MusicYOLO

A Vision-Based Framework for Automatic Singing Transcription

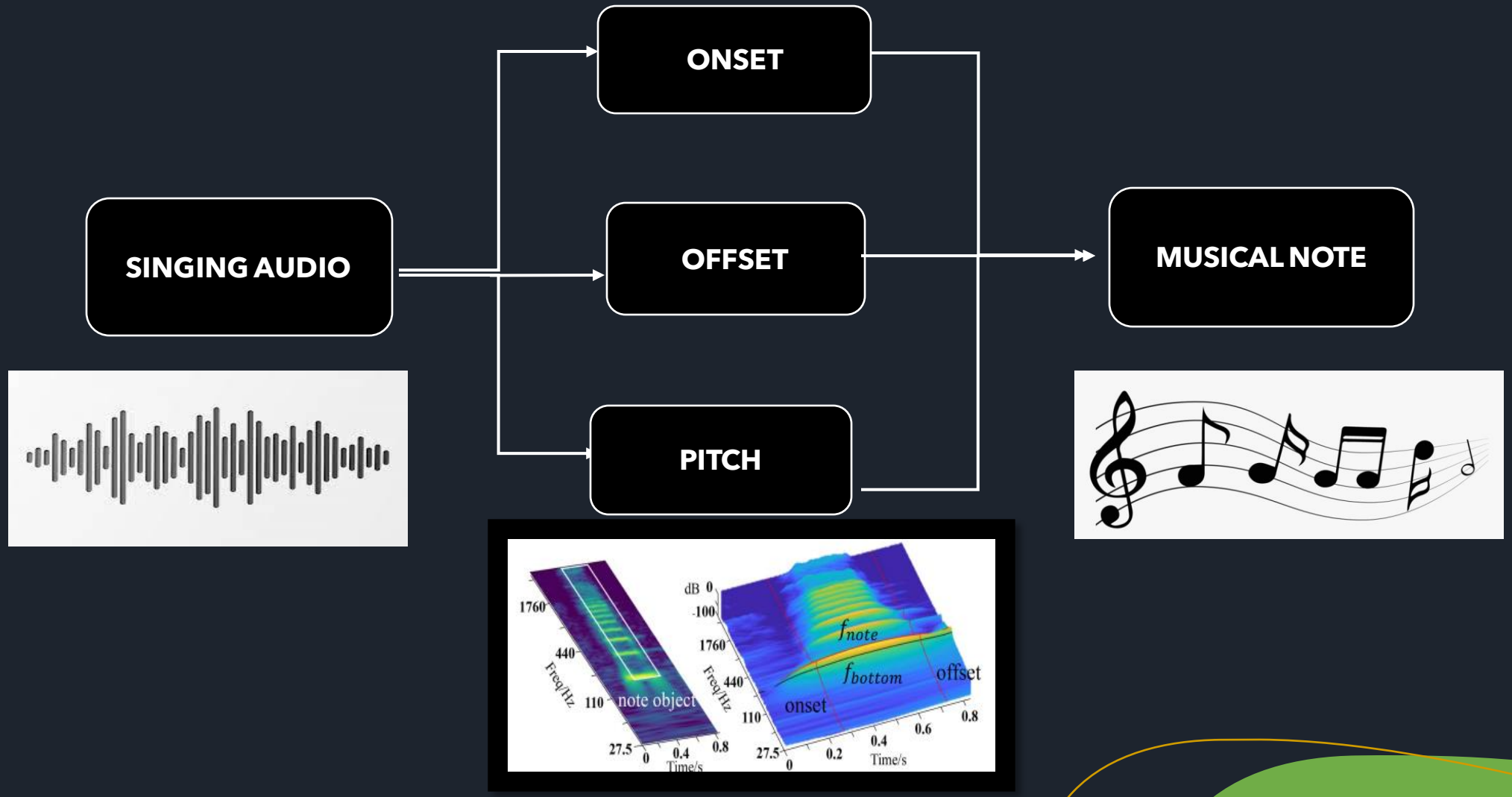
Xianke Wang , Bowen Tian , Weiming Yang, Wei Xu and Wenqing Cheng

Amruta Mahendra Parulekar
- 20D070009

Keshav Singhal
- 20D070047

Sameep Chattopadhyay
- 20D070067

What is Automatic Singing Transcription?



Some Basic Concepts

- Note segmentation
- Pitch extraction
- Singing transcription
- Audio event detection

Some Basic concepts

Note Segmentation

- Extracting the onsets and offsets from the given audio
- Segmentation necessary for subsequent pitch extraction
- Two Methods- Pitch Information & Spectrogram Features

Pitch Extraction

- Estimation of the Fo trajectory from the given audio signal
- Traditional Methods based on time & frequency domain parameters
- Current State of the Art Method: LSTM-RNN

Some Basic concepts

Singing Transcription

- Challenging task even with monophonic signals
- Use of audio processing algorithms similar to contour & edge detection
- Current State of Art Method: RNN-HMM

Audio Event Detection

- Research methods applied for both human & instruments audio detection
- Traditional Methods used Support Vector Machines
- Current State of the Art Method: R-CNN

Earlier AST Models

Method 1 : End-to-end approach to obtain onset, offset, and pitch probabilities. (note-level results through post-processing)

Method 2: Onset & offset obtained first; pitch obtained through pitch extraction

Method 3: F0 tracking conducted to obtain a frame-level pitch curve; note-level onset, offset, and pitch obtained through post-processing

