NATIONAL INSTITUTE OF TECHNOLOGY DELHI

Summer Internship Project Report

On

MUSIC GENRE CLASSIFICATION

By

PRITAM SARKAR (171210044)

KRISHAN KUMAR (171210035)

Under the supervision of

Dr. Anurag Singh

Assistant Professor

Department of Computer Science and Engineering

DECLARATION

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included , we have adequately cited and referenced the original sources, we also declare that we have adhered to all principles of academics honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not properly cited or from whom proper permission has not been taken when needed.

Date:

Dr. Anurag Singh

Department of Computer
Science and Engineering

ACKNOWLEDGEMENT

An undertaking of work life - this is never an outcome of a single person; rather it bears the imprints of a number of people who directly or indirectly helped us in completing the present study. We would be failing in our duties if we don't say a word of thanks to all those who made our training period educative and pleasurable one.

First of all, we are extremely grateful to **Dr. Anurag Singh**, Department of Computer Science and Engineering for his continuous support, mellow criticism and able directional guidance during the project. Our special thanks to him for giving us the opportunity to do this project and for his support throughout as a mentor.

We would also thanks to **Dr. Shirin Dora** for his guidance, encouragement and tutelage during the course of the internship despite his extremely busy schedule.

We would also like to thank **Mr. Abhishek Saroha** for giving their precious time and relevant information and experience, we required, without which the Project would have been incomplete.

Finally, We would like to thank all lecturers and friends for their kind support and to all who have directly or indirectly helped us in this project. And at last we thankful to all divine light and our parents, who kept our motivation and zest for knowledge always high through the tides of time.

TABLE OF CONTENTS

ABSTRACT

The lack of data tends to limit the outcomes of deep learning research. In this study, 25k tracks annotated with musical labels are available to train mel-spectrogram models. This large amount of data allows us to unrestrictedly explore two different design paradigms for music auto-tagging: assumption-free models – using waveforms as input with very small convolutional filters; and models that rely on domain knowledge – log-mel spectrograms with a convolutional neural network designed to learn timbral and temporal features. Our work focuses on studying how these two types of deep architectures perform when datasets of variable size are available for training: the MagnaTagATune (25k songs).

PROBLEM STATEMENT

To extract the feature of audio songs and use it train the model of deep neural network and provide the accurate genres that are relevant to specific song.

INTRODUCTION

Music-streaming services have made a huge volume of music accessible to users, the enormous size of the service catalogs has created the challenge of finding among so many choices the songs that fit users' tastes. A general approach to this issue has been collaborative filtering, which predicts songs of potential interest based on previous usage data, such as play history and song rating. Although collaborative filtering effectively retrieves songs and accommodates personalized recommendations, its performance is hampered by such issues as popularity bias and the cold-start problem, the challenge of recommending new music to users. The content-based approach is often regarded as a supplementary solution to those problems. Pandora radio is a representative example as it retrieves songs by exploiting the similarities of song descriptors, such as genre, mood, instruments, and vocal quality. However, high-quality manual annotation is costly and not scalable, suggesting a need for better ways to automate classification of music content. As a result, much attention in the field of music information retrieval (MIR) over the last few years has centered on finding ways to automate the process of classifying music genre and mood and tagging music. Hereafter, this article will use the term music classification and tagging as a general expression for tasks that involve taking music audio data as input and automatically annotating them with a certain form of semantic label.

Wikipedia states that "music genre is a conventional category that identifies pieces of music as belonging to a shared tradition or set of conventions." The term "genre" is a subject to interpretation and it is often the case that genres may very fuzzy in their definition. Further, genres do not always have sound music theoretic foundations, e.g. - Indian genres are geographically defined, Baroque is classical music genre based on time period. Despite the lack of a standard criteria for defining genres, the classification of music based on genres is one of the broadest and most widely used. Genre usually assumes high weight in music recommender systems. Genre classification, till now, had been done manually by appending it to metadata of audio files or including it in album info. This project however aims at content-based classification, focusing on information within the audio rather than extraneously appended information. The traditional machine learning approach for classification is used - find suitable features of data, train classifier on feature data, make predictions. The novel thing that we have tried is the use of ensemble classifier on fundamentally different classifiers to achieve our end goal.

