Crashzam: Sound-based Car Crash Detection

Matteo Sammarco[1] and Marcin Detyniecki[1,2,3]
[1]AXA Data Innovation Lab, 48 rue Carnot, 92150, Suresnes, France
[2]Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu, 75005 Paris, France
[3]Polish Academy of Sciences, IBS PAN Newelska, 6, Warsaw 01-447, Poland

Keywords: Real-time Incident Detection, Road Safety, Smart Vehicle, Audio Signal Processing.

Abstract: Connected vehicles, combined with embedded smart computation capabilities, will certainly lead to a new generation of services and opportunities for drivers, car manufacturers, insurance and service companies. One of the main challenges remaining in this field is how to detect key triggering events. One of these crucial moments is a car accident, for which not only smart connected vehicles can improve drivers' safety as car accidents are still one of the main causes of fatalities worldwide, but also help them during minor, but very stressful moments. In this paper, we present Crashzam which is an innovative way to detect any type car accidents based on sound produced by car impact, while, so far, crash detection is only a prerogative of accelerometer sensor time series analysis, or its proxy: activation of the airbag. We describe the system design, the sound detection model, and the results based on a dataset with in-car cabin sounds of real crashes. We have beforehand built such dataset with real car accident sounds. Classification is built upon features extracted from the time and frequency domain of the audio signal and from its spectrogram image. Results show that the proposed model is able to easily identify crash sounds from other sounds reproduced in-car cabins. Moreover, considering that Crashzam can run on smartphones, it is a low cost and energy solution, contributing to the spreading of such a car safety feature and reducing delays in providing assistance when an accident occurs.

1 INTRODUCTION

Although the effort in launching road safety programs in many countries, road traffic death figures remain stable worldwide at nearly 1.2 million since 2004 (World Health Organization, 2015). Causes of such a plague are diverse: speed, drink-driving, drug-driving, unused safety belt, bad weather and road conditions, and bad car break and wheel conditions. It results essential to notify a crash as fast as possible for first aid as a correlation between delaying emergency medical care and mortality rate has been proved (Evanco, 1996).

Some car manufacturers offer for their high-end products an automatic collision notification which mainly monitors the airbag deployment to detect a severe collision and calling assistance with the embedded cellular radios. The BMW's Automatic Crash Notification System and the GM's OnStar are just two examples. These products remain mostly restricted as option to luxury market sectors and a large part of the circulating vehicles do not embed an OEM automatic accident detection and notification system.

Relatively cheaper third party solutions foreseen the installation of boxes under the hood, wind-screen boxes or OBDII dongles which embed an acceleration sensor as along as a proprietary algorithm to detect shocks. In fact, state of the art solutions employ accelerometer data to detect more or less severe impacts triggered by the sudden variation of acceleration on one or more axes (White et al., 2011; Thompson et al., 2010; Zaldivar et al., 2011; Punetha et al., 2012; Lahn et al., 2015; Aloul et al., 2015). Although the use of accelerometer sensors leads to a precise impact identification both for angle and severity, dedicated hardware must be professionally installed to achieve the maximum accuracy.

The most cost effective and practical solution, instead, relies on acceleration time series recorded on drivers' smartphones. On these bases, some mobile applications are already available on the market (Zendrive, 2017; Sosmartapp, 2017; TrueMotion, 2017). Nevertheless, smartphone data is hard to analyze due to calibration, noise and rotation issues. In addition, it is not clear the optimal frequency for acceleration samples and the time window width to record.

Sammarco, M. and Detyniecki, M.
Crashzam: Sound-based Car Crash Detection.
DOI: 10.5220/0006629200270035
In *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2018)*, pages 27-35
ISBN: 978-989-758-293-6

Also, in many situations, relying only on acceleration data may lead to false predictions: street bumps, holes and bad street conditions trigger false positives, whereas collisions coming from the back while standing still may be classified as normal accelerations.

As a trendy solution, social networks and microblogs provide a global source where to share what we experience and see around us. Car accidents usually attract people curiosity, who might post tweets and photos of the event (Schulz et al., 2013). Although this provides a zero-cost solution, many cons affect it: accidents could not be immediately advertised, people might provide confused or misleading information, necessary third party people identification comes with privacy issues. Finally, the use of smartphones during driving is a source of accident itself.

As electric lamps were not invented by improving candles, the goal of this work is not to improve accelerometer-based car crashing detection algorithms, but to detect crashes by sound.

Crashzam is an innovative way to detect car crashes based on sound recognition techniques. It does not suffer from neither calibration and sensor configuration problems nor the delay and subjectiveness of human notifications. Being in continuous listening specifically for crash sounds, our solution is able to detect whatever a car impact occurred. It can take advantage of the presence of microphones inside car cabins: hands free car kits, Bluetooth kits, car audio systems with voice command, wind-screen SOS boxes, dash cameras and smartphones are some hardware equipments embedding microphones, which are usually present in a car.

