The Journal of Open Source Software

Pydiogment: A Python package for audio augmentation

Ayoub Malek[1] **and Hasna Marwa Malek**[2]

1 Yoummday GmbH **2** Grenoble Institute of Technology (Grenoble INP)

DOI: 00.00000/joss.00000

Software

- Review
- Repository
- Archive

Submitted: 25 February 2020
Published: 25 February 2020

Summary

This paper describes version 0.1.0 of `Pydiogment`: a Python package for audio augmentation based on the scipy (Jones, E., Oliphant, T., Peterson, P. & others, 2001) and ffmpeg (FFmpeg Developers, 2019) libraries. `Pydiogment` implements various augmentation techniques that can be used to improve the accuracy of various recognition tasks (speaker recognition, spoken emotions recognition, speech recognition etc.) and avoid over-fitting when training models. The paper provides a brief overview of the library's functionality, along with a small emotions recognition experiment displaying the utility of the library.

Implementation and theory

Audio data augmentation is a key step in training ML models to solve audio classification tasks. It is applied to increase the quality and size of the labeled training data set, in order to improve the recognition accuracy. Data augmentation is simply a deformation technique, that helps stretch the data, and increase its size for a better training. `Pydiogment` includes 3 general categories of deformations / augmentations:

- **Amplitude based augmentations** (`auga.py`) **:**
  - **Apply Gain** : applies a given gain (in dB) to the input signal.
  - **Add Fade** : adds a fade-in and fade-out effects to the original signal.
  - **Normalize** : normalizes the signal using the peak normalization method.
  - **Add Noise** : adds some random noise to the input signal based on a given signal to noise ratio (SNR).
- **Frequency based augmentation** (`augf.py`) **:**
  - **Change tone** : changes the pitch of the audio (lowered or raised).
  - **Apply Filter** : ...
- **Time based augmentation** (`augt.py`) **:**
  - **Time Stretching** : slows down speeds up the original audio based on a given coefficient.
  - **Time Shifting** : includes shifting the signal in a certain time direction or reversing the whole signal.
  - **Random Cropping** : generates a randomly cropped audio based on the original signal.
  - **Eliminate Silenc** : filters out silent frames from the input signal.
  - **Resample** : resamples the the input signal given an input sampling rate.

It is very important to maintain the semantic validity when augmenting the data. *For example:* one cannot change tones when doing voice based gender classification and still

expect tone to be a separating features of the predicted classes.

Experiment & Results

To prove the utility of `Pydiogment`, we display its effect on a spoken emotions recognition task. We use the Emo-DB data set (Burkhardt, F., Paeschke, A., Rolfes, M., A., Walter F. Sendlmeier & Weiss, B., 2005) as a starting point, which is a small German audio data set simulating 7 different emotions (neutral, sadness, anger, boredom, fear, happiness, disgust). In a first phase, we apply various recognition algorithms on the original data such as K-Nearest Neighbors (KNN), random forests, decision trees, Support Vector Machines (SVM) etc. In a second phase, we augment the data using `Pydiogment` by applying the following techniques:

- slow down samples using a coefficient of 0.8.
- speed up samples coefficient of 1.2.
- randomly crop samples with a minimum length of 1 second.
- add noise with an SNR = 10
- add a fade in and fade out effect.
- apply gain of -100 dB.
- apply gain of -50 dB.
- convolve with noise file using a level = 10**-2.75.
- shift time with one second (1 sec) to the right (direction = right)
- shift time with one second (1 sec) to the left (direction = left)
- change tone with tone coefficient equal to 0.9.
- change tone with tone coefficient equal to 1.1.

Then we re-run the same algorithms on the augmented and original data. The following is a comparison of the results:

Machine learning Algorithm	Accuracy (no augmentation)	Accuracy (with augmentation)
KNN	0.588	0.622
Decision Tree	0.474	0.568
AdaBoost	0.258	0.429
Random Forest	0.639	0.753
Linear SVM	0.113	0.286
Extra Trees Classifier	0.680	0.768

Conclusion

This paper introduced `Pydiogment`, a Python package for audio data augmentation, with diverse audio deformation strategies. These strategies aims to improve the accuracy of audio based recognition system by scaling the training data set and increasing its quality/diversity. The utility of `Pydiogment` was proved by showing its effects when used in a spoken emotions recognition task. In the stated experiment, the augmentation using `Pydiogment` improved the accuracy up to 50%.

Acknowledgements

This work was supported by Yoummday GmbH.

References

Burkhardt, F., Paeschke, A., Rolfes, M., A., Walter F. Sendlmeier & Weiss, B. (2005). A database of German emotional speech. *INTERSPEECH*.

FFmpeg Developers. (2019). FFmpeg tool (version 4.2.2). Retrieved from http://ffmpeg.org/

Jones, E., Oliphant, T., Peterson, P. & others. (2001). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/