PROPOSED METHODOLGY

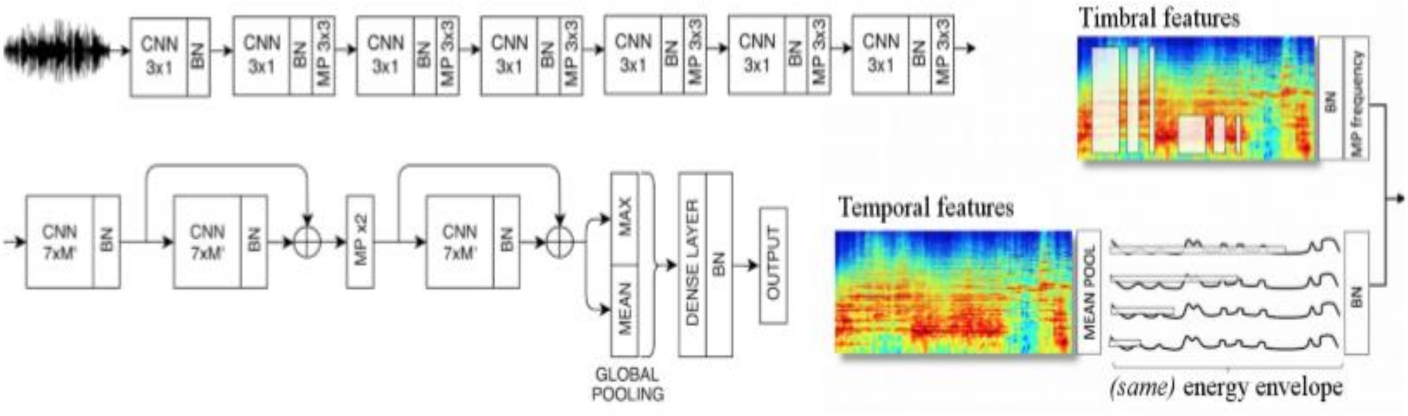


Figure 1: **Bottom-left** – back-end. **Top-left** – waveform front-end. **Right** – spectrogram front-end.

(Definitions– **M'** stands for the feature map's vertical axis, BN for batch norm, and MP for max-pool.)

Spectrogram front-end.

1. Audio segments are converted to log-mel magnitude spectrograms (15 seconds and 96 mel bins) and normalized to have zero-mean and unit-var.

2. We use vertical and horizontal filters explicitly designed to facilitate learning the timbral and temporal patterns present in spectrograms.

Note in **Figure 1(**Right) that the spectrogram front-end is a single-layer CNN with many filter shapes that are grouped into two branches .

(i) **top branch** – timbral features

(ii) **lower branch** – temporal features

The top branch is designed to capture pitch invariant timbral features that are occurring at different time-frequency scales in the spectrogram. Pitch invariance is enforced via enabling CNN filters to convolve through the frequency domain, and via max-pooling the feature map across its vertical axis .Note that several filter shapes are used to efficiently capture many different time-frequency patterns: 7×86,3×86,1×86,7×38,3×38 and 1×385 – to facilitate learning, e.g.: kick-drums (with small-rectangular filters of 7×38 capturing sub-band information for a short period of time), or string ensemble instruments (with long vertical filters of 1×86 which are capturing timbral patterns spread in the frequency axis).

The lower branch is meant to learn temporal features, and is designed to efficiently capture different time-scale representations by using several long filter shapes :165×1, 128×1,64×1 and 32×1.These filters operate over an energy envelope(not directly over the spectrogram) obtained via mean-pooling the frequency-axis of the spectrogram.

