Deep Learning for Audio and Music

Two-vocal Separation and Singing Pitch Transcription

Wen Qing Lim
MSc Sound and Music Computing

INTRODUCTION

Imagine singing your heart out in a karaoke room but your tone deaf friend is trying to harmonise in the background. How can we remove their voice so you can show off a recording with just your perfect vocals only? Furthermore, how can we prove that your vocals were, indeed, perfect in pitch?

This project aims to create a pipeline that can achieve the above, by first applying source separation to split two concurrent singing vocals, and then transcribing the singing pitch for the extracted vocals-only tracks.

DATASET

The dataset used for both segments is MIR-1K, which consists of 1000 tracks of 8 female and 11 male researchers singing to Mandarin Pop songs. It was originally created for the problem of separating vocals from background music, and the tracks have the accompaniment and vocals split into left and right channels respectively. It also contains pitch labels at 20ms intervals.

The data was split by artist name into training, validation and test sets with a 80-10-10 split, resulting in 15 artists in the training set, 2 in the validation, and 2 in the test set. This ensured that artists in the validation and test sets were not seen by the models during training, to avoid the album effect.

IMPLEMENTATION

I. MULTI-SINGER SEPARATION

Data Pre-processing

To mimic a multi-singer scenario, the vocals-only tracks were paired and mixed to create a dataset of paired vocals. This simplifies the singer separation task, since it is arguably easier to separate 2 voices singing at different tempos and non-harmonised pitches, than to separate a synced duet where 2 vocals are in matching tempo and with harmonised voices. (However, let's just pretend your friend can't keep to the tempo, has no sense of pitch harmony, and mixes up the lyrics).

The pairing method was inspired by Chen et al (2002) [3], which conducted a similar pairing strategy to create a duet singing dataset for the same task. To simplify the evaluation process, the mixed tracks were split into 3-second chunks, up to the length of the shorter vocal track.

Model

PyTorch's pre-trained Conv-TasNet was used to conduct multi-singer separation. The model was pre-trained on Libri2Mix, a speech separation dataset, for multi-speaker separation task. Since singing vocals share similarities with speaking vocals, the pre-trained model was expected to perform decently on multi-singer separation. However, in the case of synced duets, additional fine-tuning might be needed for the model to learn to split voices in harmony.

Evaluation on Test set

The metrics used to evaluate the quality of the separated vocals are the same as in [3]:
1. SDRi - Source-to-distortion Ratio improvement, which is a measure how good each track sounds, and takes the difference between the SDR of the predicted and ground-truth separated vocals, and the SDR between the mixed and ground-truth separated vocals. This translates to the improvement in SDR that came from separating the 2 vocals.
2. SI-SNRi - Scale-invariant Signal-to-Noise Ratio improvement, which was proposed as a more robust metric compared to SDR. Similarly, this translates to the improvement in SI-SNR that came from separating the 2 vocals.

	SI-SNRi	SDRi
train	14.078300	14.345107
val	3.673919	4.991039
test	10.904131	11.782592

Table 1: Vocal separation metrics on a sample of 500 vocal pairs

Conv-TasNet achieved an average SI-SNRi of 10.90dB and average SDRi of 11.78dB on a random sample of 500 pairs from the Test set, 14.08/14.35Db, and 3.67/4.99Db on the training and validation sets respectively. Interestingly, the model performed much worse on the validation set than on the training and test sets. By observing the distribution of SI-SNRi and SDRi in Figure 1 below, it is clear that the Validation set is much more difficult to separate, since most of the samples had far lower scores. Since the validation set only consists of 2 artists *Titan* and *Ariel*, perhaps the two voices happen to be more difficult to separate. When listening to an example of the separated vocals, perceptually, both are female singers with similar vocal range and timbre, which might result in the lower scores. Furthermore, trying to separate a recording of my own vocals (2 recordings of the same song with different pitch) resulted in almost no separation at all. While there could be other factors (e.g. reverb, different recording conditions), it is clear that the pre-trained Conv-TasNet does not work well for singing voices that are similar.