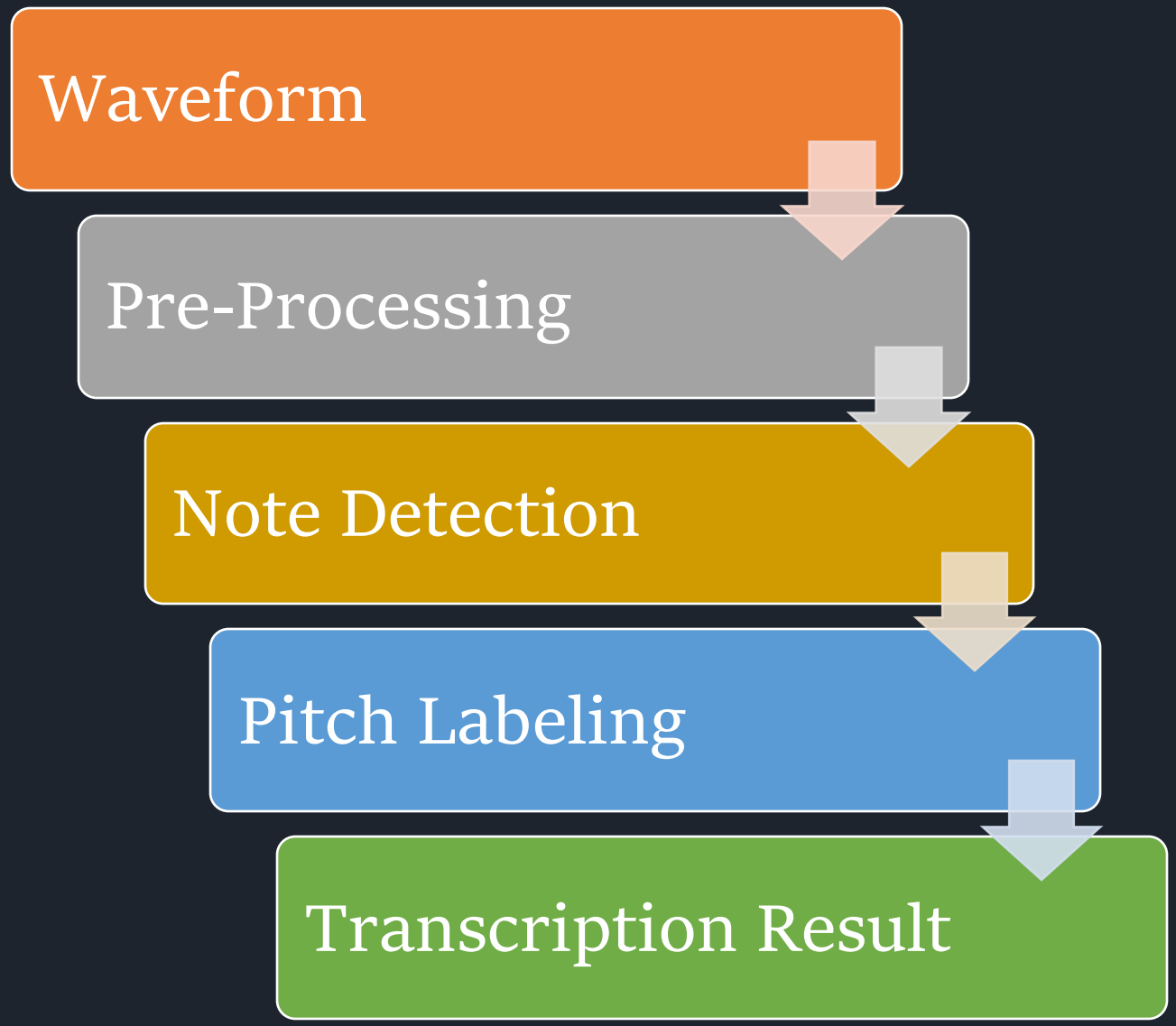
MusicYOLO Method

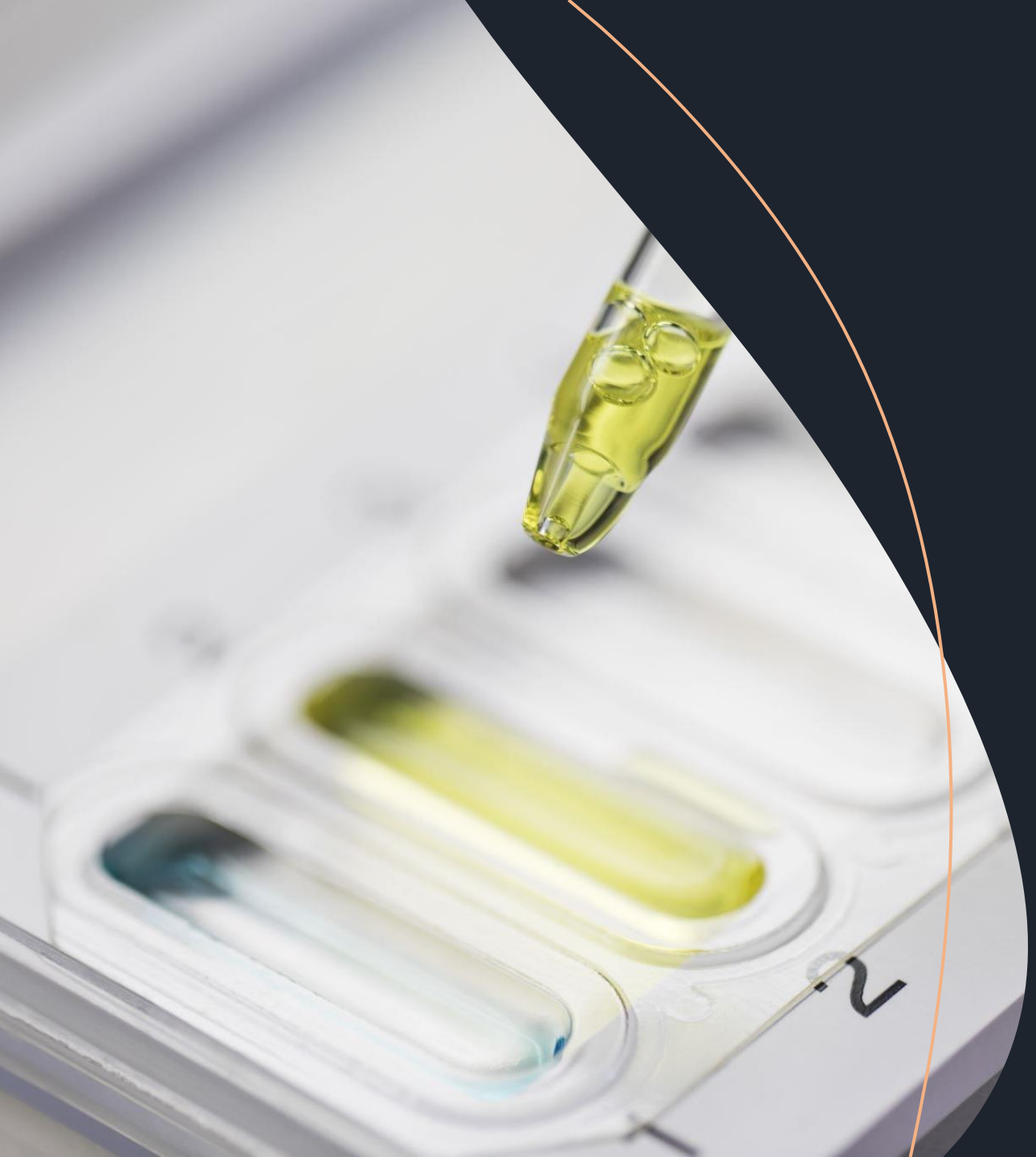
MusicYOLO inspired by the perspective of sound event detection and based on object detection

Uses YOLOX for onset/offset detection & spectrogram peak search for pitch labeling

Detects note objects rather than isolated onset/offset moments, thus greatly enhancing the transcription performance

The MusicYOLO Model





Testing Environment

Datasets

SSVD	Sight-Singing Transcription Dataset
ISMIR2014, MIR-ST500	Singing Transcription Dataset
Bach10	Monophonic Instrument Dataset

