DOMESTIC CATS SOUND CLASSIFICATION USING DEEP LEARNING

Àlex Solé, Eloi Moliner

Barcelona School of Telecommunications Engineering (ETSETB)

ABSTRACT

Domestic cats are animals able to vocalise a wide range of sounds with different and distinguished meanings. Recognising these sounds could be really useful to help us humans to determine the mood of our feline friends. This work consists on processing cat sounds and automatically classify them, using a dataset containing sounds within a range of 10 categories. Our approach is to compute the mel-spectogram for every sound and classify them using a convolutional neural network. Since the dataset is not large enough to perform a good training, a data augmentation process has been carried out. We have tested four state-of-the-art pre-trained CNNs and compared the results between them. The one which has shown better results is the ResNet50 with an 86% of accuracy.

Index Terms— cat sounds, audio classification, CNN, deep learning, data augmentation, transfer learning

1. INTRODUCTION

Cats (*Felis catus*) are one of the most popular domesticated animals in the world. People love cats, they are warm, cuddly and communicative but just wild enough to remind us of the magnificence of the animal kingdom that lies right outside our doors. Cats use an elaborate system for communicating with each other and other species. Body language, scent, touch, and sound are all important factors in communication that can help us humans to determine their mood.

One of the most important factors in cats communication (in which this work is focusing) is vocalisation. The pitch, intensity, frequency, rapidity and volume of a cat's meow reflect their different emotional states and physical needs. Cats use sound for different reasons, to communicate between offspring, to communicate with adults in their territory or with other species such as humans. Purring indicates a calm relaxed state although some cats also purr when they are in pain. Vowel patterns indicate a need for food or other desires. Loud strained intense sounds including hissing, screaming and growling are often associated with mating or aggressive behaviours.

Nicholas Nicastro and Michael J. Owren showed in [1] that cat sounds can be understood and carefully classified by humans with wide experience treating with them. Recently, Yagya Raj Pandeya and his colleagues [2], [3] elaborated an impressive work automatically classifying cat sounds using deep neural nets. As far as we are concerned, no one else had performed an study involving pet animals sound recognising, specially cat sounds.

Our work consists on automatically classifying sounds produced by cats. We are based on Pandeya et al. papers [2] [3], using the CatSound dataset provided by them, which includes a variety of ten different cat sound classes. Our contribution is experimenting with a set of state-of-the-art deep learning architectures. In section 2 the techniques carried on this project are explained in detail and, in section 3 the experiments and results are exposed. Finally the conclusions are mentioned in section 4.

2. TECHNIQUES

2.1. CatSound Dataset

The CatSound dataset was created by Yagya Raj Pandeya and Joonwhoan Lee, it is the fruit of an extensive work collecting cat sounds from different sites on the internet mostly from YouTube and Flicker. The audios were cut out in small recordings of a few seconds, in a way that only the meaningful information is preserved. The audios were accurately categorised according to ten different sound classes, figure 1 shows all the categories present in the dataset and their respective percentages. As can be noticed, the dataset is almost perfectly balanced, the number of files in each category are nearly 300, that is a percentage of 10% for each class.

2.2. Data augmentation

Even though the dataset has requested an impressive amount of time to the creators, it is not large enough to train a neural network without having over-fitting issues. Therefore, a data augmentation process has been applied to the dataset. Data augmentation consists on augmenting the number of files in the dataset by slightly modifying the original ones. The objective was to increase substantially the number of files in the augmented dataset, but having in mind that it is important to create some variation in the new files, otherwise the augmentation would be useless. In these cases, there is a trade-off about how much you "distort" the signals to create new ones.

Thanks to Yagya Raj Pandeya for providing the CatSound dataset.

