The Journal of Open Source Software

Pydiogment: A Python package for audio augmentation

Ayoub Malek[1]

1 Yoummday GmbH

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, that can be used to improve various recognition tasks (speaker recognition, spoken emotions recognition, etc.). The paper provides a brief overview of the library's functionality, along with a small emotions recognition experiment displaying the utility of the library.

Data Augmentation

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:

- `auga` **Amplitude based augmentations:**
  - *Apply Gain:* This will apply a given gain (in dB) to the input signal.
  - *Add Fade:* This adds a fade-in and fade-out effects to the original signal.
  - *Add Noise:* This adds some random noise to the input signal based on a given signal to noise ratio (SNR).
- `augf` **Frequency based augmentation:**
  - *Change tone:* The pitch of the audio is changed (lowered or raised).
  - *Apply Filter:*
- `augt` **Time based augmentation:**
  - *Time Stretching:* This slows down speeds up the original audio based on a given coefficient.
  - *Time Shifting:* This includes shifting the signal in a certain time direction or reversing the whole signal.
  - *Random Cropping:* This generates a randomly cropped audio based on the original signal.
  - *Eliminate Silence:* This deformation can will filter out silent frames from the input signal.

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 use it in a spoken emotions recognition task. We use the Emo-DB data set (Felix Burkhardt and Astrid Paeschke and Melissa A Rolfes and Walter F. Sendlmeier and Benjamin Weiss, 2005) as a starting point, which is a small German audio data set simulating 7 different emotions (neutral, sadness, anger, boredom, fear, happiness, disgust). We apply various recognition algorithms on the original data such as K-Nearest Neighbors (KNN), random forests, decision trees, Support Vector Machines (SVM) etc. then we augment the data using `pydiogment` and re-run the same algorithms. The following is a comparison of the results:

Machine learning Algorithm	Accuracy (no augmentation)	Accuracy (with augmentation)	Accuracy improvement
KNN	0.588	0.622	0.05
Decision Tree	0.474	0.568	0.09
AdaBoost	0.258	0.429	0.17
Random Forest	0.639	0.753	0.12
Linear SVM	0.113	0.286	0.17
Extra Trees Classifier	0.68	0.768	0.08

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%.

Citations

Citations to entries in paper.bib should be in rMarkdown format.

Acknowledgements

This work was supported by Yoummday GmbH.s

References

Felix Burkhardt and Astrid Paeschke and Melissa A Rolfes and Walter F. Sendlmeier and Benjamin Weiss. (2005). A database of German emotional speech. *INTERSPEECH*.