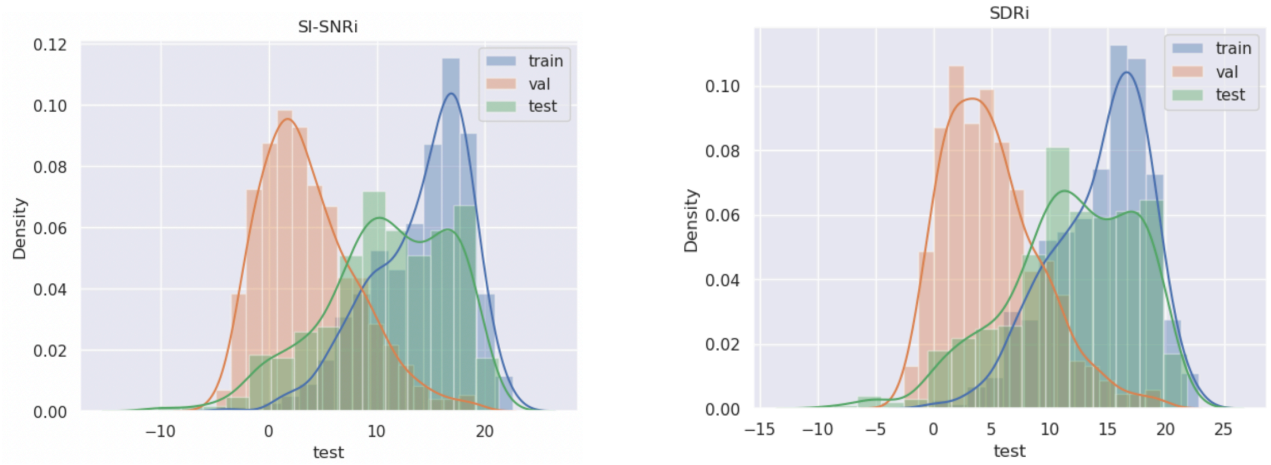


Figure 1: SI-SNRi and SDRi distributions between each dataset

II. SINGING PITCH TRANSCRIPTION

Data Pre-processing

For this task, only the vocal channel from the MIR-1K tracks are extracted from the dataset. The pitch labels are semitone pitch-contours ranging from 36.40 to 76.19. They are binned to the nearest 10-cent and converted to their corresponding note values that range from C2+0 to F5-10. This results in a total of 410 label pitch classes that in 10-cent intervals. Additionally, a 411th label was added for no-pitch predictions.

Model

CREPE [1] was implemented, and trained on only the vocal channel of the MIR-1K tracks. CREPE (Convolutional Representation for Pitch Estimation) is a Convolutional Neural Network that takes raw time-domain audio signals as input, without using any transformations to spectrograms. The model uses a simple architecture of 6 convolutional layers and a single fully-connected layer in the output layer. While the paper describes an output size of 360 nodes, the model implemented in this project outputs 410 nodes - each representing 10 cents in pitch bins.

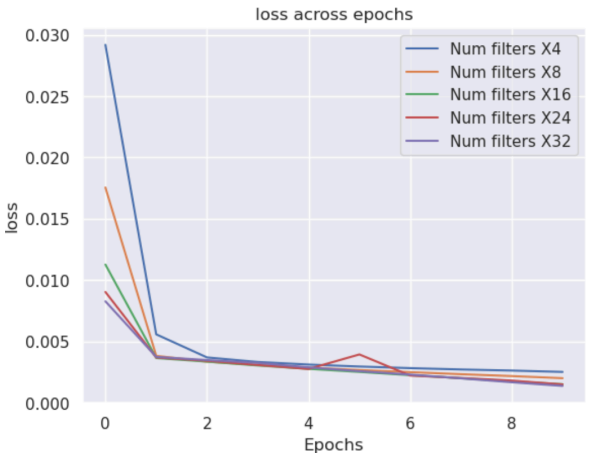
While the paper describes the full model architecture, the source code [2] for CREPE documented multiple model sizes. The different model sizes were determined by the number of filters in each layer, which differed in multiples of 4, 8, 16, 24, and 32.