It does not matter where and with which angle respect to the horizon these devices are placed in the car cabin, since the received sound will not be affected by calibration issues. The only assumption is that a crash produces a sound.

Despite some works based on crash sound detection which are mainly designed for road surveillance purposes (Rabaoui et al., 2008; Carletti et al., 2013; Foggia et al., 2016a; Clavel and Ehrette, 2008; Valenzise et al., 2007), this work focuses on drivers' safety by detecting crash sound from inside the car cabin.

But in order to perform, Crashzam must take into account the presence of environmental noise and any other overlapping sound which could happen in a car cabin (engine rotation, car horn, radio music, etc.).

The advantage for drivers is to have a low cost instantaneous crash detection rid of accelerometer sensor discrepancies.Other actors involved in road safety, like medical assistance services and insurance companies, can take advantage of this solution for first notification of loss (FNOL) and fast first aid.

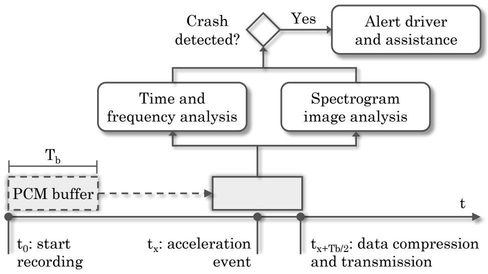


Figure 1: Crashzam high level system design: while driving, the smartphone microphone records a sliding buffer of PCM samples which are analyzed if an acceleration event occurs (e.g., acceleration norm is over a certain threshold). Accelerometer is used as a mere trigger. If Crashzam detects a crash, the driver is solicited and assistance is alerted in case of no answer from the driver.

The main contribution of this paper is the design of a low cost, smartphone-based car crash detection system through a new crash sound detection model. It can detect crash sounds and distinguish other sounds generated inside the car cabin. To this aim, we collect a novel dataset of crash sounds recorded from inside the car cabin. Then, we create and select a set of features computed from the time and frequency sound signal domain and from the sound spectrogram image. Finally we propose a combination of machine learning models for the automatic classification of sounds reproduced inside vehicles.

This manuscript is structured as follows: in Section 2 we provide a top-down description of Crashzam, while in Sections 3 and 4 we explore in detail the model to detect crashes from audio clips. In Section 5 we present the dataset that we have built and that we have used to train and test our model. In Sections 6 and 7 we discuss about the obtained results and the performances in according to the model parameters. Finally, we give our conclusions and perspectives in Section 8.

2 SYSTEM DESIGN

Considering Crashzam running on a smartphone application, we show in Figure 1 the high level system design. Android and iOS operating systems provide an activity recognition mechanism which starts at time t_0 the recording of a PCM samples buffer simultaneously to the detection of driving activity. The sliding buffer has a duration of T_b seconds. In our proof of concept we set $T_b = 5$ seconds, with PCM sampled at 16kHz and quantized at 16 bits at least. At t_0, also the location service (GPS) and the accelerometer sensor are activated. We use the accelerometer as a mere trigger: if at time t_x the norm of the three acceleration axes exceeds $2.5g$, then we continue recording for $T_b/2$ seconds before analyzing data. In this way we use half of the buffer to record what happened before a probable impact event and the other half to record what happened after it. In fact, a typical crash event includes tire skidding or horn sounds before the crash and human screams before or after it, although the crash itself will present the maximum signal amplitude. Thus, in the following we train our models to distinguish between car impact sounds which are high energy, percussive sounds and other sounds which are likely to be produced during a crash but not necessarily like car horn or harsh deceleration sounds. It is worth to note that Crashzam can also work in combination with a traditional accelerometer-based car crash detection.

At time $t_x + T_b/2$, the array of PCM is compressed and transmitted to the server as along as the last GPS position, as a JSON object to a server dedicated to analyze the recorded sound. Data is independently analyzed by two models detailed in Sections 3 and 4. Finally, both results are combined by a weighted voting classifier. The first model focuses on time and frequency aspects of the sound. It mainly detects high energy, abrupt changing sounds. But since car horn or engine starting are such kind of sound too, we include a second model based on spectrogram image analysis calibrated on the detection of percussive, high energy, hollow sounds, which correspond to crashes.

A side effect feature of having two independent models, is that the system is modular and the first model can be reused in other application fields (e.g., glass breaking detection).

The result of the classification $\hat{y}$ is based on the probability output of the two models $j \in 1,2$ in according to the relation:

$$\hat{y} = argmax_i \sum_{j=1}^{2} w_j p_{ij}, \qquad (1)$$

where $i \in 0,1$ represents the sound class (“Other” or “Crash” sound).