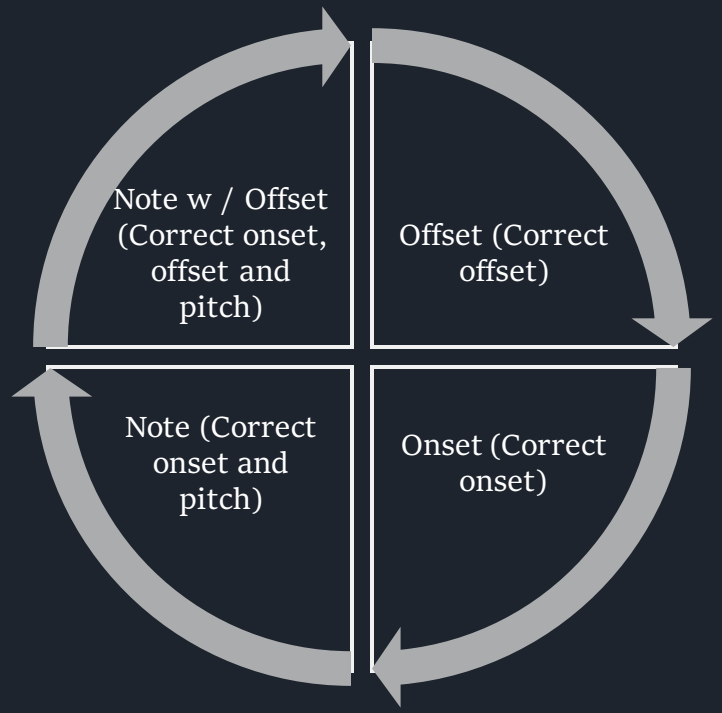
Evaluation Metrics

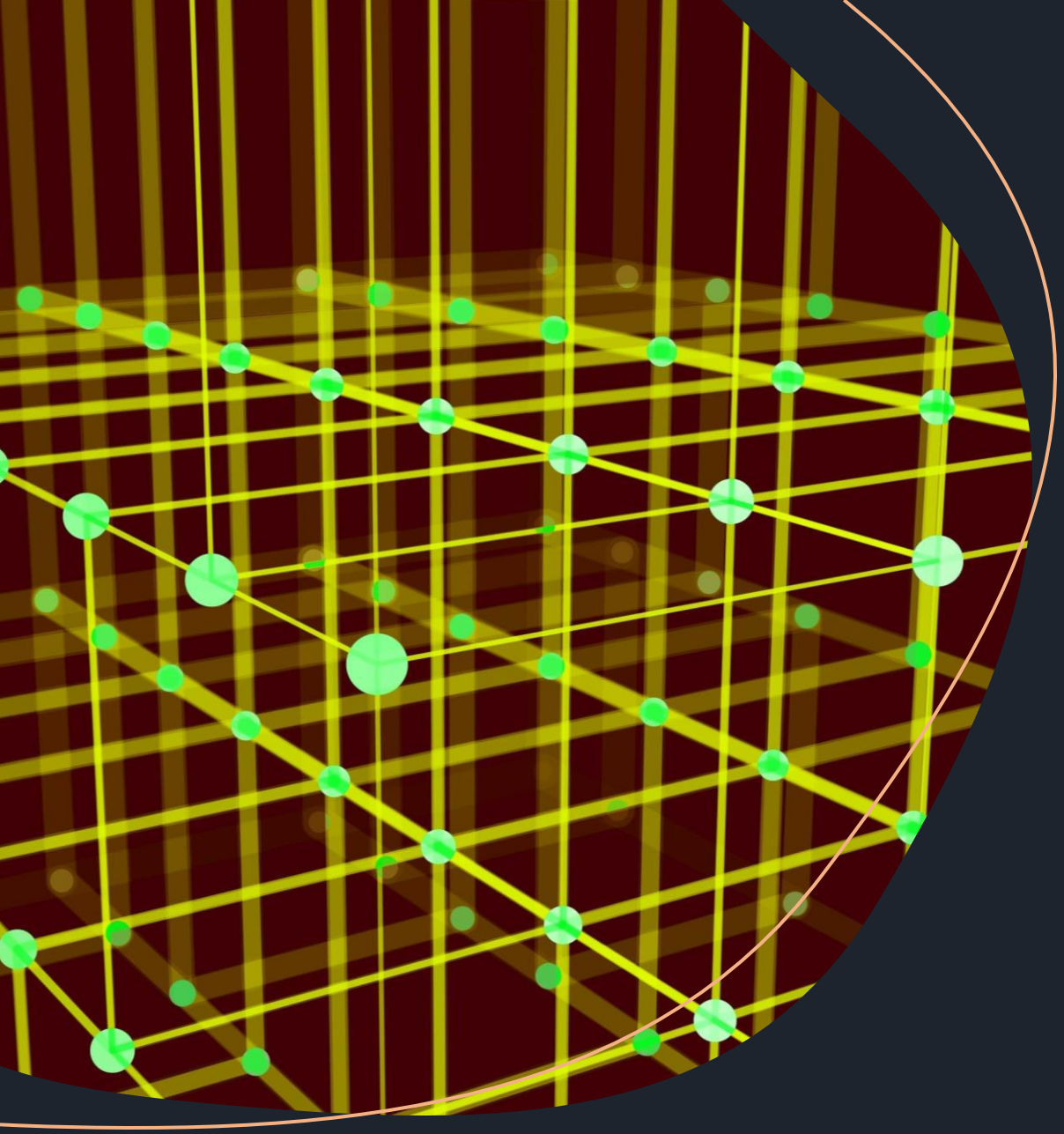
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}}$$

Using F1 Score





Algorithms

Preprocessing

- Audio Noise Reduction
- Time Frequency Transformation
- Linear Intensity Matching
- Spectrogram Cutting