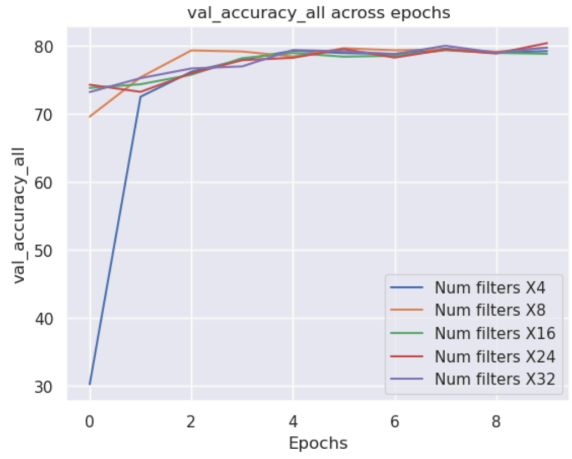
Evaluation metrics

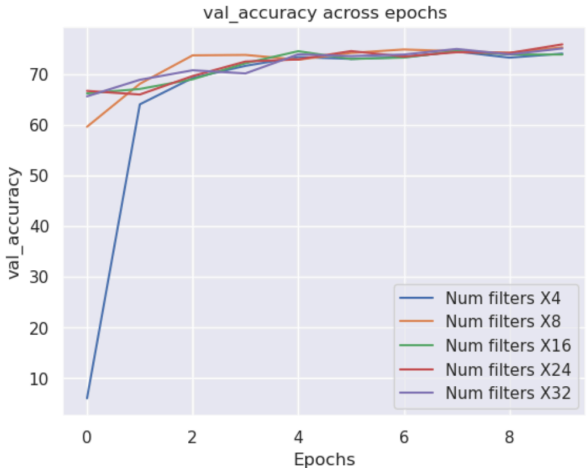
Validation metrics

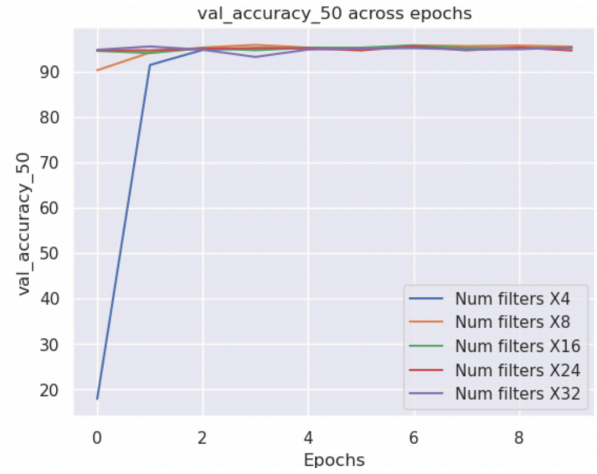
The model was trained across 10 epochs for each model size, and validated on the validation set. The validation accuracy improved across the first 3 epochs, but improvement was only marginal after the 4th epoch, despite decreasing loss. The largest model (x32) achieved a validation accuracy of 75.0% for exact pitch identification, and 95.2% with a half-semitone (50 cent) error allowance.

In the figure below, we observe that there is not much difference in model performance from the X8 model onwards, suggesting that there might not be a need for a large model, and that CREPE can perform relatively well with just a X8 or X16 model.