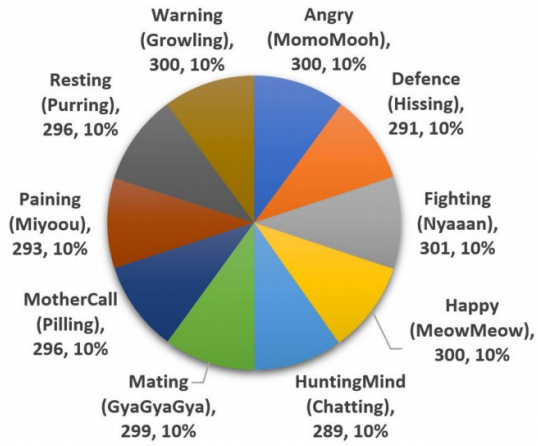

Fig. 1. Content of the CatSound dataset, extracted from [2]

Varying too less would probably cause over-fitting to the network in the training process, but varying too much would generate unreal sounds that would distract the network with sounds that are too different from the real ones in their category.

Since we have decided to have a huge percentage of augmentation (x31), we have chosen to define a wide parameter space with a quite large list of varying parameters. The parameters are the following:

- Time shift
- Time stretch
- Pitch shift
- White noise addition (with the noise deviation as a parameter)
- Dynamic range compression (4 different parameters)

We have used the *LibROSA* package for the time shift and time stretch and *SoX* for the dynamic range compression, explained in detail in appendix A.

Having many parameters is a safe way of ensuring variation between the new generated samples, this is way it is more difficult to have too similar sounds since there are more independent ways in which they can differ. Even though, we have to be careful with how these variations are performed, we do not want to generate a file were all the parameters have varied a lot from the original and the result is unreal. To prevent this, we defined the variation of each parameter (excepting the time shift) as Gaussian random variables, the means and deviation of each of them were chosen ad hoc.

Since the CatSound dataset has a total of approximately 3000 audio files, after applying a data augmentation of x31 we have an augmented dataset of 93000 files with 9300 sounds in each category.

2.3. Pre-processing

This subsection explains the processing of the audio files before they are ready to be fed to a neural network. Our approach consists in computing the mel-spectogram of the audios.

Firstly, the lengths of all the audios are standardised to 200.000 samples (4.5 seconds). The mel-spectogram was computed using a frame size of 2048 samples, a frame shift of 756 and 200 mel coefficients. The resulting spectrograms have a size of 200x266 pixels, figure 2 shows an example of one of our mel-spectrograms. We could have chosen to reduce much more the size of the spectrograms, there is a lot of redundancy using the mentioned resolution. Even though, since we have not had memory limitations and the neural networks we have trained are optimised for bigger resolutions, we have decided that this size would be appropriate for this work.

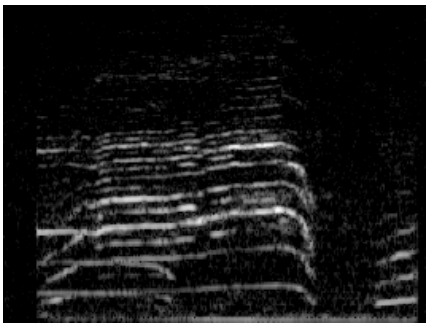

Fig. 2. Example of mel-spectrogram of a sound of an angry cat

In order to process all the dataset in an efficient way (this includes data augmentation and preprocessing) we have used a combination of multithreading bash and python scripts. We have also used LibROSA for computing the mel-spectrograms.

2.4. Deep Learning architectures

Our approach for classifying the spectrograms is using state-of-the-art convolutional neural nets. We have tested the CNN architectures for classification that have the best results in the literature. The tested neural networks are the following:

- VGG16 [4]
- VGG19 [4]
- ResNet50 [5]
- Xception [6]

All the networks have been trained with the same dataset, same augmentation and same parameters. About the optimiser, we used the *Adam* optimiser because we experienced that is the fastest to converge in our case. We have trained all the networks during 20 epochs, saving the weights of the epoch with the less validation error.