At the end of the classification pipeline, if a crash is detected, the driver is solicited for an interaction (e.g., pushing a safe button or answering to a call), otherwise assistance is alerted.

We have monitored the battery usage on a Nexus 5 smartphone when activating motion sensors (accelerometer, gyroscope, and magnetometer), and recording audio signals. Nexus 5 is equipped with a 2300 mAh, 3.8 V battery and an Android Nougat OS. Figure 2 shows that the natural discharging from 100% to 5% lasts four hours when all not vital service but the screen are switched off. Activating motion sensors only, battery life is 30 minutes shorter, while adding audio recording and motion sensors it becomes 50 minutes shorter. Knowing that the battery used for the test can provide at most 2300 mA in an hour, we conclude that the microphone consumed about 80 mA per hour which corresponds to 3.5% of the battery capacity.

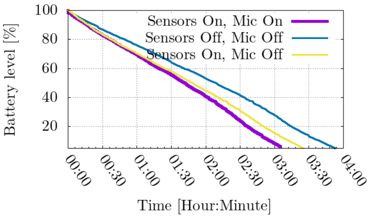


Figure 2: Nexus 5 battery discharging when activating or not motion sensors and microphone.

3 FIRST MODEL: TIME AND FREQUENCY ANALYSIS

A large set of metrics and features can be extracted from audio signals (Peeters and Rodet, 2004). For the first model, we have selected here some time and frequency-based features largely adopted in the literature for tasks of audio event detection or music genre classification.

3.1 Time Domain Features

Let us consider $x(t)$ a discrete audio signal of $\mathcal{N}$ samples.

- **Zero-Crossing Rate** (ZCR)**.** The rate a discrete signal $x(t)$ changes sign during its duration N is a key feature to recognize percussive sounds (Gouyon et al., 2000b; Gouyon et al., 2000a).

$$ZCR_x = \frac{1}{2N} \sum_{t=0}^{N-2} |sgn(x(t+1)) - sgn(x(t))|, \quad (2)$$

where $sgn(x(t)) = \begin{cases} 1, & \text{if } x(t) \geq 0, \\ -1, & \text{if } x(t) < 0. \end{cases}$

- **Signal Power.** It is the sum of squares of the signal values normalized by the signal length.

$$P_x = \frac{1}{N} \sum_{t=0}^{N-1} |x(t)|^2 \qquad (3)$$

We expect crash sounds to show high power.

- **Entropy.** Entropy of a discrete random variable X with possible values $x_1,...,x_n$ and probability function $P(X)$ is usually defined as:

$$H(x) = -\sum_{i=1}^{n} P(x_i)\log_b P(x_i). \qquad (4)$$

We consider $P(x_i) = e_j = \frac{E_{frame_j}}{E_x}$, where $E_x = \sum_{t=0}^{N-1} |x(t)|^2$ is the signal energy and E_{frame_j} is the energy of the j^{th} of K fix-sized sub-frames the signal is split into. Thus, our entropy becomes:

$$H(x) = -\sum_{j=1}^{K} e_j \log_2(e_j). \qquad (5)$$

Entropy is usually interpreted as a measure of abrupt changes in energy (Pikrakis et al., 2008; Giannakopoulos et al., 2007). We expect crash sounds having high entropy.

3.2 Frequency Domain Features

It is often useful to analyze discrete signals in the frequency domain through a Discrete Fourier Transform (DFT). The original signal is split in fixed-size smaller frames, and the DFT is applied on each frame returning an array of coefficients having the same length of the number of samples in the frame. Let us consider $X_i(k), k = 1,\ldots,M$, the magnitude of DFT coefficients of the i^{th} frame. We compute average and standard deviation over all frames.

- **Spectral Centroid** (*SC*)**.** For the i^{th} frame, *SC* is the average of frequencies present in the signal, weighted by their amplitudes:

$$SC_i = \frac{\sum_{k=1}^{M} kX_i(k)}{\sum_{k=1}^{M} X_i(k)}, \qquad (6)$$

The *SC* represents the barycenter of the spectrum and higher values correspond to brighter sounds (Grey and Gordon, 1978). Crash sounds have a low *SC*.

- **Spectral Spread** (*SS*)**.** It represents the deviation from the *SC*:

$$SS_i = \sqrt{\frac{\sum_{k=1}^{M}(k - SC_i)^2 X_i(k)}{\sum_{k=1}^{M} X_i(k)}} \qquad (7)$$

Low values of *SS* correspond to signals whose spectrum is concentrated around the spectral centroid. Usually crash sounds present a high *SS* value.

