OM{1,2,3}: MIREX AUDIO KEY DETECTION

James Owers
University of Edinburgh
james.owers@ed.ac.uk

Andrew McLeod
University of Edinburgh
amcleod8@ed.ac.uk

ABSTRACT

In this paper, we describe three models submitted to the 2018 MIREX Audio Key Detection task. The goal of the task is to identify the key of an audio recording of a piece of western music. Our three models are: (**OM1**) a novel neural network consisting of convolutional layers followed by a GRU for temporal context; (**OM2**) a reproduction of [3], the best-performing model from last year's competition, which has a similar structure to OM1, but without the potential for picking up temporal information that the GRU has; and (**OM3**) a simple logistic regression baseline. While our models are outperformed in this year's task by others in the official evaluation, there are some test datasets where even (**OM3**) performs comparably if not better than other models. Our main contributions are an examination of the key distributions of various datasets, as well as an investigation into how the datasets chosen for training might affect test performance. All of the code used in this work, including for the dataset examination, is freely available at https://github.com/apmcleod/key-detect.

1. INTRODUCTION

The key of a piece of western music identifies the tonal centre of that piece, its most common pitches, as well as its mode. As such, the key detection is an important problem in music information retrieval. In the MIREX Audio Key Detection task, an audio excerpt is given and each model is asked to identify the key, which may be any of 24: all 12 possible tonal centres and either major or minor mode (other modes are not considered).

2. SUBMISSIONS

Last year's top-performing model [3] does not seem to incorporate any long-distance temporal information into its key detection. The convolutional layers have the ability to use up to five seconds of context, but no more. This is interesting, because listening for long-distance temporal clues such as cadences and chord progressions are what we (the authors) use when performing the task by hand.

Thus, our first submission, **OM1**, is structured similarly to [3], initially stacking 5 convolutional layers. However, it then consists of a bi-directional GRU layer to give temporal context, followed by another convolutional layer and two linear layers (full implementation details are in the code).

Our second submission, **OM2**, is simply our reproduction of [3], trained on our own data.

Finally, **OM3** is a simple baseline which performs logistic regression on the average magnitude in each frequency bin across time.

We preprocess the input into CQT spectrograms with bins ranging from C1 to C7 in quarter tone steps. [1]

3. DATA

We use four different datasets for training and testing:

1. GTZAN [6]: 837 pieces from various genres. [2]
2. Giant Steps [2]: 604 pieces of EDM.
3. Giant Steps MTG: 1347 additional pieces of EDM. [3]
4. Million Songs Dataset (MSD) [1]: contemporary popular music, each labelled with a key, a key confidence, a mode, and mode confidence. We take the subset which is aligned with the Lakh MIDI Dataset [4] and the pieces which have a key confidence and mode confidence each greater than 0.5, 14122 in total.

It is important to note that the distribution of keys in the datasets is significantly skewed. Overall, we have 12014 major pieces and only 4896 minor pieces. Figures 1, and 2 investigate this skew further.

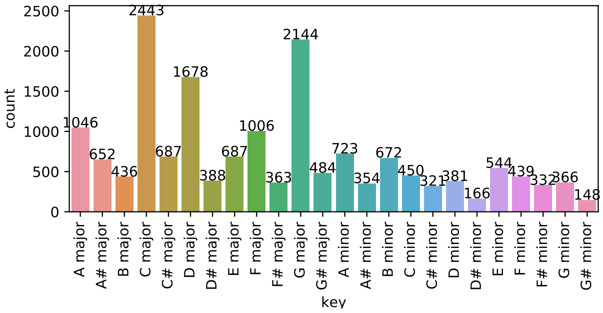

Figure 1. Counts of keys from all datasets.

Attribution: James Owers, Andrew McLeod. "OM{1,2,3}: MIREX Audio Key Detection", 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

[1] We use librosa's hybrid_cqt.
[2] GTZAN labels are taken from https://github.com/alexanderlerch/gtzan_key
[3] https://github.com/GiantSteps/giantsteps-mtg-key-dataset

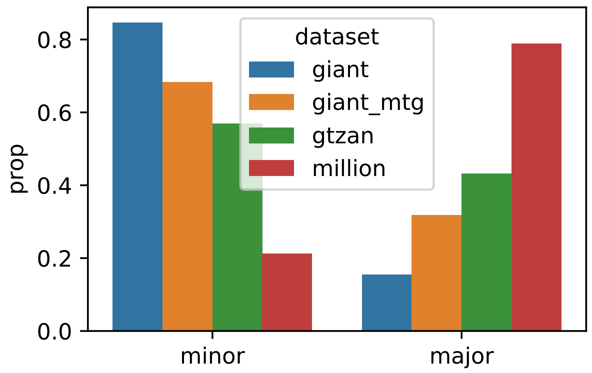

Figure 2. Proportion of major/minor keys in each dataset.