3. EXPERIMENTS AND DISCUSSION

As we have mentioned before, we trained all the architectures with the same dataset and the same parameters. All the weights of all the networks were initialised using transfer learning with the weights of the respective network pretrained with the imagenet dataset. After training all the networks we obtained the results shown in table 1.

	Test accuracy
VGG16	84%
VGG19	84'6%
ResNet50	**86%**
Xception	75'8%

Table 1. Accuracy results of our experiments

	Precision	Recall	F-Score	Support
Angry	0.83	0.81	0.82	83
Defence	**0.99**	**0.98**	**0.98**	84
Fighting	0.91	0.90	0.90	78
Happy	0.77	0.79	0.78	87
Hunting Mind	0.95	0.94	0.94	81
Mating	0.90	0.87	0.88	84
Mother Call	0.88	0.89	0.88	87
Paining	0.66	0.71	0.69	84
Resting	0.90	0.98	0.93	81
Warning	0.87	0.76	0.81	85
Average	0.86	0.86	0.86	834

Table 2. Classification report for ResNet50

We can observe that the ResNet50 achieves the bests results for our problem and the VGG19 and the VGG16 are also really close in terms of accuracy. The VGG16 is easier to train and the convergence time is shorter than the ResNet50, so depending on the task we have to consider if it is really necessary the extra computational cost for only an extra 1% or 2% of accuracy. The VGG16 and the VGG19 are quite similar, the only difference between them is that the VGG19 has more layers than the VGG16. We saw that adding these 3 layers not only does not contribute in achieving better performance but also slightly increases the computational cost. In the case of Xception, it is the architecture that took more training time and also is the one which has the worst results, so we conclude that the this architecture is not really good for our problem.

Table 2 shows the classification report for the ResNet50 network, which includes precision, recall and F-score measures for each of the ten classes. The last column shows the number of files in the test set for each class. At the last rows, the averages values for every metric are computed. Figure 3 shows the ROC curve averaged from all the classes, also for the ResNet50 network.

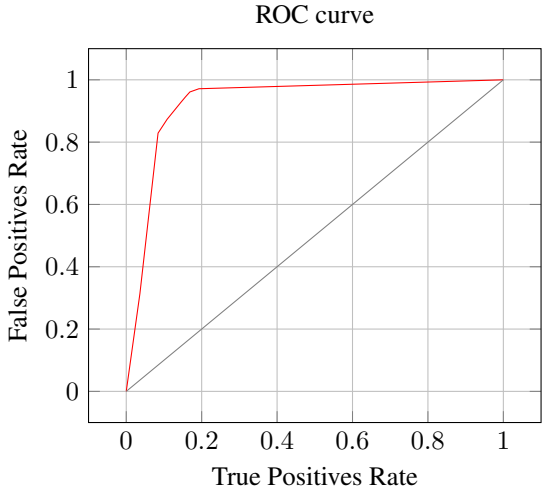

Fig. 3. Average ROC curve for all classes

Table 3 shows the confusion matrix for the network with best results, the ResNet50. This matrix lets us know which classes have more similarities and can be easily induced to produce errors. Looking at table 3 we can see that the sounds that incite more confusion are the ones from the classes Happy (meow-meow) and Paining (miyoou) which, despite having drastically different meanings, share similar characteristics. On the other side, we can observe that the sounds from the classes Resting (Purring) and Defence (Hissing) are unique to other cat sound.

4. CONCLUSIONS

Our idea was to work on a system that classifies the sounds produced by cats in an autonomous way. We have proven that our approach based on the computation of mel-spectrograms and classification using pre-trained CNNs has satisfying results. The result of 86% test accuracy using the ResNet50 neural network demonstrates that domestic cats sound classification using deep learning is a suitable task. We expect that if we were using a larger dataset with no need of performing data augmentation the results would be much better. For future works, this research could be extended increasing substantially the size of the dataset. Another interesting task involving the creation of the dataset would be to implement a cat sound detection algorithm in order to automatically create cat sound audio samples from larger files like microphone recordings.

