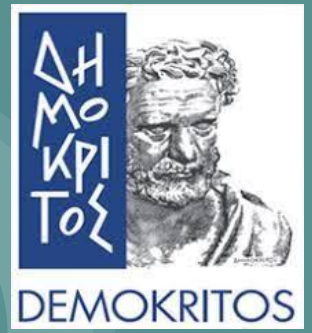


Multimodal Project Presentation

Audio fingerprinting

Authors: Gkatsis Vasilis Varsou Penny

Introduction

- Audio fingerprinting is something that nowadays became more useful and useful in many different fields.

- All the algorithms like shazam and etc. are close and cannot be used in order to develop more apps.

Github link: https://github.com/VarsouPenny/dejavu

Project consists the following steps

- Download 5800 songs from youtube using Youtube-DLG app.
- Convert all songs to monocanal using ffmpeg.
- We run Dejavu fingerprinting and create three distinct databases with 5800, 1000, and 100 songs.
- We run Dejavu recognize:
  - For different time queries (songs in disk).
  - For different time queries with various noise level (songs in disk).
  - For different time queries through microphone.

Dejavu library

- An audio fingerprinting and recognition algorithm implemented in Python.
- Can memorize an audio by listening to it and fingerprinting it.
- Can recognize an recognize a songs by recording microphone input or reading from disk.
- The above happens because it attempts to match the audio against the fingerprints held in database.

Dejavu library

- An audio fingerprinting and recognition algorithm implemented in Python.
- Can memorize an audio by listening to it and fingerprinting it.
- Can recognize an recognize a songs by recording microphone input or reading from disk.
- The above happens because it attempts to match the audio against the fingerprints held in database.

Dejavu library

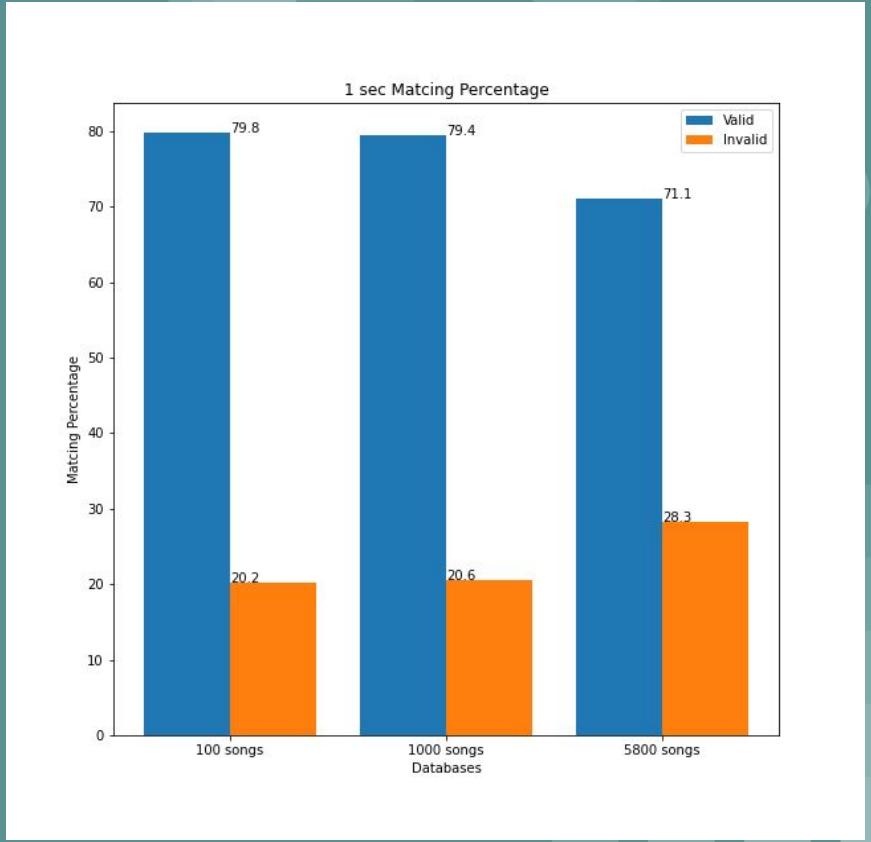
- An audio fingerprinting and recognition algorithm implemented in Python.
- Can memorize an audio by listening to it and fingerprinting it.
- Can recognize an recognize a songs by recording microphone input or reading from disk.
- The above happens because it attempts to match the audio against the fingerprints held in database.