Audio Noise Reduction

SPLEETER

HUMAN VOICE + INSTRUMENTAL

AMPLITUDE SPECTROGRAM

UNET FOR HUMAN VOICE

MASK FOR HUMAN VOICE

UNET FOR INSTRUMENTAL

MASK FOR INSTRUMENTAL

MASK

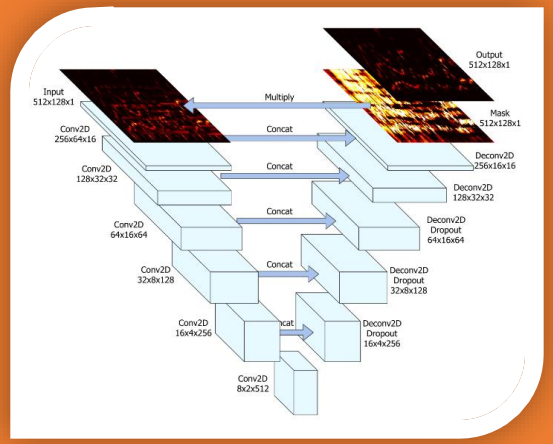
HUMAN VOICE + INSTRUMENTAL

AMPLITUDE SPECTROGRAM

HUMAN VOICE

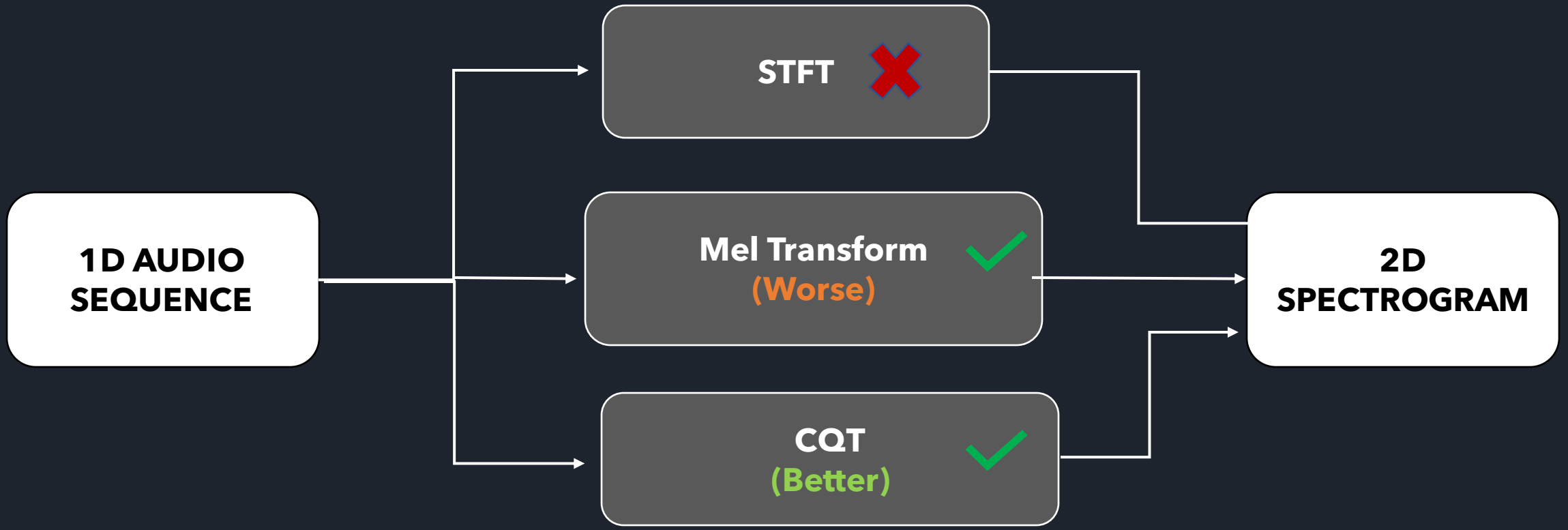
INSTRUMENTAL

AMPLITUDE SPECTROGRAM



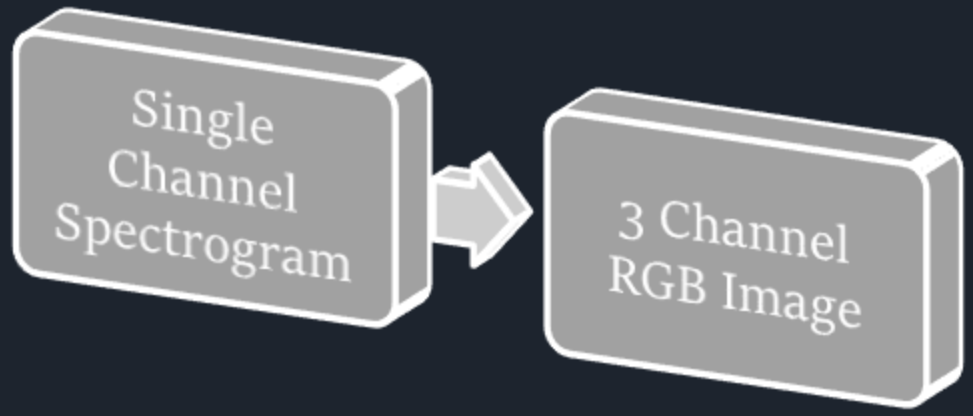
ARCHITECTURE

Time Frequency Transformation



Linear Intensity Matching

A linear intensity mapping effectively quantifies the original spectrogram according to the spectral intensity.



Spectrogram Cutting

Spectrogram Image (Aspect ratio too large to conveniently process)

Obtain Ratio

$$ratio = w/h - 1$$

where w and h represent the width and height of slice

$$w = frame\_len \cdot scale$$

$$h = n\_bins \cdot scale$$

where $frame\_len$ is the frame length of the CQT matrix, n_bins is the frequency bins of CQT, and $scale$ is the scaling factor from the matrix size to image size. $frame\_len$ is calculated as follows:

$$frame\_len = t \cdot sr/hop\_len$$

where sr and hop_len are the CQT parameters, and t is the time length corresponding to the slice. From the above, we derive the $ratio$ calculation function written as R:

$$R(t) = t \cdot \frac{sr}{hop\_len \cdot n\_bins}$$

Split audio into unmute segments

Obtain Segmentation Points

initialize $\boldsymbol{splits}$;
set the iteration number $r = 1$ and the maximum iteration number M;
repeat
  $last\_time = \boldsymbol{splits}(-1)$;
  $ratio = R(end_i - last\_time)$;
  if $ratio \geq max\_ratio$ **then**
    $\boldsymbol{splits} = [\boldsymbol{splits},\ end_i]$;
  else if $best\_ratio < ratio \leq max\_ratio$**then**
    $ratio\_silence = R(start_{i+1} - last\_time)$;
    $ratio\_next = R(end_{i+1} - start_{i+1})$;
    if $ratio\_silence > max\_ratio$ **then**
      $\boldsymbol{splits} = C(\boldsymbol{splits},\ S_{end\_i},\ start_{i+1})$;
    else if $ratio\_next > max\_ratio$**then**
      $\boldsymbol{splits} = [\boldsymbol{splits},\ start_{i+1}]$;
    else
      $\boldsymbol{splits} = [\boldsymbol{splits},\ end_i]$;
  else
    A process similar to the above;
  $r \leftarrow r + 1$;
until $r > M$.