- **Spectral Flux** (*SF*)**.** *SF* represents the spectral change by comparing the power spectrum of two consecutive frames:

$$SF_{i,i-1} = \sum_{k=2}^{M}\left(\frac{X_i(k)}{\sum_{l=1}^{M} X_i(l)} - \frac{X_{i-1}(k)}{\sum_{l=1}^{M} X_{i-1}(l)}\right)^2 \qquad (8)$$

SF is mainly used for onset detection, thus applicable to crash detection too (Dixon, 2006).

- **Spectral Rolloff** (SR)**.** The *SR* is the frequency below which 90% of the magnitude distribution of the spectrum is concentrated. *SR* is the frequency which satisfies the following relation:

$$\sum_{k=1}^{SR} X_i(k) = 0.9 \sum_{k=1}^{M} X_i(k). \qquad (9)$$

It is useful to discriminate sounds like human voice signals whose energy is concentrated under 4 kHz and music.

- **Spectral Entropy** (*SE*)**.** Similarly to the entropy in the time domain, let us consider the spectrum divided in k fixed-size frequency sub-bands, *SE* is:

$$SE = -\sum_{i=1}^{k} P_i \log_2(P_i), \qquad (10)$$

where $P_i = \frac{E_i}{\sum_{i=1}^{k} E_i}$ and E_i is the energy in the i^th sub-band. In crash sounds, *SE* should have low values as the energy is spread on all the sub-bands.

- **Chroma Vector** (CV)**.** With the Chroma Vector we group all the DFT coefficients into 12 bins corresponding to the 12 pitches of an equal tempered scale. Each element of the vector is the mean of DFT coefficients:

$$v_k = \sum_{f \in F_k} \frac{X_i(f)}{|F_k|}, \quad k \in [1-12], \qquad (11)$$

where F_k is the set of frequencies included in the same bin. CV is widely used for audio matching (Kurth and Müller, 2008; Mueller et al., 2005).

- **Mel-Frequency Cepstral Coefficients** (MFCCs)**.** MFCCs are 13 coefficients forming a cepstral representation where the frequencies are distributed according to the mel scale. The mapping between the mel and frequency scale using k triangular overlapping windows is the following:

$$Mel(f) = 2595\log_{10}(1+\frac{f}{100}). \tag{12}$$

Then, MFCCs are computed as the discrete cosine transform (DCT) of cepstrum powers at each mel frequency:

$$c(k) = DCF\log|DFTm(n)|. \tag{13}$$

MFCC are useful to analyze abrupt changes in the spectrum and widely employed in human speech domain where they are particularly effective (Gonzalez, 2013; Sengupta et al., 2016; Ganchev et al., 2005; Müller, 2007).

4 SECOND MODEL: SPECTROGRAM IMAGE ANALYSIS

The second model is based on spectrogram image features and it is specifically designed to discern between percussive and sustained sounds. The former have high amplitude values distributed on all the frequencies at a certain time, while the latter present high values of amplitude at certain frequencies for long time. The intuition behind this model is that a crash is a percussive sound, while many other sounds reproduced in car like car horn, harsh acceleration or tire skidding are sustained sounds.

4.1 Specific Spectrogram Image Features

Starting from a spectrogram images like the ones in Figure 3, we extract the amplitude matrix and we select a constellation of peaks, which are points, located in time and frequency, exceeding a certain amplitude threshold A_{th}. Such peaks are local maxima, meaning that in a region with radius $R = 2*D+1$, where D is the maximum Manhattan distance from the center of the region, they show the maximum value. If more points in a region are candidate to be a peak (i.e., they have the same amplitude), all of them are selected. Thus, we convert the amplitude matrix to a matrix P which localizes a constellation of peaks p_{ft} and we analyze the patterns created by such peaks. P is a $F \times T$ binary, sparse matrix, where $f \in F$ denotes a frequency in the range $[0, \frac{sampling\ \ rate}{2}]$ and $t \in T$ denotes a small time bin the original signal is split into to compute the DFT.

A considerable advantage considering peaks extracted from the spectrogram is that we filter out noise and background sounds.

- **Peaks Vertical Alignment.** Given the matrix of peaks, we get the mean μ_V and standard deviation σ_V from the distribution of frequency gaps among peaks in the same time bin.

$$\mu_V = \frac{1}{N_v}\sum_{t\in T}\sum_{j=1}^{|I|-1}(I_{j+1}-I_j), \tag{14}$$

where $I = \{i \in F | p_{it} \neq 0 \ \ and \ \ \sum_i p_{it} > 1\} \quad \forall t$, and $N_v = \sum_{t\in T}[(\sum_{f\in F} p_{ft}) - 1]$.

$$\sigma_V = \sqrt{\frac{1}{N_v}\sum_{t\in T}\sum_{j=1}^{|I|-1}[(I_{j+1}-I_j)-\mu_V]^2}. \tag{15}$$

Being a percussive sound, a crash will show peaks stacked on the same time bin along all the frequency range as shown in Figure 3(a). Thus, we expect low average and standard deviation values.