Songs

- We try to include a variety of different kinds of songs in many languages.
- We download them as mp3.
- We convert them to monocanals.

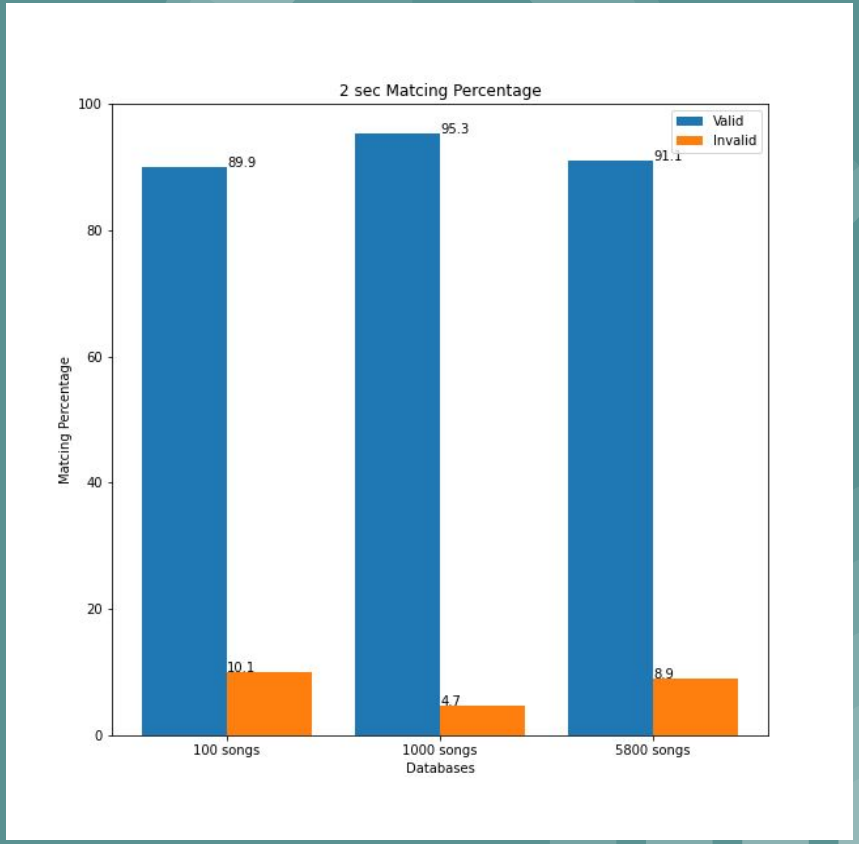
Integration test

- We test evaluation and performance with different time queries in all databases
  - For 1 sec query:



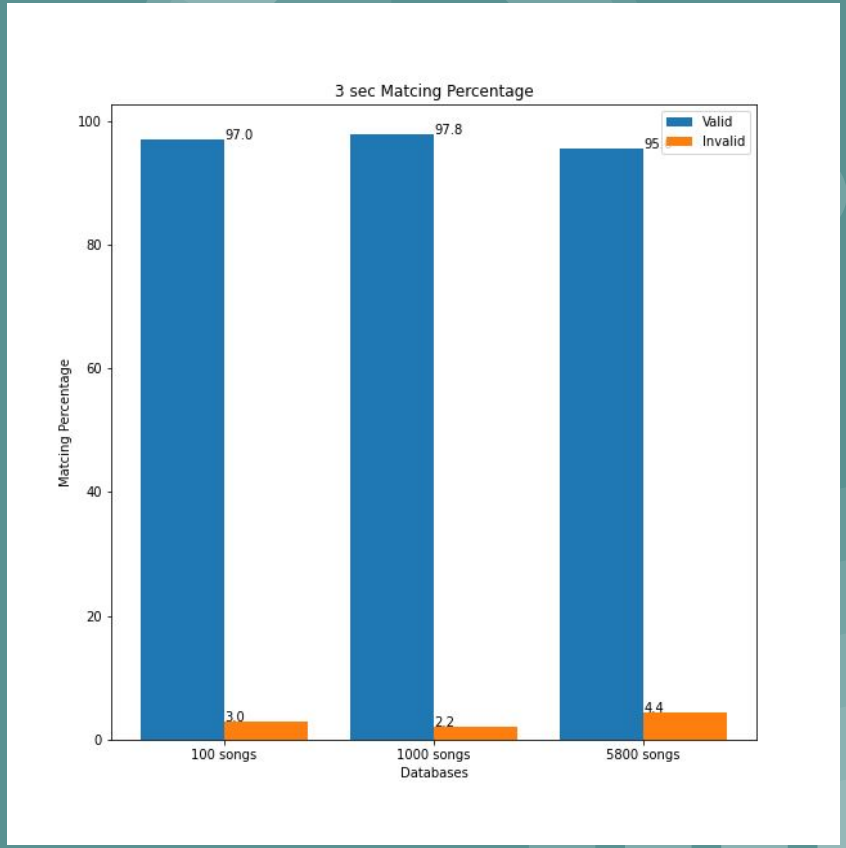
Integration test

- We test evaluation and performance with different time queries in all databases
  - For 2 sec query:



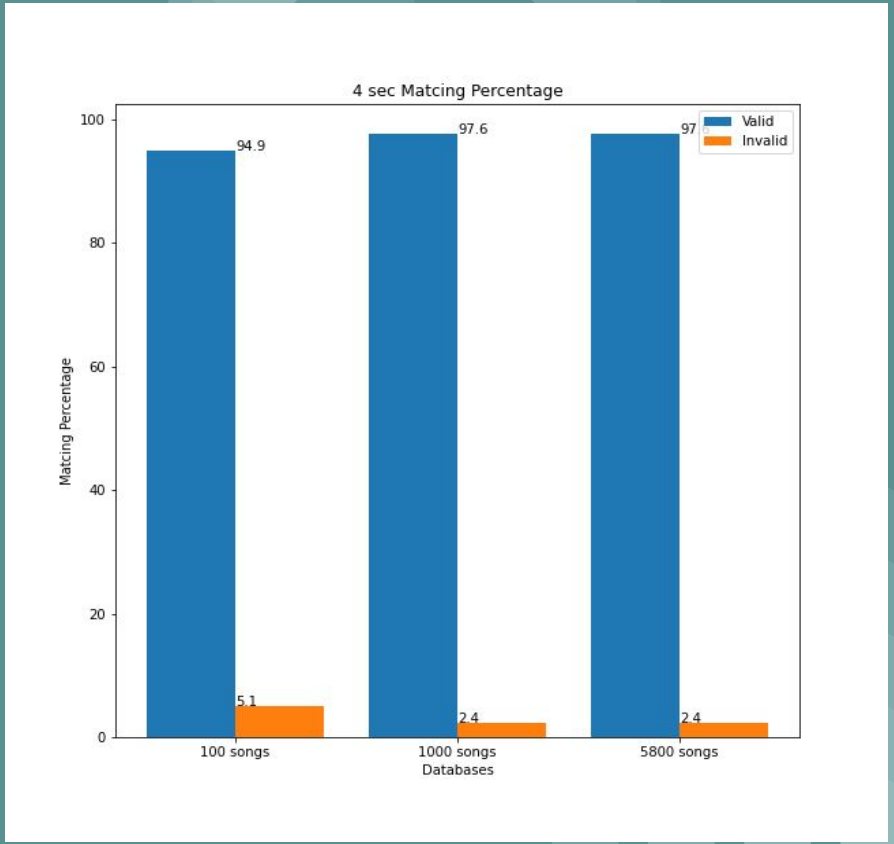
Integration test

- We test evaluation and performance with different time queries in all databases
  - For 3 sec query:



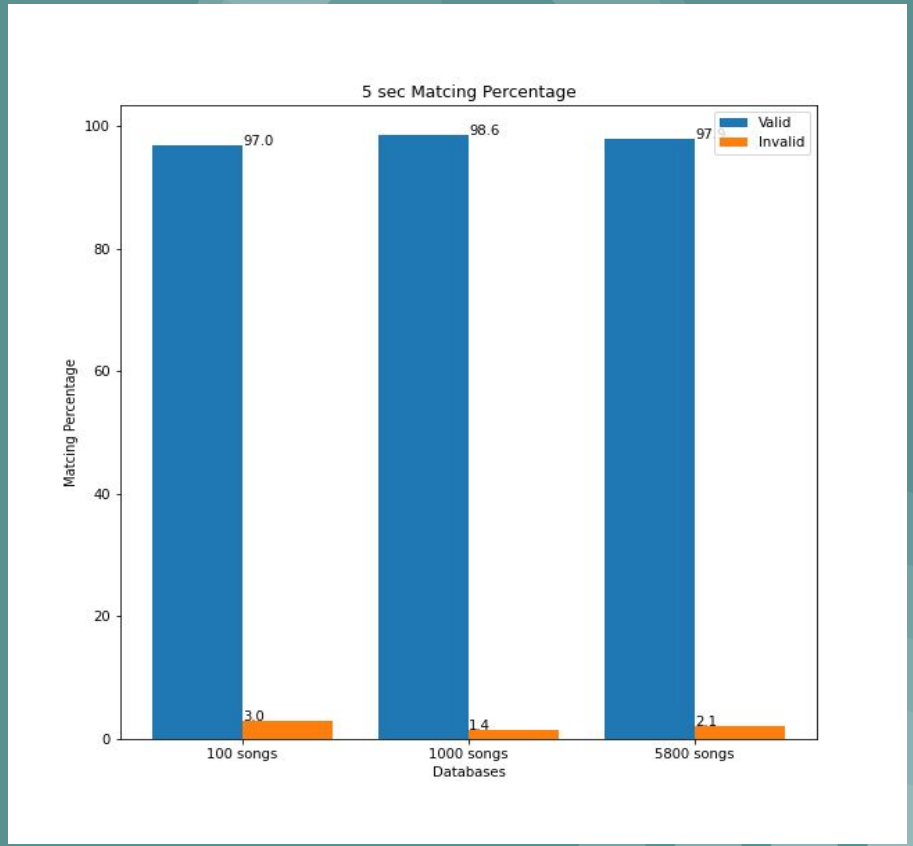
Integration test

- We test evaluation and performance with different time queries in all databases
  - For 4 sec query:



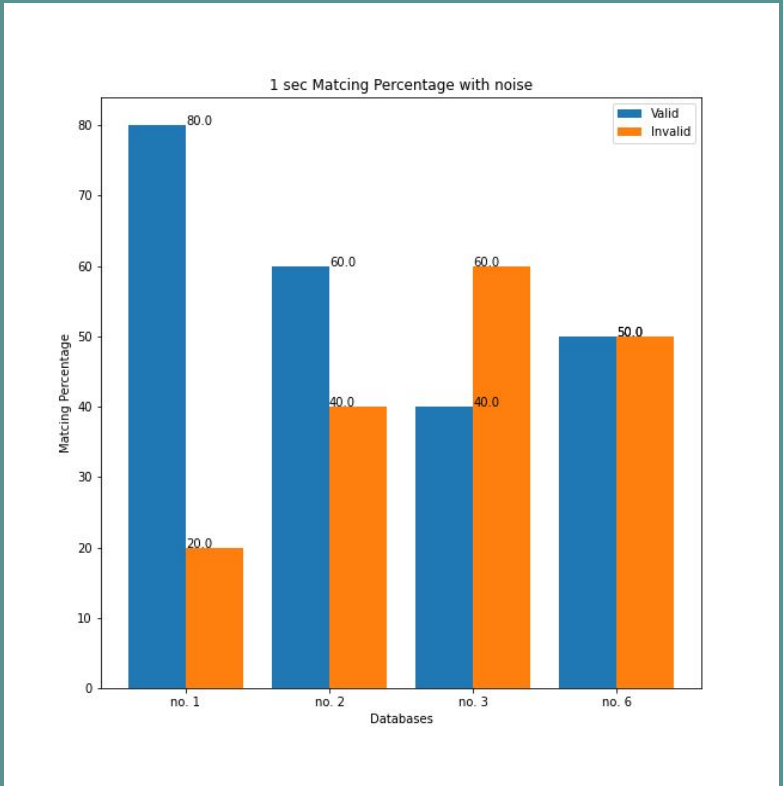
Integration test

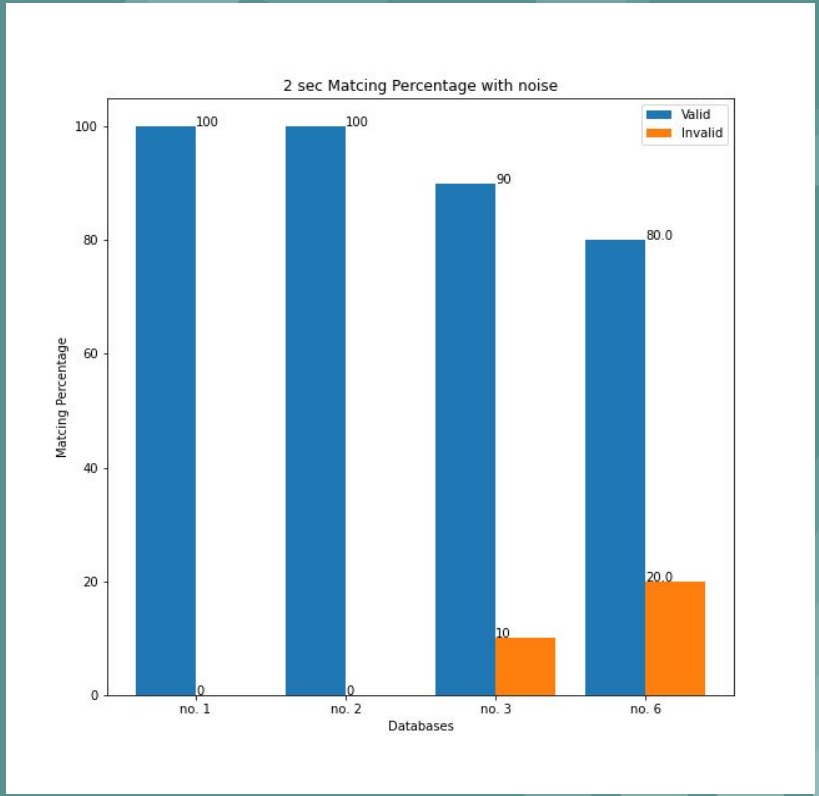
- We test evaluation and performance with different time queries in all databases
  - For 5 sec query:



Integration test

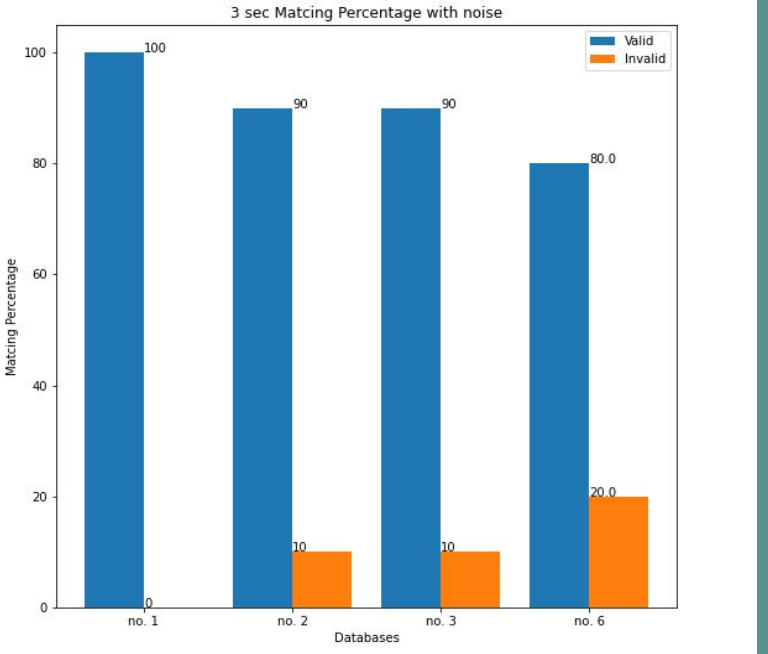
- Test evaluation and performance with different time queries and 4 distincts noise levels

