifier combination was Cepstral features using a linear SVM kernel, although the performance using Trimbral features are closely comparable. This feature set outperformed Ginna and Shape features by over 9%. Ginna and Shape features using an SVM RBF kernel was the best performing combination in the work by Libby and Stentz [15]. The worst performance was from Ginna features using an SVM RBF kernel. It can also be seen that the feature sets containing MFCCs show comparatively better results than the others.

Our DCNN yields an overall accuracy of $97.36 \pm 0.12\%$, which is a substantial improvement over the hand-crafted feature sets. We get an improvement of 7% over the best performing Cepstral features and 12% over Ginna and Shape features using the same clip length of 300 ms. Furthermore, using a clip window size of 500 ms, our network achieves an accuracy of 99.41%, a 9% improvement over the best performing baseline approach. This strongly demonstrates the potential for using sound to classify vehicle-terrain interactions in a variety of environments.

5.2 Overall DCNN Performance

To further investigate classification performance of our network we computed the confusion matrix, which helps us understand the misclassifications between the classes. Fig. 4 shows the confusion matrix for ten-fold cross validation.

The best performing classes were carpet and asphalt, while the most misclassified was offroad and paving, which were sometimes confused with each other. Both these classes have similar spectral responses when the clip window gets smaller than 500 ms. Our system still outperforms all baseline approaches by wide margin. We also compared the per-class recall as it gives an insight on the ratio of correctly classified instances. Fig. 5 shows the per-class recall using ten-fold cross validation. The network achieves an overall recall of 97.61%.

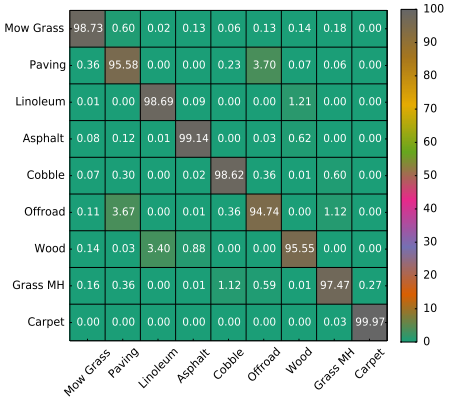

Fig. 4 Confusion matrix of our approach for ten-fold cross validation, using an audio clip length of 300 ms. The network seemed to get mostly confused with Offroad and Paving, as well as Linoleum and Wood.

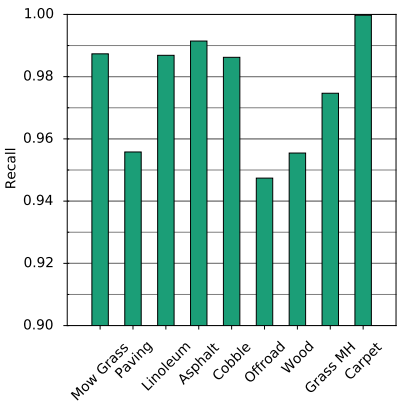

Fig. 5 Per-class recall of our network on ten-fold cross validation, using an audio clip length of 300 ms. The class with the lowest recall was Paving.

5.3 Varying Clip Length

We compared the average cross-validated accuracy of our network using varying audio clip lengths and execution times. Each clip is essentially a new sample for classification, therefore the shorter the clip, the higher is the rate at which we can infer the terrain. In addition, the shorter the clip, the faster is the execution time. For an application such as ours, fast classification and execution rates are essential for making quick trafficability decisions. Table 2 shows the overall classification accuracy using the DCNN approach with various window sizes.

Table 2 Classification accuracy of our system at varying audio clip lengths and the corresponding time taken to process though the pipeline.

Clip Length (ms)	2000	1500	1000	500	300
Accuracy (%)	99.86	99.82	99.76	99.41	**97.36**
Time (ms)	45.40	34.10	21.40	13.30	**9.15**

From the above table it can be seen that the deep network approach significantly outperforms classification using hand-crafted feature sets. We get an improvement of 7% over the best performing Cepstral features and 12% over Ginna and Shape features using the same clip length of 300 ms. Furthermore, using a window size of 500 ms, our network achieves an accuracy of 99.41%, a 9% improvement over the best performing baseline approach.

5.4 Robustness to Noise

For real-world applications such as ours, robustness to noise is a critical property. However models can only be insensitive to noise up to a certain level. We analyzed the effect of Gaussian white noise on the classification performance at several SNRs as shown in Fig. 6. It can be seen that for some classes such as carpet, grass and cobble, the performance decreases exponentially at different intensities, while for others such as linoleum and asphalt, the performance seems to be affected marginally compared to others. On the other extreme, wood and paving show remarkable robustness for SNRs upto 20 dB, thereafter the performance drops to zero. This can be attributed to the fact that spectral components are much wider for the classes that show more robustness and for the -10 dB SNR, only the classes that have certain pulses still over the noise signal are recognizable.