- **Peaks Horizontal Alignment.** Given the matrix of peaks, we get the mean μ_H and standard deviation σ_H from the distribution of time delays among peaks at the same frequency.

$$\mu_H = \frac{1}{N_h}\sum_{f\in F}\sum_{i=1}^{|J|-1}(J_{i+1}-J_i), \tag{16}$$

where $J = \{j \in T | p_{fj} \neq 0 \ \ and \ \ \sum_j p_{fj} > 1\} \quad \forall f$, and $N_h = \sum_{f\in F}[(\sum_{t\in T} p_{ft}) - 1]$.

$$\sigma_H = \sqrt{\frac{1}{N_h}\sum_{f\in F}\sum_{i=1}^{|J|-1}[(J_{i+1}-J_i)-\mu_H]^2}. \tag{17}$$

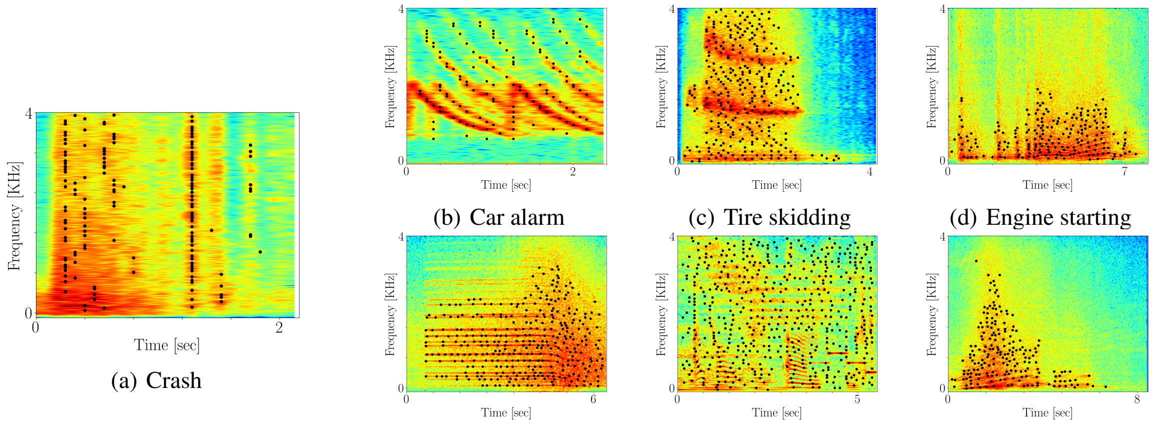
Sounds such as car horn, tire skidding, and harsh acceleration are sustained sounds and thus they will present horizontal stripes of peaks on the spectrogram as shown in Figures 3(c) and 3(e).

- **Peak Entropy.** Entropy of the peak constellation is computed similarly to the entropy in the time and the frequency domain. We split the spectrogram in k time bins and for the i^{th} bin, we compute the ratio between peaks included in that bin and all the peaks in the constellation: $e_p = \frac{\sum peaks_i}{\sum peaks}$.

$$PE = -\sum_{i=1}^{k} e_p \log_2(e_p). \tag{18}$$

5 DATASET

In Spring 2016, AXA Winterthur has set the annual crash test campaign in Switzerland. During this occasion, the AXA Data Innovation Lab collected about



(a) Crash (b) Car alarm (c) Tire skidding (d) Engine starting (e) Car horn (f) Radio Music (g) Harsh acceleration

Figure 3: Example of spectrograms with constellation of peaks.

6.2 GB of data, including audio files recording crashes from inside the car cabins. This original dataset is composed by 46 signals of crash and, to the best of our knowledge, it is the only dataset collecting crash sounds from inside the car cabin[1]. Moreover, each test was controlled in impact speed and angle, thus such dataset has been essential to study crash dynamics.

Therefore, to extend the set of controlled positive samples, we extracted sounds recorded by dash cameras published on the "Car Crash Time" Youtube channel which provides several hours of crash recordings (Car Crash Time, 2017). With respect to the controlled dataset, such sounds are very realistic and genuine, including background noises like rain, hail or screams before the impacts. Overall, we collected 410 crash sounds recorded inside the cabin of cars involved in the shock. Most of them (87%) come from "Car Crash Time" and the rest from the AXA Winterthur crash test campaign. We kept 100 samples aside as test set.

As negative samples, we choose to include in the dataset any sound which might be generated or listened to inside a car cabin like radio music, people talking, engine starting, car door opening or closing. We also included sounds that are often linked to car shocks like car horn, harsh decelerations, and tire skidding to control the false positive rate. Such sounds have been mostly imported from FreeSound (FreeSound, 2017) and Urban Sound Dataset (Urban Sound Dataset, 2017). Although gathering negative sounds is a simpler task compared to the positive samples, we collected an equal number of samples in order to have a balanced dataset. Table 1 summarize the dataset distribution into categories "Crash" and "Other" and sub-categories.