Obtain approximately square slices along the spectrogram time axis

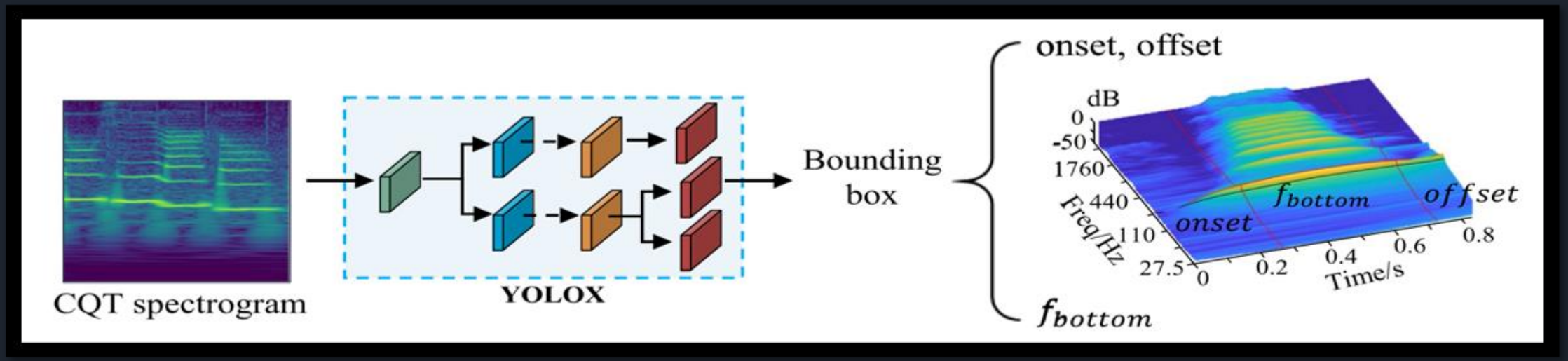
Note Detection

Note Object Detection

Post-processing the Bounding Box

Time Shift

Note Object Detection

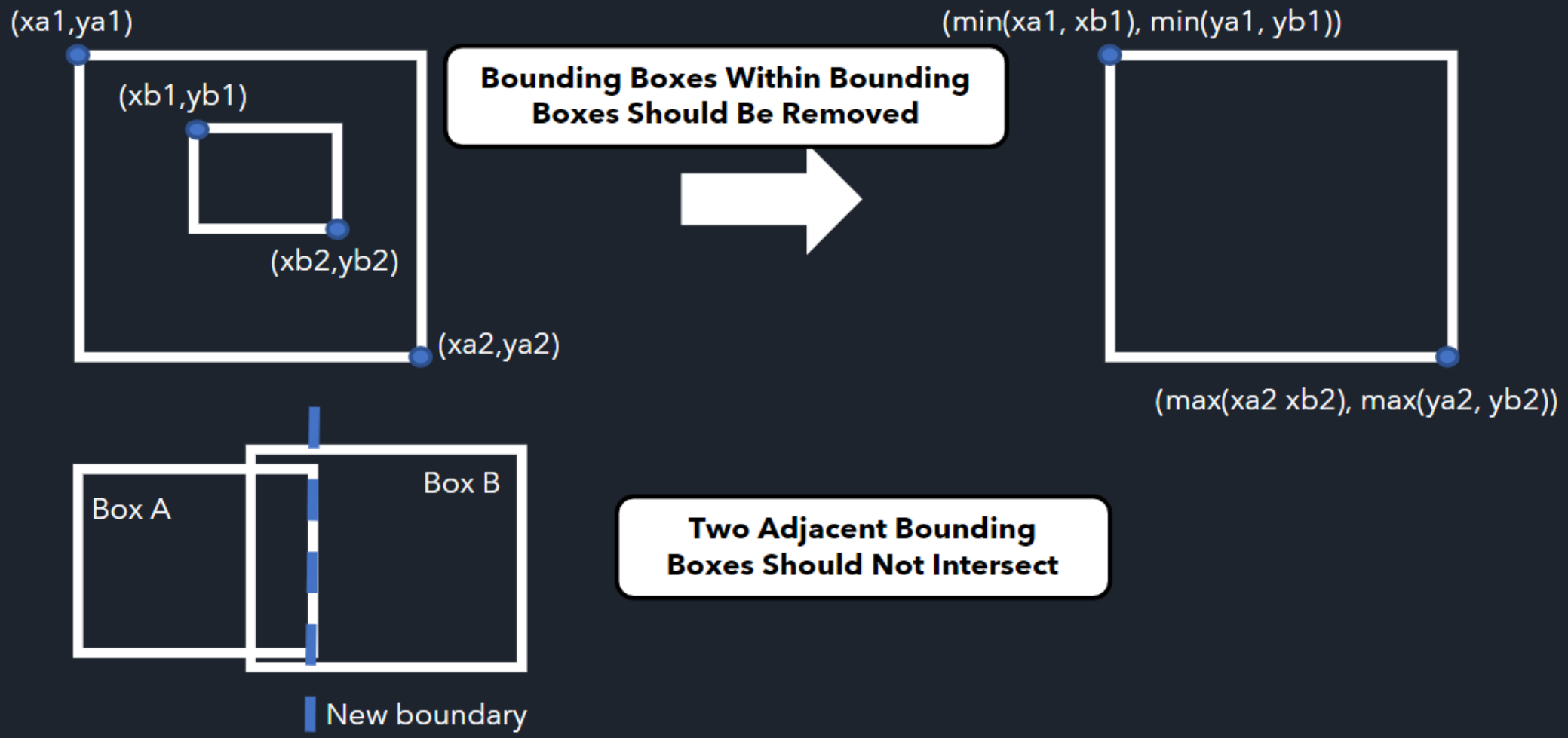


Note Object

Object detection to get bounding box

Left boundary = onset
Right boundary = offset
Bottom boundary = f_bottom

Post-processing the Bounding Box



Time Shift

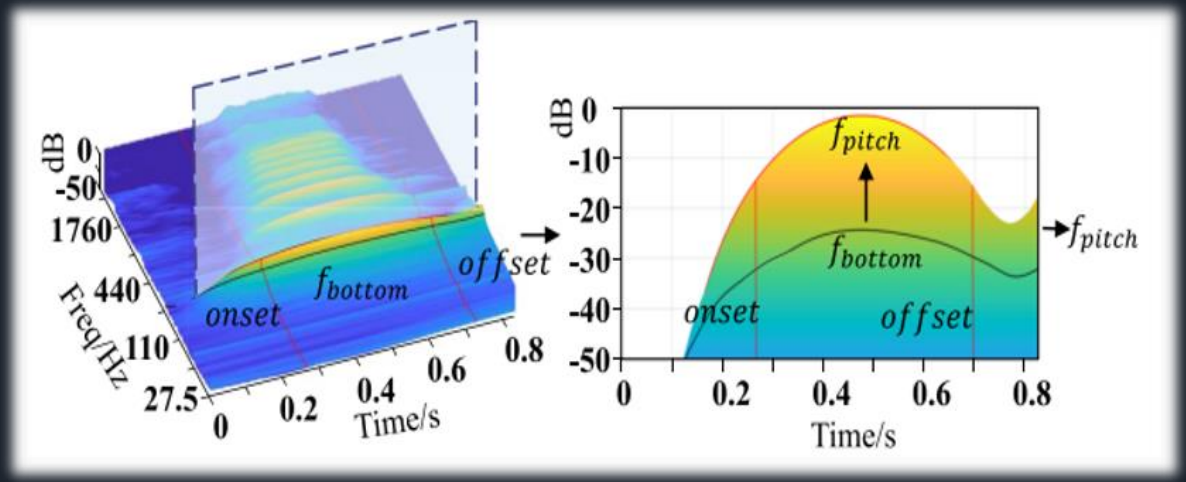
- We shift the onset and offset to obtain the complete audio's onset and offset values.
- As we have previously cut the spectrogram, we splice all image slices along the time axis and add the corresponding time shift to the previously detected onset and offset.