3.1 Augmentation

After splitting our data randomly into train and test sets, we perform data augmentation on our training set to try to increase our models' generalisation across keys. In particular, we augment the data so that each possible key (both major and minor) has exactly the same number of pieces. This is in contrast to previous work, such as [3] and [5], which augment each piece to every possible tonal centre, resulting in no change to the skewed distribution of keys.

We augment each possible key up to the maximum count (2443 of C major) by pitch shifting pieces from surrounding keys to it (in order of increasing pitch difference) until the desired count of 2443 was reached. This results in 31741 augmented pieces in the training set, compared to a total of 12683 non-augmented pieces.

4. RESULTS

We found it surprisingly difficult to beat our simple baseline **OM3**. The confusion matrices in Figure 3 show that the baseline does generally well across all keys, and **OM1** and **OM2** both struggle on certain keys. **OM1** in particular struggles with rare keys, even with data augmentation: C# major, and A# and G# minor. **OM2** performs poorly on C#, F, and G major, and A, B, C#, and D# minor. The overall score of each model on our test set is found in Table 1.

Figures 4 and 5 investigate each model's performance on the train and test sets respectively, broken down by each type of error. From these figures it can be seen that, while **OM1** and **OM3** seem to perform similarly, **OM2** doesn't get as many keys correct, instead making perfect 5th or relative key errors. Figures 6 and 7 show **OM1** and **OM2**'s errors on the test set respectively, divided by dataset. From these plots, two things seem clear: (1) GTZAN and MSD seem to be easier datasets than either Giant Steps or Giant Steps MTG (it is worth remembering that the latter two datsets skew more heavily towards minor keys, of which there is less non-augmented training data); and (2) **OM2**'s additional perfect 5th and relative key errors seem to come mostly from from GTZAN and MSD.

While **OM1** does outperform **OM2** on our data, both seem to underperform [3] significantly (although the test sets are different). This suggests that either (1) our reimplementation of [3] is incorrect, or (2) these models are particularly sensitive to the training data and procedure.

Model	OM1	OM2	OM3
Score	72.2	62.7	69.9

Table 1. The overall score of each model on our test set.

5. CONCLUSION

Whether our new model **OM1** has learned any temporal information has not been investigated. Given the minimal increase in performance over the baseline **OM3**, which explicitly has no access to any temporal information, it seems unlikely. Future work will evaluate this explicitly and tweak the model until it does use time. Additionally, the difference in performance of all of our models between our test set and the published MIREX results should be investigated. It seems that our models are particularly sensitive to the specific training distribution. We would ideally like a model which is able to generalise across keys, no matter the training set distribution. Furthermore, that **OM3** outperforms some models on some datasets, even in the official evaluation, shows that the problem can theoretically be addressed with very few parameters, drawing into question whether the complexity of our other models is warranted.

This work was partially funded by EU ERC H2020 Advanced Fellowship GA 742137 SEMANTAX and a Google faculty award, and also by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

6. REFERENCES

[1] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, pages 591–596, 2011.

[2] Peter Knees, Ángel Faraldo, Perfecto Herrera, Richard Vogl, Sebastian Böck, Florian Hörschläger, and Mickael Le Goff. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *ISMIR*, 2015.

[3] Filip Korzeniowski and Gerhard Widmer. End-to-end musical key estimation using a convolutional neural network. In *European Signal Processing Conference (EUSIPCO)*, pages 966–970. IEEE, 2017.

[4] Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. PhD thesis, 2016.

[5] Hendrik Schreiber. MIREX 2017 CNN-based automatic musical key detection submissions HS1/HS2/HS3. 2017.

[6] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

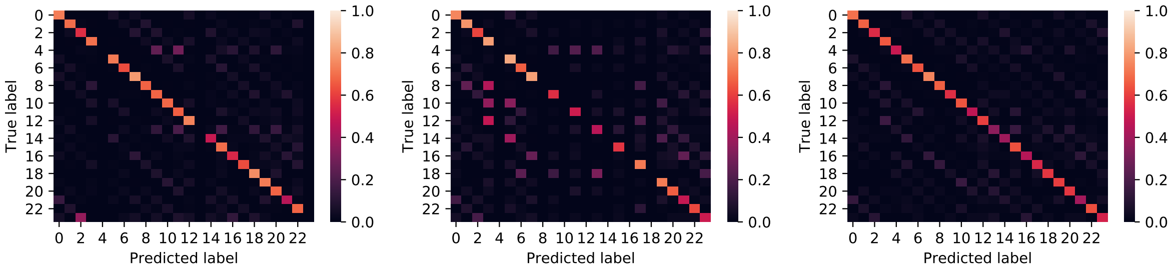

Figure 3. Confusion matrices for the three models (OM1, 2, and 3, left-to-right) over the randomly selected test dataset. Labels run from 0 = A major up to 11 = G# major, then 12 = A minor up to 23 = G# minor.