Table 1: Dataset distribution in classes and sub-classes.

Class	Sound type	%
Crash	AXA Winterthur crash campaign	13
	Car Crash Time	87
Other	Harsh acceleration or deceleration	10
	Car horn	10
	Car door opening and closing	8
	Radio music	11
	People talking	14
	Tire skidding	10
	Car alarm	5
	Rain, hail, strong wind	10
	Engine during driving	22

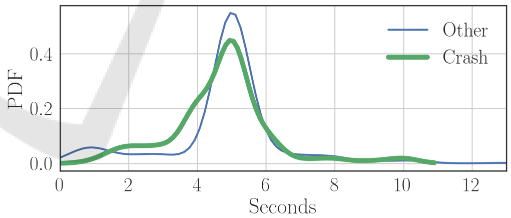


Figure 4: PDF for samples duration in the dataset.

All audio clips are sampled at 16 KHz, quantized at 16 bits and all amplitudes are normalized. They represent a mix of collision types: vehicle to barrier, vehicle to vehicle, frontal impact, side impact, and at different speeds.

A crash can occur between two or more cars, thus it can last more or less time. Figure 4 shows the distribution of sample durations. The vast majority of crashes involve only two cars and last five seconds,

[1] There exist a dataset of sounds provided by the MIVIA Computer Science department of Università di Salerno (Italy) for research purposes which includes crash sounds recorded only from outside the car cabin (Foggia et al., 2016b). Thus, such sounds were not eligible to include in our dataset. Nevertheless it also contains tire skidding and scream sounds.

while the longest crash can last also the double. We selected negative samples in order to follow the same distribution as crash sounds.

6 EVALUATION

Both time-frequency domain and spectrogram image classifiers are random forest classifiers trained with cross validation using the Python scikit-learn library. Number of trees are 200 and 100 respectively while all the other parameters are left unchanged.

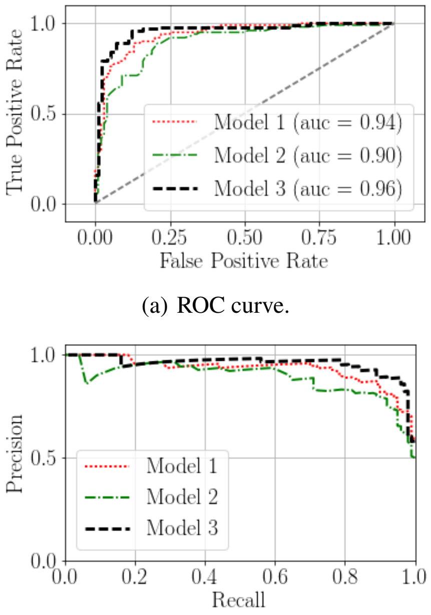
Figures 5 shows ROC and precision-recall curves for classification done by the two models independently and by combining their result probabilities together. The time and frequency based classifier (Model 1) performs in line with other models presented in the literature on sound detection like gun shots or human screams for surveillance purposes (Crocco et al., 2016). Its accuracy $(\frac{\sum(True \quad Positive)+\sum(True \quad Negative)}{\sum Samples})$ is equal to 0.875. Nevertheless, it is wrong for some specific sub-types of sounds with abrupt changes and high energy like car horn or tire skidding.

As the spectrogram based classifier (Model 2) is specifically designed to discern between sustained and percussive sounds, it is able to help the first models in such ambiguous situations. Although in absolute terms Model 2 performs worse than Model 1, the combination of the two (Model 3) in according to Equation 1 brings the overall accuracy to 0.9. Models weights in Equation 1 are set to $w_1 = 0.6$ and $w_2 = 0.4$ respectively. They come from an exhaustive research to obtain the best results.

Most of the final misclassifications correspond to slammed door sounds.

7 DISCUSSION

Creating the constellation of peaks is highly dependent from the spectrogram image in background. Spectrogram is created spitting the time series of PCM in small windows having a certain overlap and computing the Fourier transform in each window. Being samples also quantized at 16 bits, the constellation of peaks will be depended by the number of samples per window (NFFT) and by the overlap between consecutive windows. A short window will tend to produce many peaks at the same frequency since the same sound amplitude will be replicated for many windows. Having a large overlap will have the same impact. On the other hand, a large window will increment horizontal peak gaps, but will tend to reduce vertical frequency gaps among peaks. Figure 6 shows how the accuracy on the testing dataset changes, varying both NFFT and the overlap, where the overlap is at most equal to NFFT. The impact of such parameters is quite important since the accuracy scale is in the range [0.65-0.82]. Intermediate values for NFFT, 256 or 512, introduce the least offset and they are more independent to the overlap, thus we choose $NFFT = 512$ and $overlap = 354$. Once the NFFT and overlap set, the accuracy varies a little changing the threshold and the distance to find the local peaks. We have set a threshold $A_{th} = 50 \quad dB$ and a minimum distance $D = 3$.