	Angry	Defence	Fighting	Happy	Hunting Mind	Mating	Mother Call	Paining	Resting	Warning
Angry	**67**	0	2	1	1	3	2	4	0	3
Defence	0	**82**	0	0	1	0	0	0	0	1
Fighting	1	0	**70**	1	0	1	0	4	0	1
Happy	2	0	1	**69**	0	0	3	12	0	0
Hunting Mind	0	0	0	0	**76**	1	1	2	1	0
Mating	0	0	0	0	1	**73**	1	2	3	4
Mother Call	0	0	0	3	0	0	**77**	5	2	0
Paining	4	0	2	14	0	0	4	**60**	0	0
Resting	0	0	0	0	1	0	0	0	**79**	1
Warning	7	1	2	2	0	3	0	2	3	**65**

Table 3. Confusion Matrix for ResNet50

5. REFERENCES

[1] Nicholas Nicastro and Michael Owren, "Classification of domestic cat (felis catus) vocalizations by naive and experienced human listeners," *Journal of comparative psychology*, vol. 117, pp. 44–52, 04 2003.

[2] Yagya Raj Pandeya and Joonwhoan Lee, "Domestic cat sound classification using transfer learning," *INTERNATIONAL JOURNAL of FUZZY LOGIC and INTELLIGENT SYSTEMS*, vol. 18, pp. 154–160, 06 2018.

[3] Yagya Raj Pandeya, Dongwhoon Kim, and Joonwhoan Lee, "Domestic cat sound classification using learned features from deep neural nets," *Applied Sciences*, vol. 8, pp. 1949, 10 2018.

[4] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[6] François Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016.

A. DYNAMIC RANGE COMPRESSION FOR DATA AUGMENTATION

In order to improve the data augmentation process, we have included dynamic range compression on it. For it, we have used the *SoX* toolkit, specifically the *compand* function. This function has the following arguments:

```
compand attack1,decay1 knee:value1,value2,
        value3 gain initial-volume delay
```

We have set up an attack of 0.15s and a decay of 0.4s for all the cases. These values are chosen to be quite large because we do not want in any case to distort the signals. The soft-knee is one of the variable parameters, we have set it up as a normal random variable. The three following values define the transfer function of the compressor, all three have been also set up as normal random variables. Figure 4 shows the transfer function for the values of: -70,-60,-20. The last three

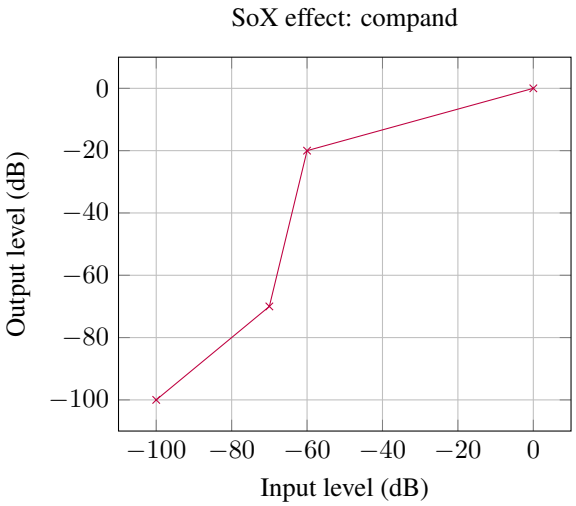

Fig. 4. Example of *SoX: compand* transfer function with the values of -70,-60,-20

parameters are fixed, we have chosen a gain of 0dB, an initial volume of -90dB and a delay of 0.1s.

The values chosen for the random variables are the following, as stated before all of them are normal distributions. All of these values are chosen ad hoc.

- knee: $\mu = 4dB, \sigma = 2$
- value1: $\mu = -65dB, \sigma = 9$
- value2: $\mu = -30dB, \sigma = 8$
- value3: $\mu = -22dB, \sigma = 6$