Pitch Labelling

The Peak Search Algorithm

- The errors between F_bottom and F_pitch (actual frequency) are due to Sacrifice of frequency resolution due to time resolution and Pitch fluctuations due to vibrato and portamento

- These can be tackled using the peak search algorithm



Require: Cqt: spetrogram matrix; $onset_i, offset_i$: onset and offset of the ith detected note; f_{bottom_i}: bottom frequency of the ith detected note; bpo: the number of frequency bins per octave; Δb: the search range of frequency bins.

Ensure: $Pitch$: a pitch list of the detected notes.

set N as the number of the notes
foreach $i \in \{1, 2, \ldots, N\}$ **do**
  $f_{init} = f_{bottom_i}$
  $\Delta f1 = f_{init} \times (1 - 2^{-\Delta b/bpo})$
  $\Delta f2 = f_{init} \times (2^{\Delta b/bpo} - 1)$
  $f_{low} = f_{init} - \Delta f1$
  $f_{high} = f_{init} + \Delta f2$
  $f_0 = 0$
  foreach $t \in \{onset_i, \ldots, offset_i\}$ **do**
    $f_0 = f_0 + f_{low} + argmax(Cqt[f_{low} : f_{high}, t])$
  $f_{pitch} = f_0/(offset_i - onset_i)$
  $Pitch_i = 69 + 12 \times \log(f_{pitch}/440)$

Results

Results For Different Model Configuration

NOTE DETECTION RESULTS OF DIFFERENT MODEL CONFIGURATIONS

		CQT	MEL	YOLOX	Faster RCNN	SSVD			ISMIR2014			Mixed		
						P	R	F	P	R	F	P	R	F
Onset	M1	✓		✓		95.03	95.65	**95.32**	92.40	89.39	**90.71**	93.23	92.00	**92.50**
	M2		✓	✓		95.25	95.28	95.24	89.85	86.46	89.85	94.37	90.45	92.21
	M3	✓			✓	92.33	94.79	93.45	89.53	85.20	87.00	91.44	89.86	90.43
	M4		✓		✓	92.06	94.57	93.19	89.87	84.60	86.85	91.15	89.39	90.03
Offset	M1	✓		✓		97.49	98.13	**97.78**	87.10	84.53	**85.65**	92.34	91.32	**91.73**
	M2		✓	✓		97.72	97.75	97.71	88.87	81.88	85.04	93.66	89.96	91.62
	M3	✓			✓	94.85	97.41	96.02	85.53	81.65	83.26	90.97	89.58	90.07
	M4		✓		✓	94.64	97.25	95.81	85.72	80.93	82.97	90.81	89.25	89.79

Why YOLOX ?

- The decoupling detection heads make the model converge faster and perform better.
- The use of anchor free for training eliminates the process of clustering to obtain prior bounding boxes from the dataset, which reduces the overfitting potential.
- SimOTA enables the model to automatically analyze the number of positive samples corresponding to each ground truth.

Note Detection Results

NOTE DETECTION RESULTS OF DIFFERENT MODELS

		ISMIR2014			MST500			SSVD			Bach10		
		P	R	F	P	R	F	P	R	F	P	R	F
Onset	Tony	72.13	64.19	67.63	50.98	56.88	53.45	79.00	78.96	78.90	91.94	88.55	90.15
	Fu&Su	83.00	75.43	78.76	50.78	53.97	52.10	79.70	81.04	80.11	87.17	79.41	83.00
	Wang I	81.09	82.49	81.60	73.50	78.59	75.67	76.88	82.44	79.31	87.38	88.65	87.61
	MusicYOLO I	89.55	82.76	85.88	81.89	75.02	**78.17**	88.02	86.88	87.41	92.56	84.68	88.31
	Wang II	88.38	40.15	53.90	65.07	29.70	40.34	91.36	65.79	75.70	84.74	24.48	37.21
	MusicYOLO II	91.01	89.24	**90.01**	62.53	65.20	63.59	95.99	96.38	**96.17**	93.97	94.19	**94.07**
Offset	Tony	79.88	70.37	74.47	55.82	62.44	58.59	93.59	93.50	93.43	95.71	92.16	93.84
	Fu&Su	79.60	72.94	75.87	53.26	56.82	54.74	84.48	86.09	84.99	92.39	84.07	87.91
	Wang I	76.39	77.86	76.95	65.25	69.89	67.25	71.86	77.78	74.43	87.00	88.70	87.43
	MusicYOLO I	83.43	77.07	79.99	78.78	72.01	**75.10**	85.14	84.21	84.64	97.24	88.80	92.68
	Wang II	21.39	9.12	12.46	26.50	12.05	16.38	13.84	10.00	11.45	42.47	12.33	18.80
	MusicYOLO II	85.79	84.34	**84.96**	62.89	65.57	63.94	97.89	98.28	**98.07**	96.80	97.06	**96.91**

Note Detection Error Analysis

- We classify the detection errors into four categories-

Successive Missing (SM)

- Notes missed consecutively

Isolated Missing (SM)