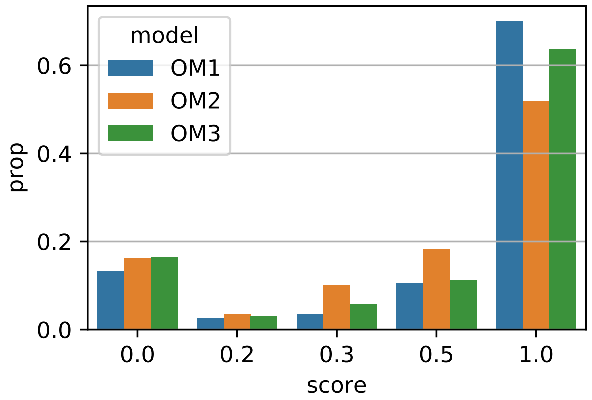

Figure 4. Comparison of scores on training data between models. Scores are defined as: 1=key correct, 0.5=perfect 5th off (either direction), 0.3=relative minor/major, 0.2=parallel minor/major, 0=other error.

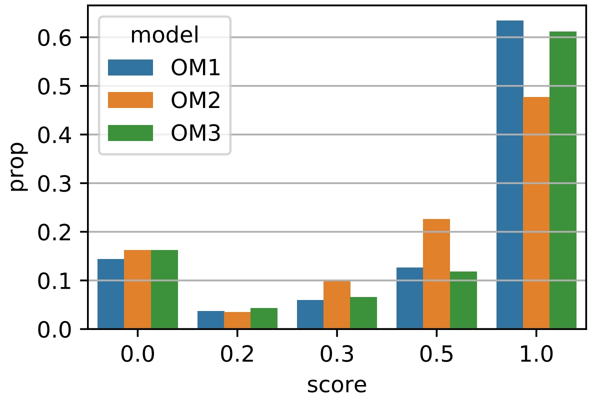

Figure 5. Comparison of scores on test data between models. Scores are defined as: 1=key correct, 0.5=perfect 5th off (either direction), 0.3=relative minor/major, 0.2=parallel minor/major, 0=other error.

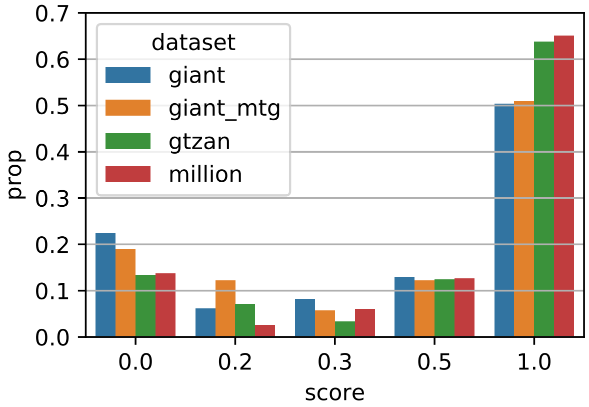

Figure 6. Comparison of OM1's scores on test data broken down by source dataset.

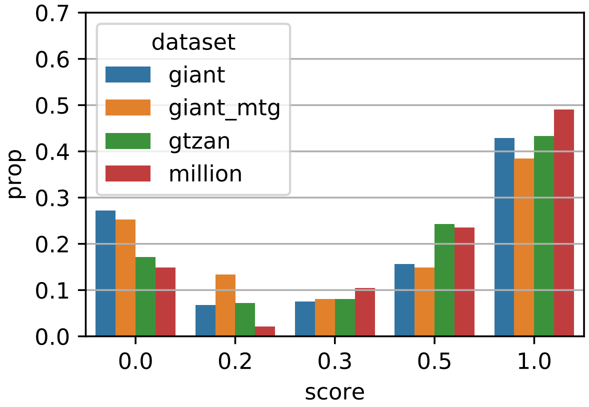

Figure 7. Comparison of OM2's scores on test data broken down by source dataset.