Shared back-end:

It consists of three CNN layers (with 512 filters each and two residual connections), two pooling layers and a dense layer – see **Figure 1 (Bottom-left).**

We introduced residual connections in our model to explore very deep architectures, such that we can take advantage of the large data available. Although adding more residual layers did not drastically improve our results, we observed that adding these residual connections stabilized learning while slightly improving performance. The used DCNN filters are computationally efficient and shaped such that all extracted features are considered across reasonable amount of temporal context (note the 7×M' filter shapes, representing time ×all features). We also make a drastic use of temporal pooling: firstly, down-sampling x2 the temporal dimensionality of the feature maps.

By making use of global pooling with mean and max statistics. The global pooling strategy allows for variable length inputs to the network and therefore, such a model can be classified as a "variable-length input" back-end. Finally, a dense layer with 500 units connects the pooled features to a sigmoidal output.

OPTIMIZER:

50% dropout before every dense layer, ReLUs as non-linearities, and our models are trained with SGD employing Adam (with an initial learning rate of 0.0001) as optimizer.

PROGRESS AND CURRENT RESULT

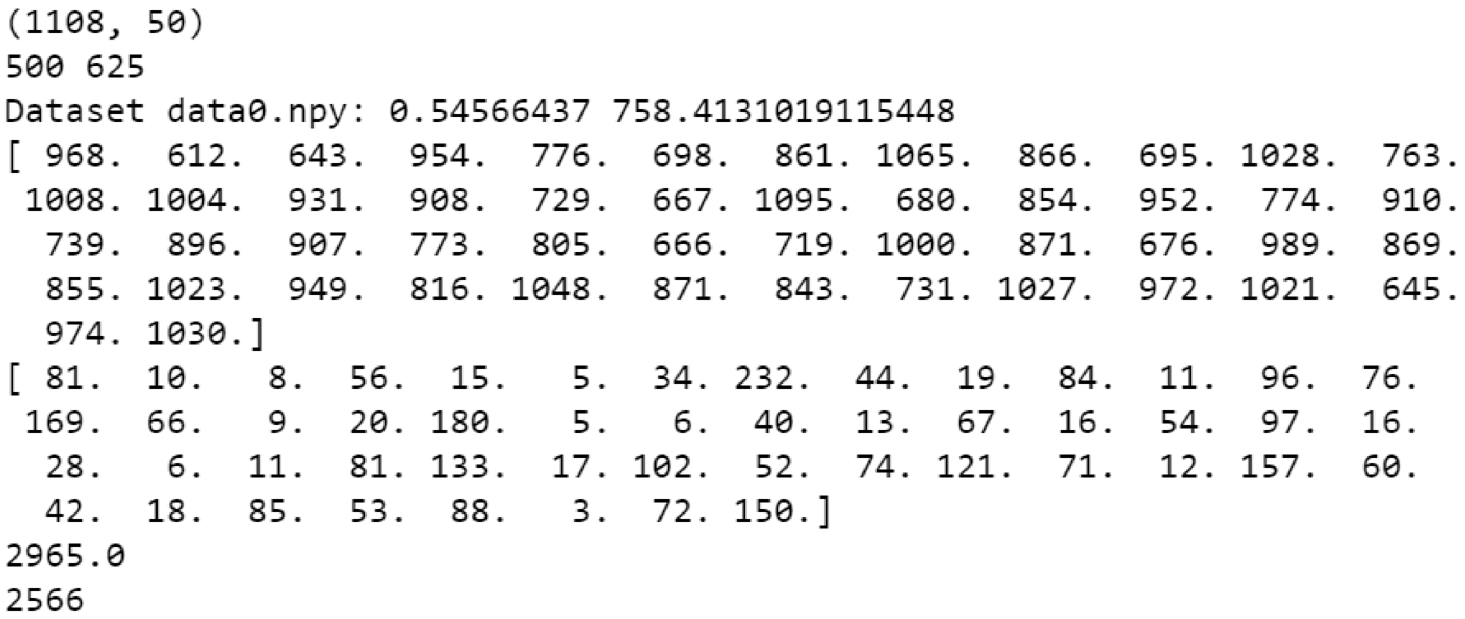
We have implemented music genre classification to classify songs using tensorflow. This implementation

1. uses mel-spectrogram front end with shared CNN back end.