- Single note missed by algorithm

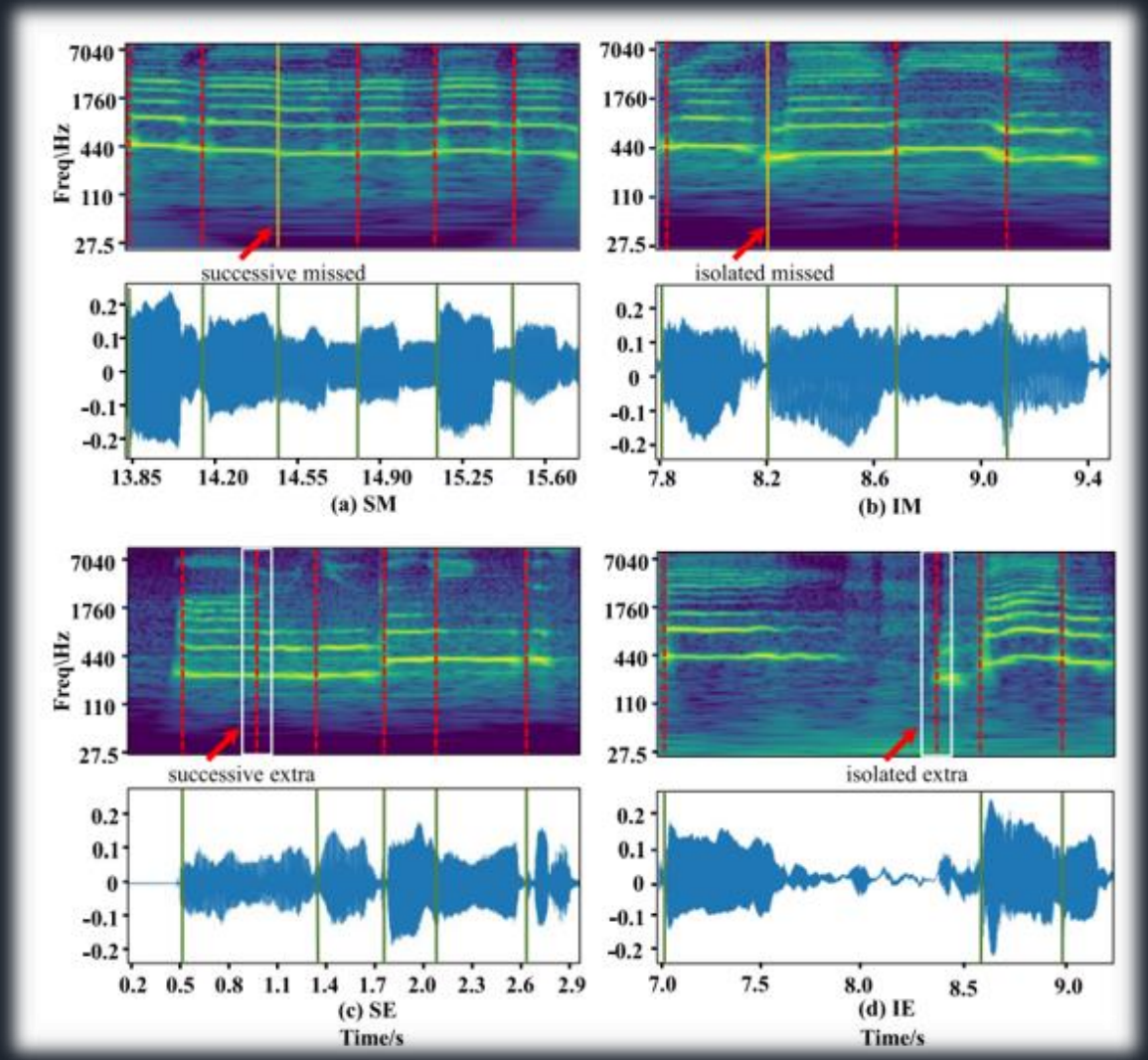
Successive Extras (SE)

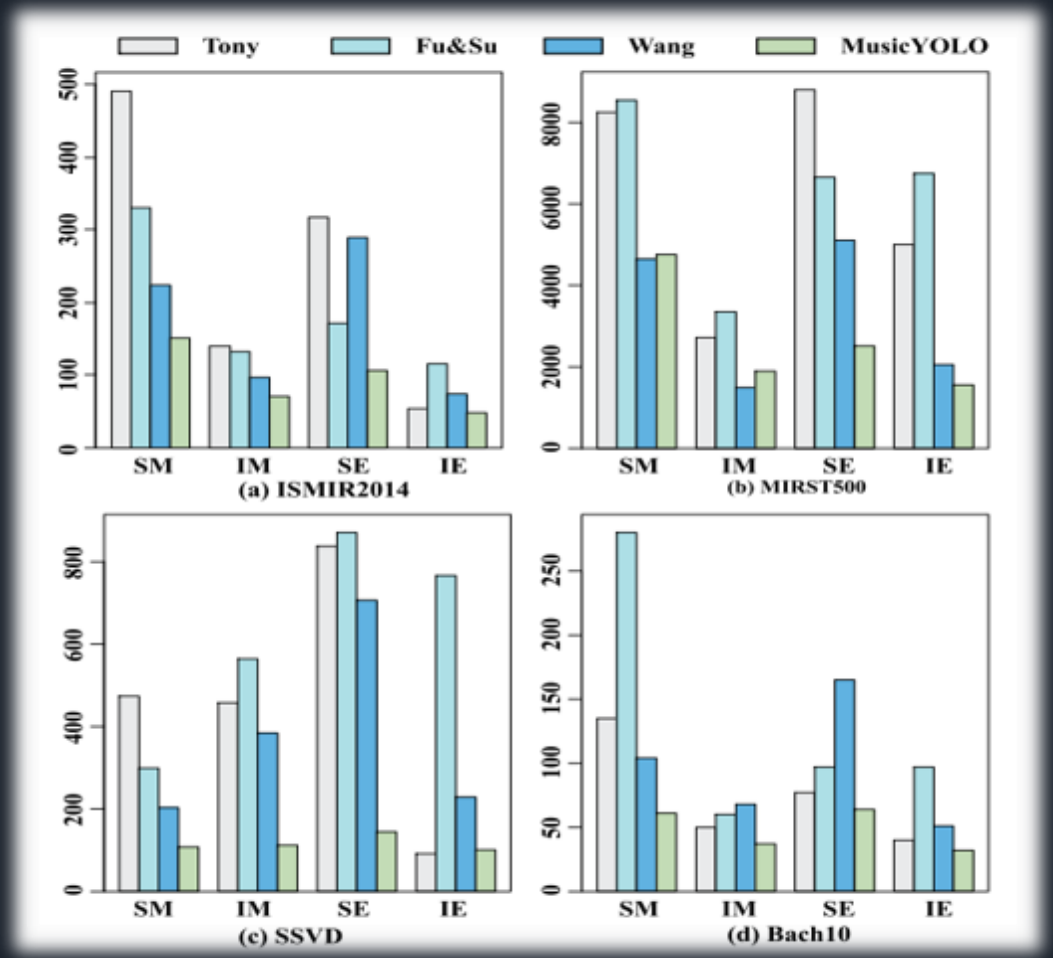
- Extra detection errors in the middle of long notes

Isolated Extras (IE)