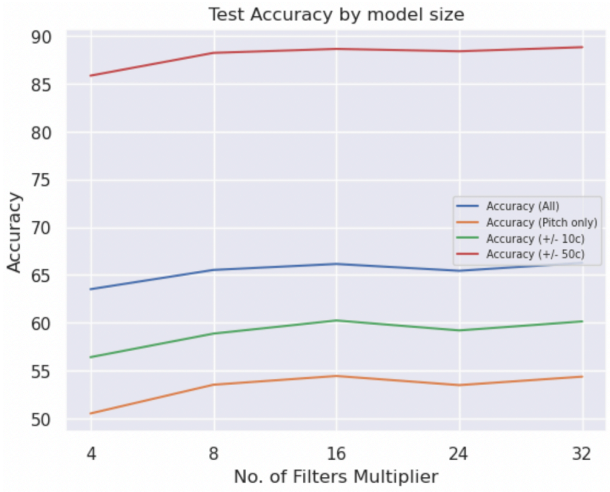






Test set metrics

num_filters_x	accuracy_all	accuracy	accuracy_10	accuracy_50
4	63.521016	50.526172	56.410968	85.848796
8	65.533304	53.530878	58.875656	88.230407
16	66.157812	54.444754	60.246468	88.645804
24	65.444094	53.489339	59.207976	88.396567
32	66.276765	54.375517	60.149544	88.825810

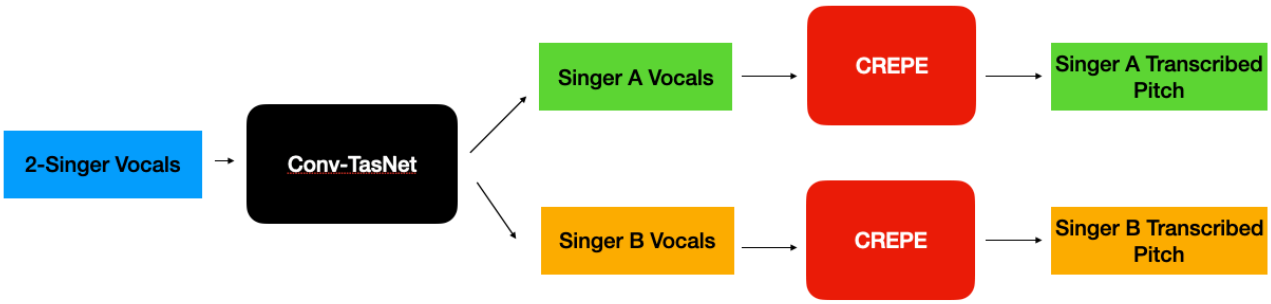


Evaluating each trained model on our test set, we find that the model performance does not increase by much for the model size 16 and above. Our model achieved an accuracy of 88.8% on the Test set with an error allowance of 50 cents, but had a far lower accuracy of 54.4% for exact pitch accuracy. This might be indicative of how generalisable the model is on unseen voices - it can perform well up to half a semitone, but it cannot do well on exact pitch transcription.

One thing of note is that the accuracy and accuracy_10 metric now see a difference, which was previously not observed in the validation metrics (See Appendix A). This might point towards the quality of the labels in the validation set, where the hand labelled pitches might have been consistently more than 10cents off from the labels in the training or test sets.

III. COMBINING (I) and (II)

The two models are combined by simply calling them sequentially, and the process is shown below.



The input to the pipeline is an audio file with two concurrent singing vocals, without background music. This input is passed into the pre-trained Conv-TasNet model to retrieve two separated vocal tracks. The two tracks are then passed into the trained CREPE model, which outputs two sets of pitch transcriptions for each vocal track.

Case Study

To study the performance of the combined pipeline, we selected *Leon_7* and *Jmzen_5* from the Test set, since they had similar tempos of about 117BPM.

After mixing both full tracks, Conv-TasNet was used to separate the 2 vocals. However, a listening test revealed that the 2 split vocals do have different singers each, but they switch between singers in each track. This might be due to the fact that there were many moments in the mixed track where the vocals were mot overlapping but each vocal was during the pauses of the other track. This resulted in an SI-SNRi of -0.652 and SDRi of -0.319, which was not great.