(a) ROC curve.

(b) Precision-recall curve.

Figure 5: Comparison of accuracy, precision and recall among singular models (Model 1 and 2) and the combination of such models in according to Equation 1 (Model 3).

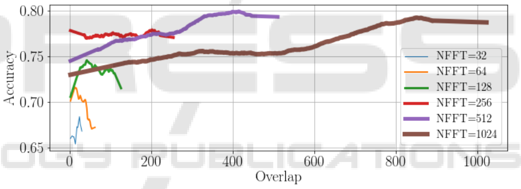


Figure 6: Effects on classification tuning the number of data points used in each block for the Fast Fourier Transform (NFFT) to create the spectrogram and the number of overlapped points.

8 CONCLUSION AND PERSPECTIVES

This manuscript introduces Crashzam, an innovative way to detect car accidents with sound recorded by smartphones or any other microphone-equipped smart device installed in cars. The goal is to enhance drivers' safety with a low cost solution, capable to propose domain specific services like medical assistance calling, first notification of loss (FNOL) or advice and coordination during a minor event.

Sounds are analyzed by two models and the final detection result reflects a combination of both of them. The first model analyses the signal in time and frequency, computing well-known features in the sound recognition literature. The second one is based on the analysis of spectrogram images and the discovery of patterns among a constellation of amplitude peaks.

As no dataset of crash sounds recorded inside the car cabin existed, we built one. Crash samples come from both controlled experiments and real, genuine conditions.

Detection results show a pretty accurate classification between crash sounds and other sounds likely to be reproduced in car. Also, combining models analyzing specific aspects of sound signals (time series, frequency, spectrogram) helps the system to be more accurate and reliable when the environment is noisy.

A wide range of perspectives are possible. We proposed Crashzam in the context of driving safety and connected car, but the same concept could be applied to other domains such as the connected home and the connected health. For instance, Minut, a Scandinavian startup, is specialized on sound-based home surveillance (Minut, 2017). Their devices are able to detect alarm or glass breaking. Remaining in a the telematics context, the detection of repetitive car horn sounds gives an insight on the drivers' driving style.

As regards the system design, the tendency is to embed models into devices. For instance, one of the most advertised novelty in the 2017 Google I/O was the possibility to embed TensorFlow models inside smartphones (Android and iOS) (Google I/O '17, 2017). In the safe driving context, having an autonomous system that gets rid of server calling would mean a full time detection availability.

Deep learning has gained more and more estimation on image and audio classification tasks. Spectrogram image processing may be tested with Convolutional Neural Networks to discover patterns. Nevertheless, in some working domains like insurance, it is essential to know how to explain the results.

ACKNOWLEDGEMENTS

The authors would like to thank AXA Winterthur for inviting the AXA Data Innovation Lab to participate to the Spring 2016 crash test campaign and AXA Singapore for helping with the data crawling and cleaning.

REFERENCES

Aloul, F. A., Zualkernan, I. A., Abu-Salma, R., Al-Ali, H., and Al-Merri, M. (2015). ibump: Smartphone application to detect car accidents. *Computers & Electrical Engineering*, 43:66–75.

Car Crash Time (2017). Youtube channel. https://www.youtube.com/channel/UCwuZi_C_yFtHsfCicthqygw. Accessed: 2017-05-03.

Carletti, V., Foggia, P., Percannella, G., Saggese, A., Strisciuglio, N., and Vento, M. (2013). Audio surveillance using a bag of aural words classifier. *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2013*, (December):81–86.

Clavel, C. and Ehrette, T. (2008). Fear-type emotion recognition and abnormal events detection for an audio-based surveillance system. *WIT Transactions on Information and Communication Technologies*, 39:471–479.

Crocco, M., Cristani, M., Trucco, A., and Murino, V. (2016). Audio Surveillance: a Systematic Review. *ACM Computing Surveys*, 48(4):1–44.

Dixon, S. (2006). Onset Detection Revisited. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, pages 133–137, Montreal, Quebec, Canada.

Evanco, W. M. (1996). The impact of rapid incident detection on freeway accident fatalities. Technical report.

Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., and Vento, M. (2016a). Audio surveillance of roads: A system for detecting anomalous sounds. *IEEE Transactions on Intelligent Transportation Systems*.

Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., and Vento, M. (2016b). Audio surveillance of roads: A system for detecting anomalous sounds. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):279–288.

FreeSound (2017). On-line free sound samples. www.freesound.org. Accessed: 2017-05-03.