- Caused by pitch fluctuations

Note Detection Error Analysis





Transcription Results

TRANSCRIPTION RESULTS OF DIFFERENT MODELS

		ISMIR2014			MST500			SSVD			Bach10		
		P	R	F	P	R	F	P	R	F	P	R	F
Note	Tony	65.69	58.60	61.70	38.92	43.29	40.78	71.54	71.44	71.41	91.07	87.74	89.31
	Fu&Su	77.81	70.62	73.80	36.46	38.55	37.33	63.08	64.07	63.37	83.93	76.48	79.92
	Wang I	66.79	68.15	67.33	66.09	70.88	68.18	58.24	62.25	60.00	82.12	83.65	82.55
	MusicYOLO I	85.24	78.84	81.78	74.69	68.61	**71.42**	79.19	78.15	78.64	89.27	81.72	85.20
	MusicYOLO II	84.00	82.44	**83.12**	45.16	47.08	45.94	85.77	86.10	**85.92**	91.77	91.99	**91.86**
Note w/ offset	Tony	49.99	44.87	47.10	24.98	27.98	26.27	67.12	67.16	67.07	84.65	81.77	83.13
	Fu&Su	62.52	56.87	59.40	22.69	24.03	23.25	56.96	57.60	57.17	77.79	70.96	74.12
	Wang I	53.09	54.09	53.49	46.61	50.04	48.12	44.36	47.24	45.64	73.77	75.04	74.19
	MusicYOLO I	72.21	66.90	69.34	61.22	56.27	**58.56**	68.62	67.77	68.17	87.28	79.91	83.30
	MusicYOLO II	72.18	70.96	**71.49**	31.11	32.54	31.70	84.45	84.78	**84.60**	89.28	89.49	**89.37**

Transcription Error Analysis

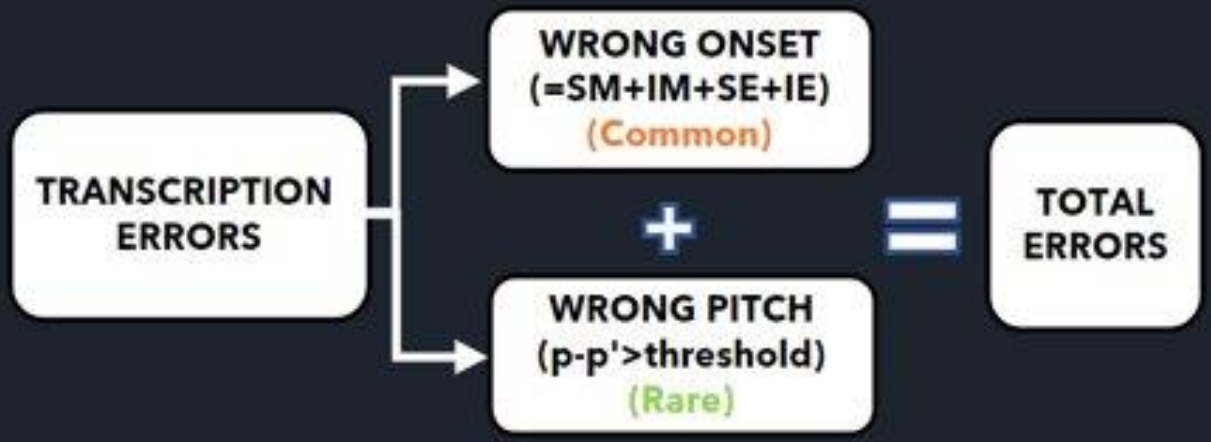
- We classify the transcription errors into two categories-

Wrong Onset (WO)

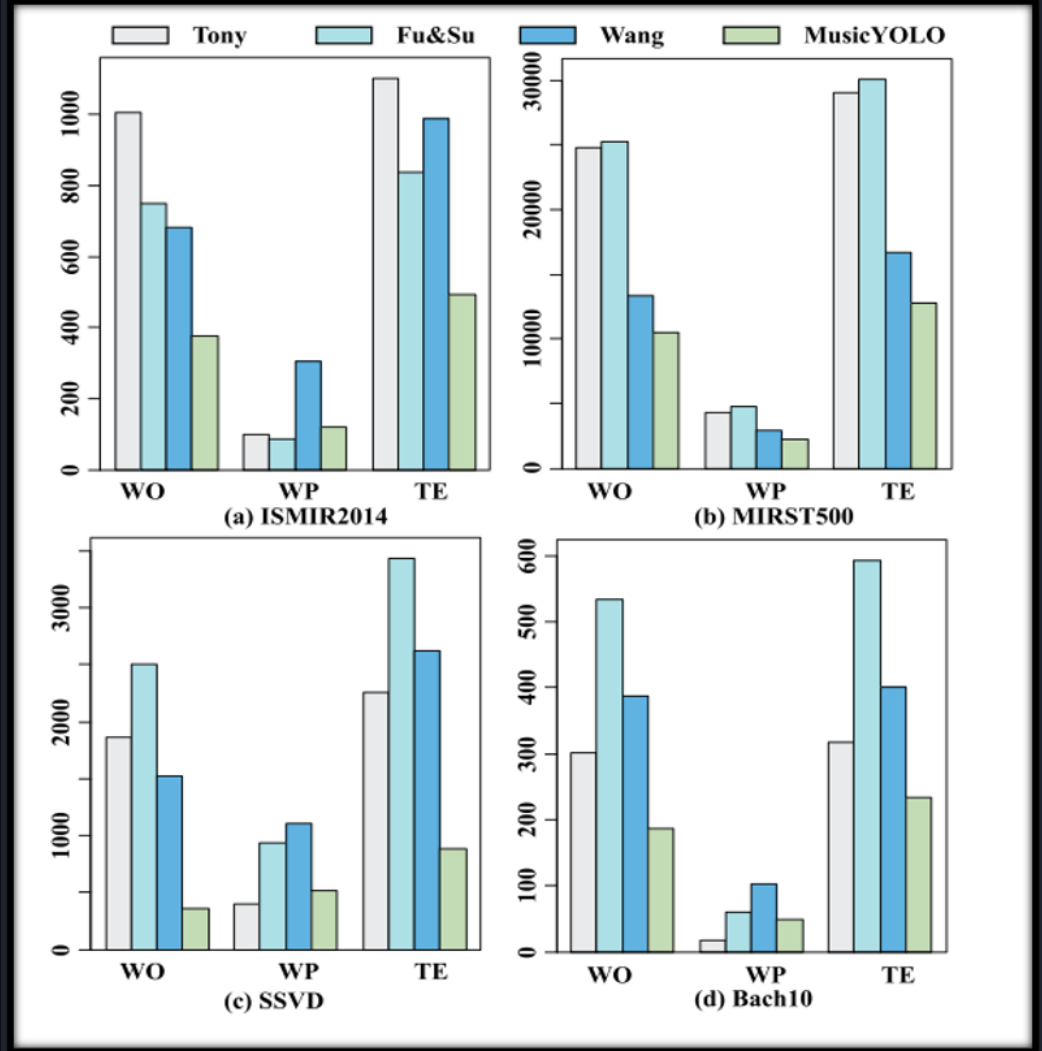
- Caused by onset error

Wrong Pitch (WP)

- When the onset is correct, but the difference between the predicted pitch and annotated pitch exceeds the pitch threshold



Transcription Error Analysis



- The figure shows that the onset detection errors are the main contributor to transcription error.
- Meanwhile pitch errors are rarely found.
- This indicates that extracting pitch in singing transcription is easy
- Onset/offset detection is the key to transcription performance.

THANK YOU