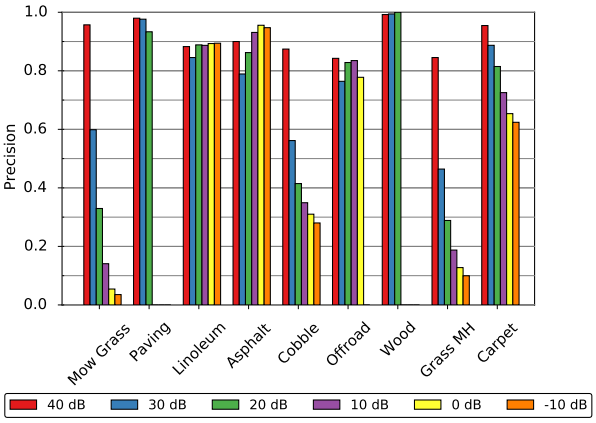

Fig. 6 Per-class precision of our network when subject to different levels of white Gaussian noise. The levels mentioned in the legend are SNRs.

As a solution to this problem, we fine-tuned our trained model on samples with additive Gaussian white noise as described in Sect. 3.2. Table 3 shows the average cross-validated recognition accuracy of our network at different SNR, before and after fine-tuning. Our fine-tuned model significantly outperforms our base model on a wide range of SNRs. The best performing classes were mowed grass, linoleum, asphalt, wood and carpet, with over 99% accuracy in all the SNRs shown in Table 3. Paving, cobble and offroad classes yielded a recognition accuracy of about 95%, averaged over all the SNRs. The only class that was slightly negatively affected by the fine-tuning was wood at SNR of 20 dB, where there was a 0.2% loss in recognition performance.

We also tested our fine-tuned model on the test set with no noise samples and the average accuracy over all the classes was 99.57%, which is a 2.21% improvement over our base models performance, clearly showing that noise adaptive fine-tuning is

Table 3 Influence of white Gaussian noise onto the classification rate. SNR is in dB and accuracy is in percent. The standard deviations were less than 1%.

SNR	40	30	20	10	0	-10
Before FT	91.42	76.45	70.66	45.06	41.91	32.01
After FT	99.49	99.12	98.56	97.97	97.09	95.90

FT = Fine-tuning

a necessary step. This improvement can be attributed to the fact that by augmenting the signals with noise samples, we provide the network some prior knowledge about the distribution of the signals which boosts the recognition performance. The only significant misclassification was in the offroad class, which was 1% of the times misclassified as paving. The other classes had almost negligible misclassifications.

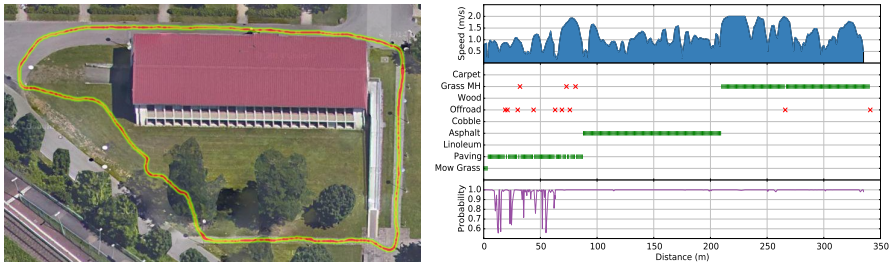

Fig. 7 The map on the left shows the trajectory taken by the robot during a classification test run using a mobile phone microphone. The variation in speed along the path is indicated in red and wider red points denote slower speed. The graph on the right shows the classification result, along with the corresponding probabilities for the path shown in the map. True positives are shown as green markers and false positives are shown are red markers.

To further stress test our network, we collected noisy samples in a new environment using a mobile phone that also tagged each sample with a GPS location. The mobile phone has a condenser microphone, which unlike the shotgun microphone that we used before, collects sounds from every direction, thereby adding considerable amount of background noise. One of the test paths that the robot traversed is shown in the map in Fig. 7. The figure also shows then variation in speed $(0-2\,\mathrm{m\,s^{-1}})$ along the path. Thicker red lines in the map, indicate slower speed. Our network achieved an accuracy of 98.54% on the mobile phone dataset. This shows the recognition robustness, not only to real-world environments but also invariant to the type of microphone. In addition, the graph in Fig. 7 shows the false positives and true positives along the traversed path. It can be seen that most of the false positives are in the paving class and this primarily occurs when the speed is above $1\,\mathrm{m\,s^{-1}}$ and the height of the paving is highly irregular, thereby misclassifying as offroad. Interestingly, there is also significant fluctuations in the class probabilities of the false positives along the paving path when the speed is below $1\,\mathrm{m\,s^{-1}}$.

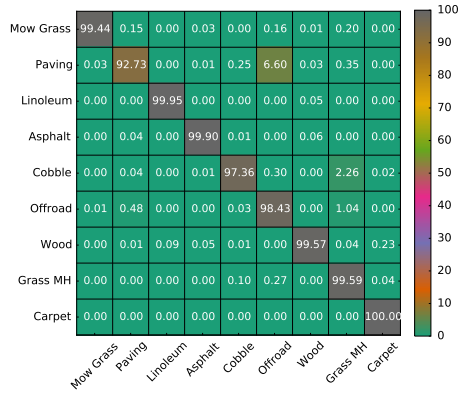

Fig. 8 Confusion matrix for classification runs using data from a mobile phone microphone. Paving and Cobble show decreased performance due to false positives with Offroad and Grass.