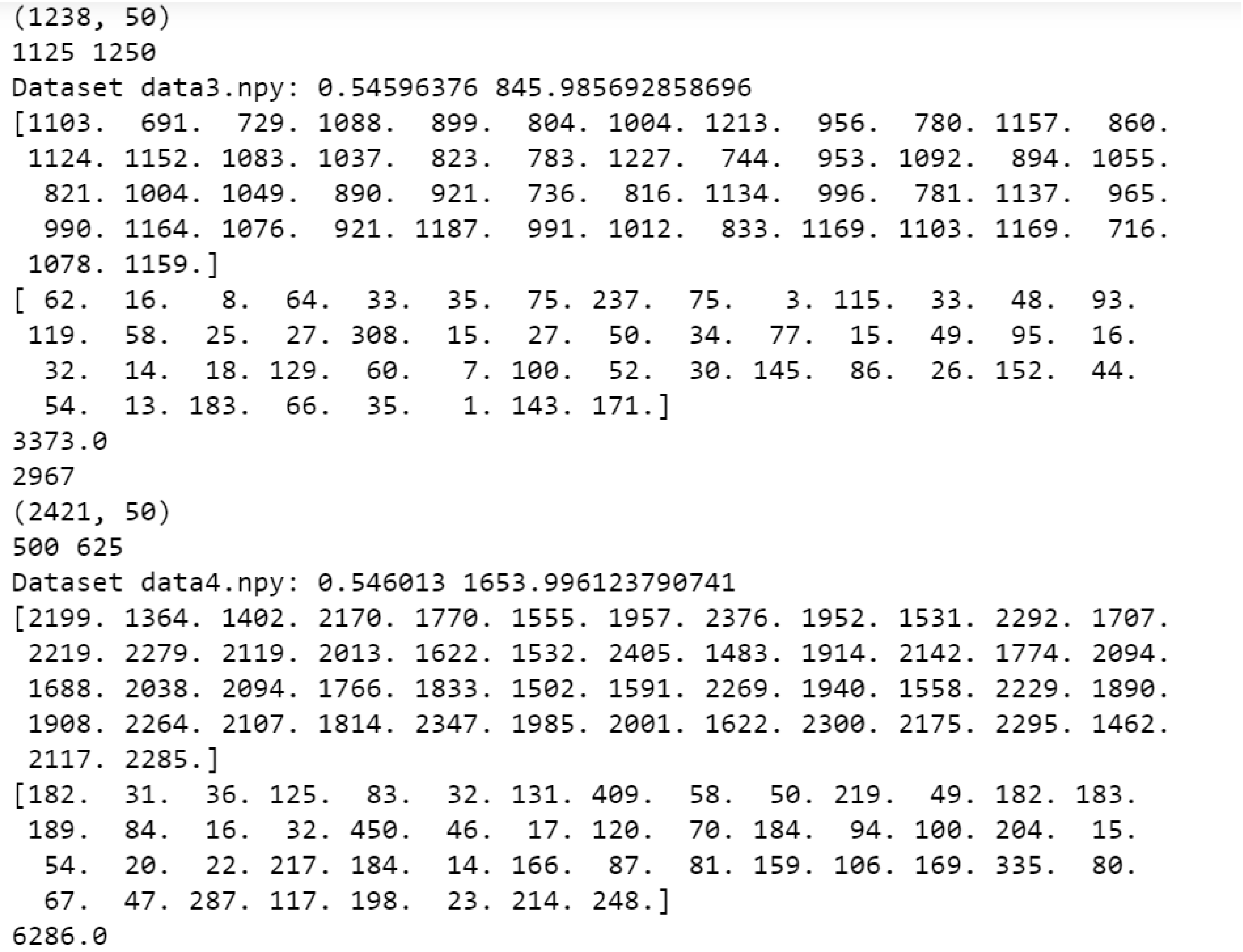
2. extracts 5 timbral feature and 4 temporal features from mel-spectrogram of audio songs.