Ganchev, T., Fakotakis, N., and Kokkinakis, G. (2005). Comparative evaluation of various mfcc implementations on the speaker verification task. In *in Proc. of the SPECOM-2005*, pages 191–194.

Giannakopoulos, T., Pikrakis, A., and Theodoridis, S. (2007). A multi-class audio classification method with respect to violent content in movies using bayesian networks. In *IEEE 9th Workshop on Multimedia Signal Processing, MMSP 2007, Chania, Crete, Greece, October 1-3, 2007*, pages 90–93.

Gonzalez, R. (2013). Better than MFCC audio classification features. *The Era of Interactive Media*, pages 291–301.

Google I/O ’17 (2017). Android meets TensorFlow: How to accelerate your app with AI. https://www.youtube.com/watch?v=25ISTLhz0ys. Accessed: 2017-05-03.

Gouyon, F., Delerue, O., and Pachet, F. (2000a). Classifying percussive sounds: a matter of zero-crossing rate ? In *Proceedings of DAFX 00*, Verona (It).

Gouyon, F., Pachet, F., and Delerue, O. (2000b). On the use of Zero-Crossing rate for an application of classification of percussive sounds. *International Conference on Digital Audio Effects*, (August 2002):3–8.

Grey, J. M. and Gordon, J. W. (1978). Perceptual effects of spectral modifications on musical timbres. *The Journal of the Acoustical Society of America*, 63(5):1493–1500.

Kurth, F. and Müller, M. (2008). Efficient index-based audio matching. *IEEE Trans. Audio, Speech & Language Processing*, 16(2):382–395.

Lahn, J., Peter, H., and Braun, P. (2015). Car crash detection on smartphones. In *Proceedings of the 2Nd International Workshop on Sensor-based Activity Recognition and Interaction*, WOAR ’15, pages 12:1–12:4, New York, NY, USA. ACM.

Minut (2017). Home surveillance. https://minut.com/. Accessed: 2017-05-03.

Mueller, M., Kurth, F., and Clausen, M. (2005). Audio Matching via Chroma-Based Statistical Features. pages 288–295, London, UK. University of London.

Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Peeters, G. and Rodet, X. (2004). A large set of audio feature for sound description (similarity and classification) in the cuidado project. Technical report, Ircam, Analysis/Synthesis Team, 1 pl. Igor Stravinsky, 75004 Paris, France.

Pikrakis, A., Giannakopoulos, T., and Theodoridis, S. (2008). Gunshot detection in audio streams from movies by means of dynamic programming and bayesian networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 21–24.

Punetha, D., Kumar, D., and Mehta, V. (2012). Article: Design and realization of the accelerometer based transportation system (ats). *International Journal of Computer Applications*, 49(15):17–20. Full text available.

Rabaoui, A., Davy, M., Rossignol, S., and Ellouze, N. (2008). Using one-class SVMs and wavelets for audio surveillance. *IEEE Transactions on Information Forensics and Security*, 3(4):763–775.

Schulz, A., Ristoski, P., and Paulheim, H. (2013). I see a car crash: Real-time detection of small scale incidents in microblogs. In *Extended Semantic Web Conference*, pages 22–33. Springer.

Sengupta, N., Sahidullah, M., and Saha, G. (2016). Lung sound classification using cepstral-based statistical features. *Computers in Biology and Medicine*, 75:118–129.

Sosmartapp (2017). Automatic car crash detection app. http://www.sosmartapp.com. Accessed: 2017-05-03.

Thompson, C., White, J., Dougherty, B., Albright, A., and Schmidt, D. C. (2010). *Using Smartphones to Detect Car Accidents and Provide Situational Awareness to Emergency Responders*, pages 29–42. Springer Berlin Heidelberg, Berlin, Heidelberg.

TrueMotion (2017). Driving Intelligence. https://gotruemotion.com/. Accessed: 2017-05-03.

Urban Sound Dataset (2017). https://serv.cusp.nyu.edu/projects/urbansounddataset/urbansound8k.html. Accessed: 2017-09-03.

Valenzise, G., Gerosa, L., Tagliasacchi, M., Antonacci, F., and Sarti, A. (2007). Scream and gunshot detection and localization for audio-surveillance systems. *2007 IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007 Proceedings*, pages 21–26.

White, J., Thompson, C., Turner, H., Dougherty, B., and Schmidt, D. C. (2011). Wreckwatch: Automatic traffic accident detection and notification with smartphones. *Mobile Networks and Applications*, 16(3):285–303.

World Health Organization (2015). Global status report on road safety.

Zaldivar, J., Calafate, C. T., Cano, J. C., and Manzoni, P. (2011). Providing accident detection in vehicular networks through obd-ii devices and android-based smartphones. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pages 813–819. IEEE.

Zendrive (2017). Safer Drivers, Safer Roads. http://www.zendrive.com/. Accessed: 2017-05-03.