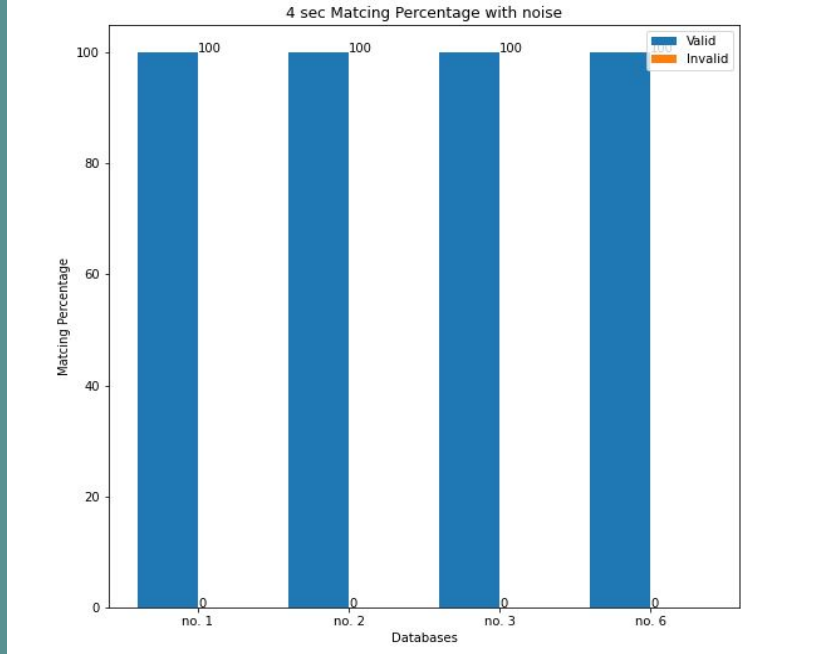




Integration test

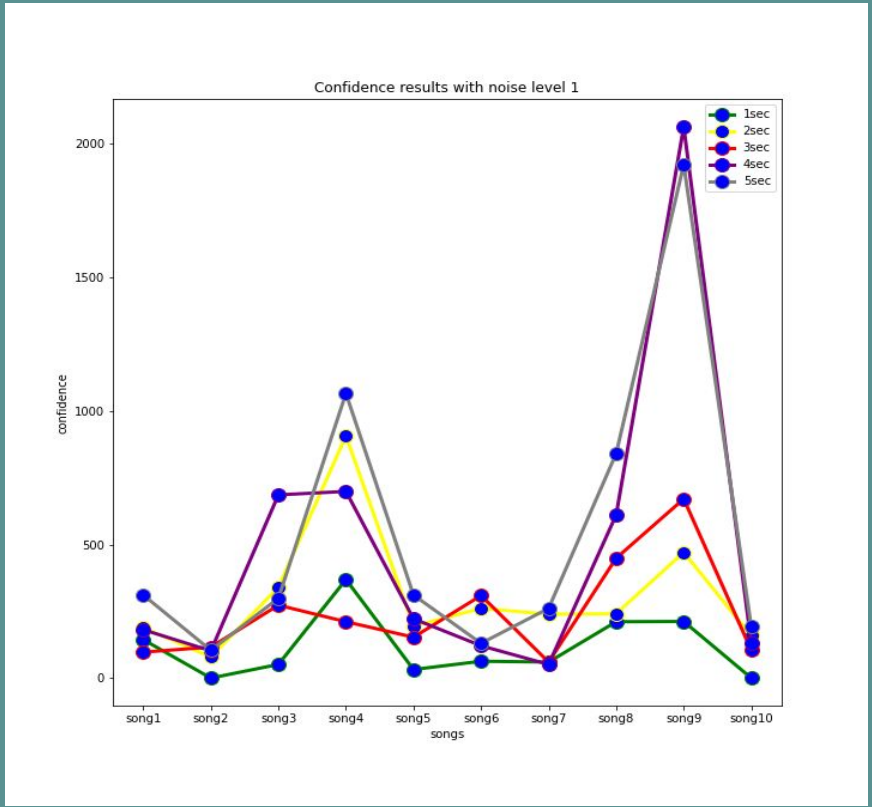
- Test evaluation and performance with diffent time queries and distinct noise level