3. train the model (i.e. shared backend) with help of 9 features that are extracted as discuss in point 2.

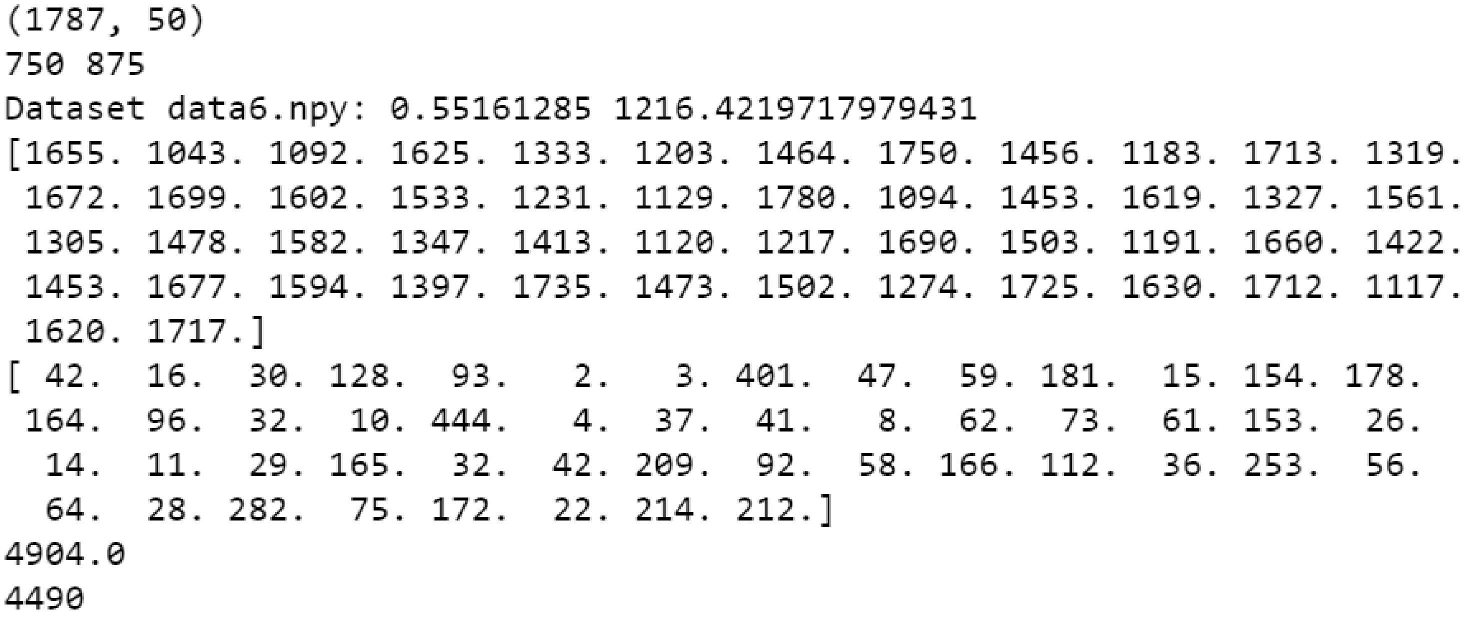
Generated Result:



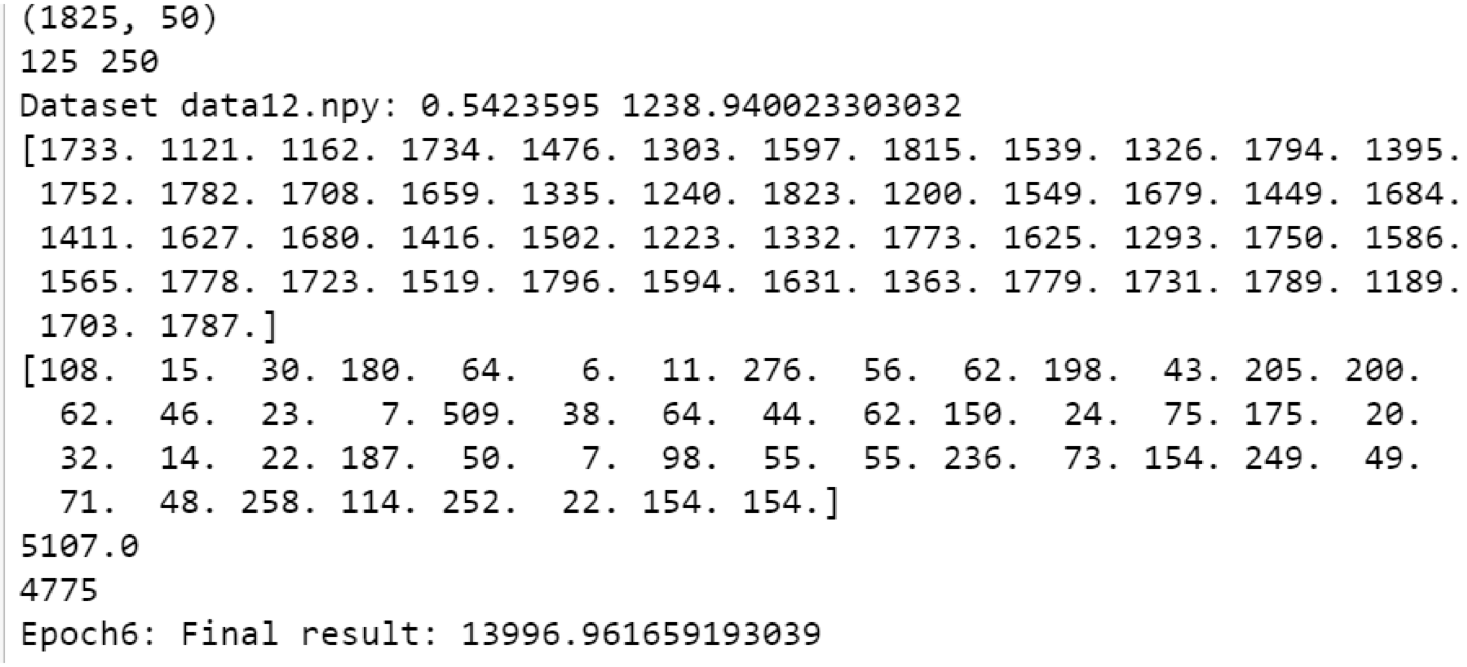
```
(1108, 50)
500 625
Dataset data0.npy: 0.54566437 758.4131019115448
[ 968.  612.  643.  954.  776.  698.  861. 1065.  866.  695. 1028.  763.
 1008. 1004.  931.  908.  729.  667. 1095.  680.  854.  952.  774.  910.
  739.  896.  907.  773.  805.  666.  719. 1000.  871.  676.  989.  869.
  855. 1023.  949.  816. 1048.  871.  843.  731. 1027.  972. 1021.  645.
  974. 1030.]
[ 81.  10.   8.  56.  15.   5.  34. 232.  44.  19.  84.  11.  96.  76.
 169.  66.   9.  20. 180.   5.   6.  40.  13.  67.  16.  54.  97.  16.
  28.   6.  11.  81. 133.  17. 102.  52.  74. 121.  71.  12. 157.  60.
  42.  18.  85.  53.  88.   3.  72. 150.]
2965.0
2566
```



```
(1238, 50)
1125 1250
Dataset data3.npy: 0.54596376 845.985692858696
[1103.  691.  729. 1088.  899.  804. 1004. 1213.  956.  780. 1157.  860.
 1124. 1152. 1083. 1037.  823.  783. 1227.  744.  953. 1092.  894. 1055.
  821. 1004. 1049.  890.  921.  736.  816. 1134.  996.  781. 1137.  965.
  990. 1164. 1076.  921. 1187.  991. 1012.  833. 1169. 1103. 1169.  716.
 1078. 1159.]
[ 62.  16.   8.  64.  33.  35.  75. 237.  75.   3. 115.  33.  48.  93.
 119.  58.  25.  27. 308.  15.  27.  50.  34.  77.  15.  49.  95.  16.
  32.  14.  18. 129.  60.   7. 100.  52.  30. 145.  86.  26. 152.  44.
  54.  13. 183.  66.  35.   1. 143. 171.]
3373.0
2967
(2421, 50)
500 625
Dataset data4.npy: 0.546013 1653.996123790741
[2199. 1364. 1402. 2170. 1770. 1555. 1957. 2376. 1952. 1531. 2292. 1707.
 2219. 2279. 2119. 2013. 1622. 1532. 2405. 1483. 1914. 2142. 1774. 2094.
 1688. 2038. 2094. 1766. 1833. 1502. 1591. 2269. 1940. 1558. 2229. 1890.
 1908. 2264. 2107. 1814. 2347. 1985. 2001. 1622. 2300. 2175. 2295. 1462.
 2117. 2285.]