Consequenty, we tried mixing only one part of the 2 tracks, *Leon_7_03* and *Jmzen_5_03*. This resulted in an SI-SNRi of 15.69 and SDRi of 15.92, which was a huge improvement. From this observation, perhaps the pre-trained Conv-TasNet is not suitable for long vocal tracks, but for only for shorter clips.

Next, we take the separated vocals and resample them to 16KHz, which is expected by the CREPE model we trained. We then pass them into our CREPE models, comparing the performances of different model sizes as shown below:

	accuracy_all	accuracy	accuracy_10	accuracy_50
4	15.463917	19.108281	19.320594	27.600849
8	22.238587	29.936305	30.148619	41.188958
16	28.571430	34.607220	34.819531	42.887473
24	36.082473	34.394905	34.819531	52.229297
32	39.911634	44.585988	45.859873	73.036093

	accuracy_all	accuracy	accuracy_10	accuracy_50
4	6.185567	6.712963	8.101851	12.268519
8	4.565538	7.175926	7.407407	11.111111
16	21.060383	12.962963	13.888890	21.296297
24	29.307806	22.453703	26.620370	39.120370
32	25.920472	22.222222	26.851851	46.064815

Left: Metrics for leon_7_03; right: Metrics for jmzen_5_03

The pitches for *Leon_7_03* had better performance than *Jmzen_5_03*, due to the noticeably worse quality of audio in the split Jmzen_5_03 track. Furthermore, the performance for both are far worse than our test metrics in section II, implying that the vocal separator added noise to the tracks that might have affected the pitch transcription performance. Also, the difference in performance between model sizes is much clearer here, where the X32 model outperforms smaller models by a fair amount. Hence, the size 32 model is more robust to noise and quality distortions than the smaller models.

CONCLUSION

We have trained a Convolutional Neural Network, CREPE, that performs well on pitch transcription on clean vocals, and found that a pre-trained Conv-TasNet model works well on separating vocals that are short and dissimilar.

When combining both models, we found that the quality of separated vocals greatly affects the pitches transcribed in the second model. While larger CREPE models are not necessary for training and testing on vocals with similar quality recordings, a larger CREPE model is able to perform better on more noisy, distorted recordings that are output by the Conv-TasNet separator.

For further improvement, perhaps the source separator can be fine-tuned to the dataset to split singing voices. If using MIR-1K as a dataset, perhaps more work can be done to mimic a singing duet dataset with more realistic harmony and overlapping vocals, such as matching vocals of the same Tempo, similar Onset timings, and with same or harmonic Keys.

Looks like we won't know how perfect our singing is, yet, but we'd definitely be humbled when CREPE tells us our pitch aren't so accurate.

REFERENCES

[1] Kim, J. W., Salamon, J., Li, P., & Bello, J. P. (2018, April). Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 161-165). IEEE.

[2] https://github.com/marl/crepe/blob/master/crepe/core.py

[3] Chen, H. Y., Chen, X., & Jang, J. S. R. (2021). Singer separation for karaoke content generation. *arXiv preprint arXiv:2110.06707*.

APPENDIX

Table of metrics for Model X4 during training, for all 10 epochs. Notice that val_accuracy and val_accuracy_10 are the same:

loss	val_accuracy_all	val_accuracy	val_accuracy_10	val_accuracy_50
0.029185	30.237678	5.925778	5.925778	17.777334
0.005574	72.491199	64.006382	64.006382	91.400641
0.003690	76.159036	69.213885	69.213885	94.732642
0.003321	77.934271	71.608138	71.608138	95.111734
0.003110	79.254699	73.343974	73.343974	95.091778
0.002950	78.902584	72.984838	72.984838	95.051873
0.002818	78.785211	73.324025	73.324025	95.530725
0.002709	79.548120	74.441344	74.441344	95.011973
0.002621	78.946596	73.224264	73.224264	95.071828
0.002509	79.196012	74.022347	74.022347	95.291299