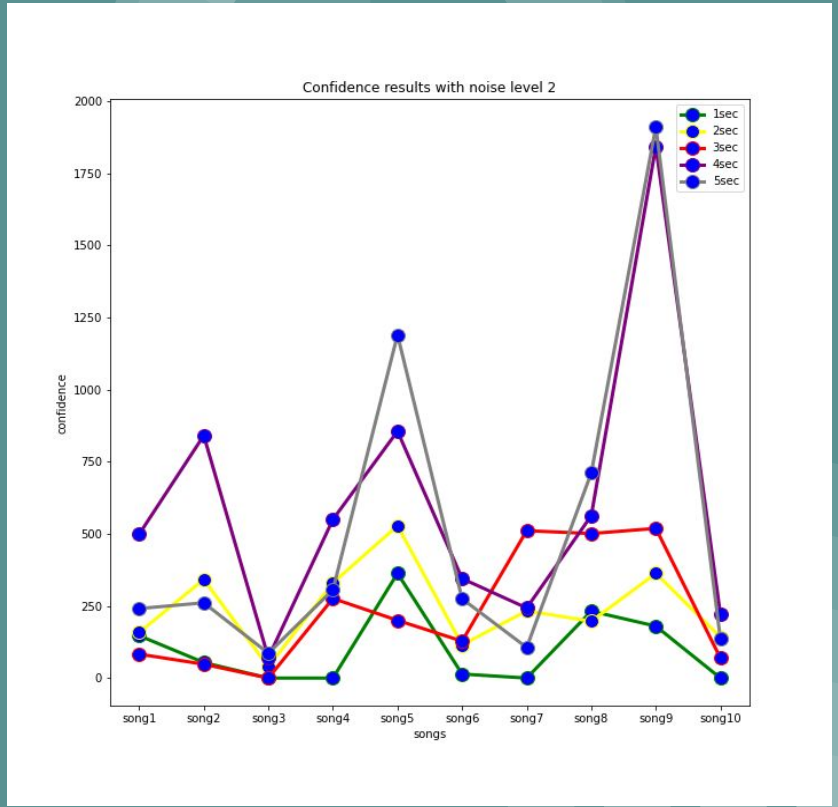




Integration test

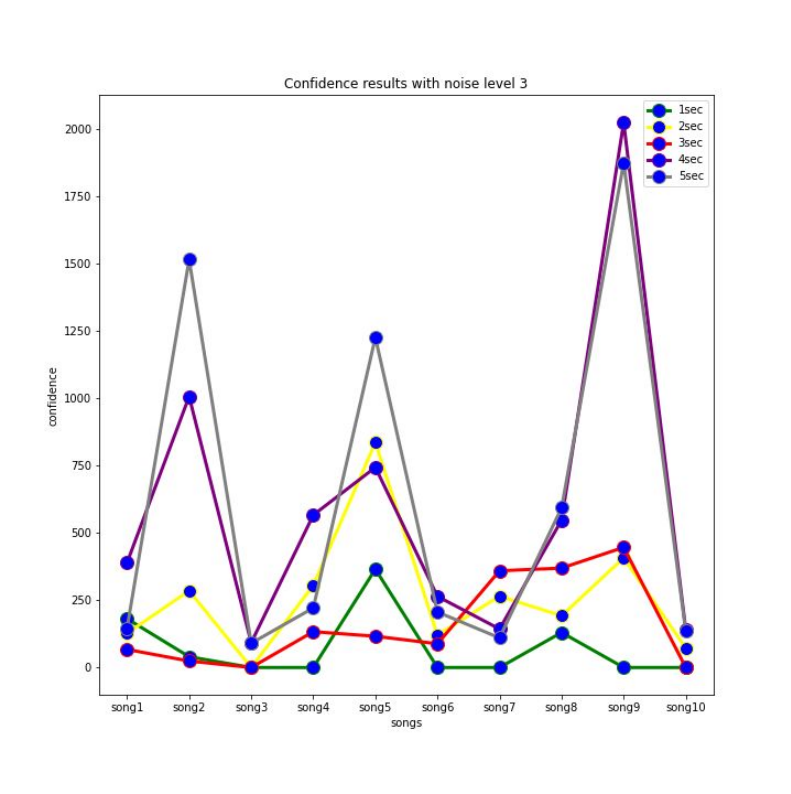
- We also make plots with the confidence that Dejavu replies each time