[182.  31.  36. 125.  83.  32. 131. 409.  58.  50. 219.  49. 182. 183.
 189.  84.  16.  32. 450.  46.  17. 120.  70. 184.  94. 100. 204.  15.
  54.  20.  22. 217. 184.  14. 166.  87.  81. 159. 106. 169. 335.  80.
  67.  47. 287. 117. 198.  23. 214. 248.]
6286.0
```



```
(1787, 50)
750 875
Dataset data6.npy: 0.55161285 1216.4219717979431
[1655. 1043. 1092. 1625. 1333. 1203. 1464. 1750. 1456. 1183. 1713. 1319.
 1672. 1699. 1602. 1533. 1231. 1129. 1780. 1094. 1453. 1619. 1327. 1561.
 1305. 1478. 1582. 1347. 1413. 1120. 1217. 1690. 1503. 1191. 1660. 1422.
 1453. 1677. 1594. 1397. 1735. 1473. 1502. 1274. 1725. 1630. 1712. 1117.
 1620. 1717.]
[ 42.  16.  30. 128.  93.   2.   3. 401.  47.  59. 181.  15. 154. 178.
 164.  96.  32.  10. 444.   4.  37.  41.   8.  62.  73.  61. 153.  26.
  14.  11.  29. 165.  32.  42. 209.  92.  58. 166. 112.  36. 253.  56.
  64.  28. 282.  75. 172.  22. 214. 212.]
4904.0
4490
```



```
(1825, 50)
125 250
Dataset data12.npy: 0.5423595 1238.940023303032
[1733. 1121. 1162. 1734. 1476. 1303. 1597. 1815. 1539. 1326. 1794. 1395.
 1752. 1782. 1708. 1659. 1335. 1240. 1823. 1200. 1549. 1679. 1449. 1684.
 1411. 1627. 1680. 1416. 1502. 1223. 1332. 1773. 1625. 1293. 1750. 1586.
 1565. 1778. 1723. 1519. 1796. 1594. 1631. 1363. 1779. 1731. 1789. 1189.
 1703. 1787.]
[108.  15.  30. 180.  64.   6.  11. 276.  56.  62. 198.  43. 205. 200.
  62.  46.  23.   7. 509.  38.  64.  44.  62. 150.  24.  75. 175.  20.
  32.  14.  22. 187.  50.   7.  98.  55.  55. 236.  73. 154. 249.  49.
  71.  48. 258. 114. 252.  22. 154. 154.]
5107.0
4775
Epoch6: Final result: 13996.961659193039
```

BIBLIOGRAPHY

1.https://ismir2017.smcnus.org/lbds/Pons2017.pdf.

2. https://ieeexplore.ieee.org/document/8588424.

3. http://tagatune.org/