Fig. 8 shows the confusion matrix for the entire mobile phone microphone dataset which contains about 2.15 hours of audio data. The classes that show a dip in performance are paving, cobblestones and offroad. The paving class shows a non-negligible false positive rate as it is often misclassified as offroad. Part of this misclassification is due variation in speed and the false positives in the terrain transition boundaries.

6 Conclusion

In this paper, we introduced a novel approach that uses only sound from vehicle-terrain interactions to robustly classify a wide range of indoor and outdoor terrains. We evaluated several baseline audio features and presented a new deep convolutional neural network architecture that achieves state-of-the-art performance in proprioceptive terrain classification. Our GPU-based implementation operates on 300 ms windows and is 1,800 times faster than real-time, i.e., our system can classify a years worth of audio data in roughly 4.8 hours. Additionally, our experiments in classifying audio signal corrupted with white Gaussian noise demonstrate our networks robustness to a great extent. We additionally show that our network fine-tuned with noisy samples performs exceptionally well even at very low signal-to-noise ratios. Furthermore, our empirical evaluations with an inexpensive low-quality microphone shows that our approach is invariant to the type of microphone and can handle significant amount of real-world noise.

Acknowledgements This work has been partly supported by the European Commission under the grant numbers ERC-AGPE7-267686-LifeNav and FP7-610603-EUROPA2, and from the Ministry of Science, Research and Arts of Baden-Württemberg (Az: 32-7545.24-9/1/1) as well as by the German Ministry for Research and Technology under grant ZAFH-AAL.

References

1. V. Brijesh , and M. Blumenstein, "Pattern Recognition Technologies and Applications: Recent Advances", IGI Global, 2008.
2. C. A. Brooks and K. Iagnemma, "Vibration-Based Terrain Classification for Planetary Exploration Rovers", IEEE Transactions on Robotics, vol.21, no.6, pp.1185-1191, Dec. 2005.
3. C. A. Brooks and K. Iagnemma, "Self-Supervised Classification for Planetary Rover Terrain Sensing", 2007 IEEE Aerospace Conference, pp.1-9, March 2007.
4. D. Ellis, "Classifying music audio with timbral and chroma features", 8th International Conference on Music Information Retrieval, 2007.
5. J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, H. Balakrishnan, "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring", 6th Annual International conference on Mobile Systems, Applications and Services, June 2008 .
6. T. Giannakopoulos, K. Dimitrios, A. Andreas, and T. Sergios, "Violence Content Classification Using Audio Features", Hellenic Artificial Intelligence Conference, 2006.
7. X. Glorot, Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", International Conference on Artificial Intelligence and Statistics, pp. 249-256, 2010.
8. R. Hadsell, S. Samarasekera, A. Divakaran, "Audio based robot control and navigation", U.S. Patent 8532863 B2, Sept 28, 2010.
9. K. He, X. Zhang, S. Ren, J. Sun, "Delving Deep into Rectifiers Surpassing Human-Level Performance on ImageNet Classification", arXiv:1502.01852, 2015.
10. G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors", arxiv:cs/1207.0580v3, 2012.
11. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding", arXiv:1408.5093, 2014.
12. P. Khunarsal, C. Lursinsap, T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching", Journal of Information Sciences, pp. 57-74, vol. 243, 2013.
13. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in Neural Information and Processing Systems 25, pp. 1097-1105, 2012.
14. H. Lee, Y. Largman, P. Pham, and A.Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks", Advances in Neural Information Processing Systems 22, pp. 1096-1104, 2009.
15. J. Libby and A. Stentz, "Using Sound to Classify Vehicle-Terrain Interactions in Outdoor Environments", 2012 IEEE International Conference on Robotics & Automation, May, 2012.
16. M. Lin, Q. Chen, S. Yan, "Network In Network", International Conference on Learning Representations, arXiv:1409.1556, 2014.
17. L. Ojeda, J. Borenstein, G. Witus, and R. Karlen, "Terrain Characterization and Classification with a Mobile Robot", Journal of Field Robotics, vol. 29, no. 1, 2006.
18. A. Oord, S. Dieleman, B. Schrauwen, "Deep content-based music recommendation", Advances in Neural Information Processing Systems 26, 2013.
19. E. Trautmann and L. Ray, "Mobility characterization for autonomous mobile robots using machine learning", Autonomous Robots, vol.30, no.4, pp.369-383, 2011.
20. G. Tzanetakis and P. Cook, "Musical genre classification of audio signals", IEEE Transaction on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, 2002.
21. C. Weiss, H. Frohlich and A. Zell, "Vibration-based Terrain Classification Using Support Vector Machines", 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.4429,4434, 9-15 Oct. 2006
22. M. C. Wellman, N. Srour, and D. B. Hillis, "Feature Extraction and Fusion of Acoustic and Seismic Sensors for Target Identification", in Proc. SPIE 3081, 1997.