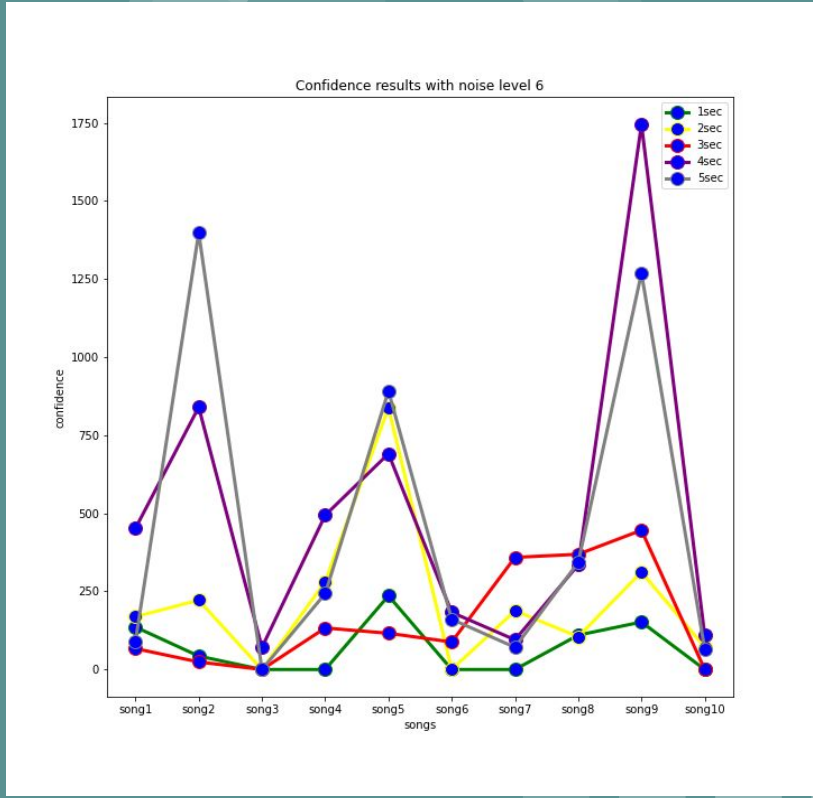




Integration test

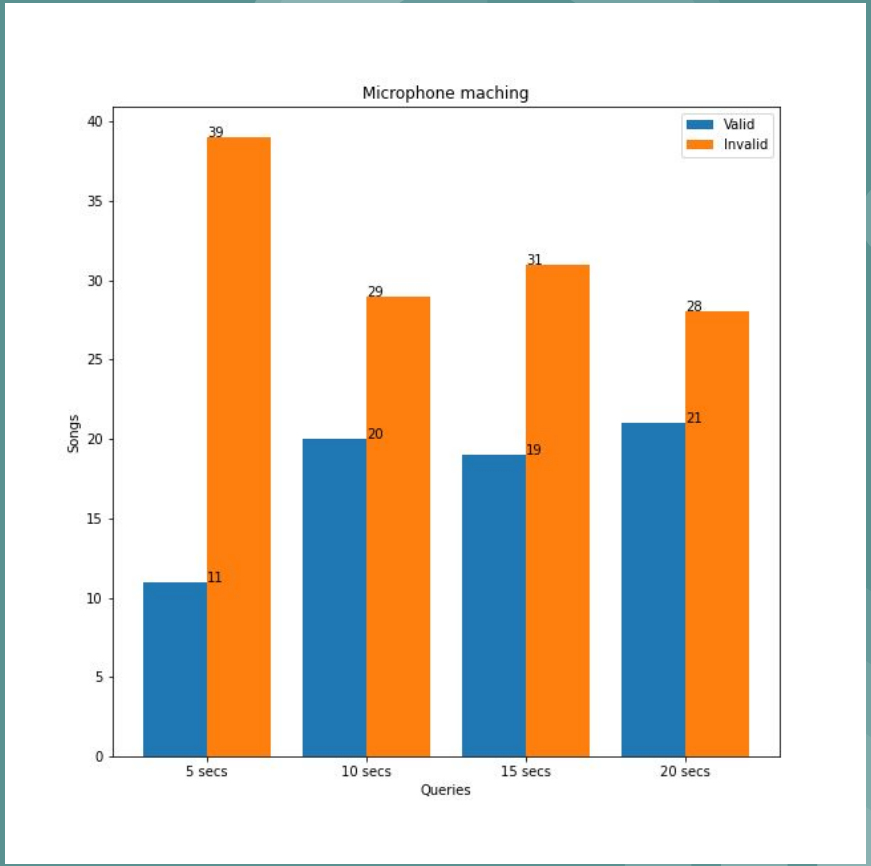
- We also make plots with the confidence that Dejavu replies each time





Integration test

- Last but not least we test song recognition by recording microphone input .



Some conclutions

- Dejavu is more sensitive to small time queries than in noise level.
- For microphone recognition it depends on the amount of songs that database has and the part of song that is given as input to identify.

Future work

- Try with more that 10k songs.
- Calculate query duration in different databases and compare.
- Calculate and compare confidence in different databases.

References

1. https://willdrevo.com/fingerprinting-and-audio-recognition-with-python/
2. A. Wang, “An industrial strength audio search algorithm.,” Jan 2003.
3. https://pypi.org/project/PyDejavu/
4. https://www.jesusninoc.com/02/06/dejavu-audio-fingerprinting-and-recognition-in-python/