



**Final Project**

**Automatic music  
transcription :  
Polyphonic Piano  
Music Transcription**

**Group 40**

陳劭傑 0756078

王昶鈞 0756612

# Reference

[1] Sigtia, Siddharth, Emmanouil Benetos, and Simon Dixon. "An end-to-end neural network for polyphonic piano music transcription." IEEE/ACM Transactions on Audio, Speech, and Language Processing 24.5 (2016): 927-939.

[2] Boulanger-Lewandowski, Nicolas, Yoshua Bengio, and Pascal Vincent. "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription." arXiv preprint arXiv:1206.6392 (2012).

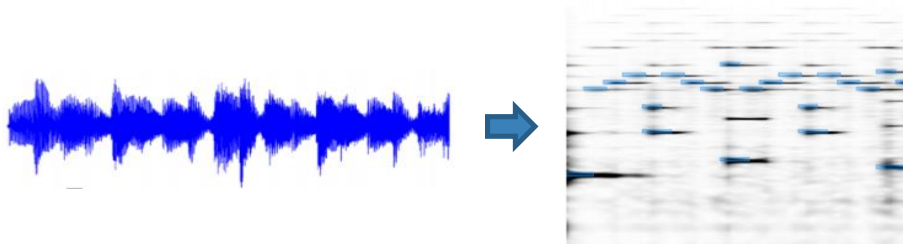
[3] Boulanger-Lewandowski, Nicolas, Yoshua Bengio, and Pascal Vincent. "High-dimensional sequence transduction." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.

# Introduction

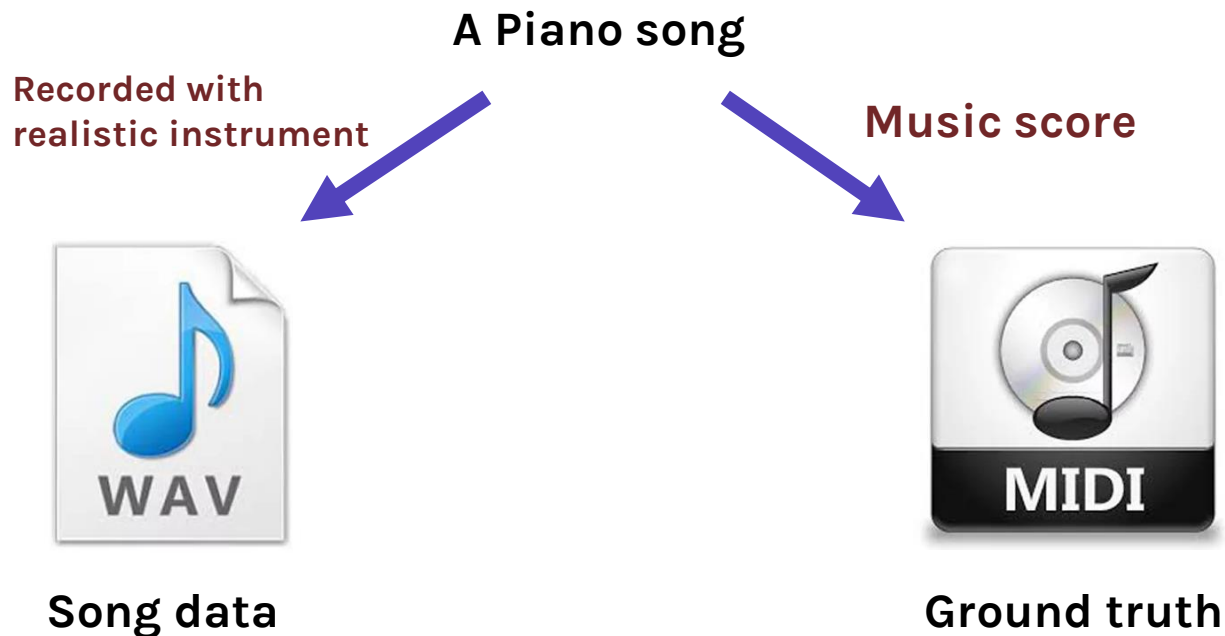
- **Automatic Music Transcription (AMT)** automates the process of converting an acoustic musical signal into some form of musical notation.
- The AMT problem can be divided into several subtasks: multi-pitch detection, note onset/offset detection, quantization ... etc.

# Goals and Objectives

- The core problem in automatic transcription is the estimation of concurrent pitches in a time frame, also called multi-pitch detection
- We implemented a Hybrid-RNN proposed by [1].



# Methodology - Data Representation



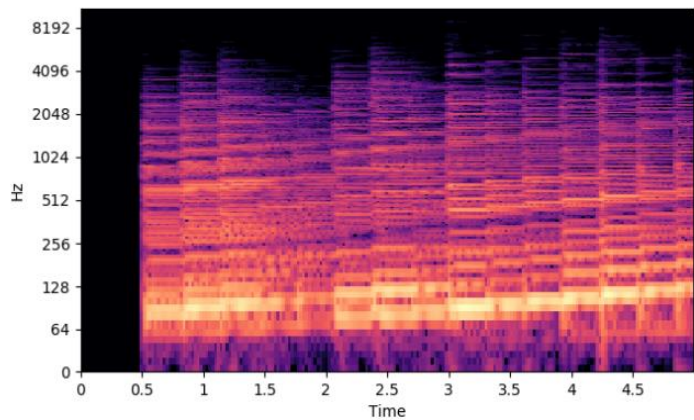
# Methodology - Datasets

- [MAESTRO](#)
  - 1184 piano songs with .wav and .midi files.
  - Training set : 900 songs
  - Testing set : 284 songs

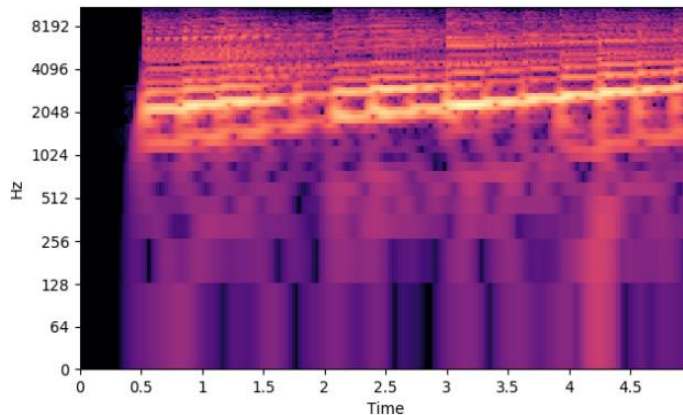
	maestro-v1.0.0
類型:	檔案資料夾 (.0)
位置:	D:\
大小:	102 GB (110,369,383,226 位元組)
磁碟大小:	102 GB (110,374,211,584 位元組)

# Methodology - Data Preprocessing (1/2)

- Convert (wav.) file to Constant-Q transform(CQT) spectrogram, which has stretches the low frequency domain comparing with Short-Time Fourier Transform(STFT).



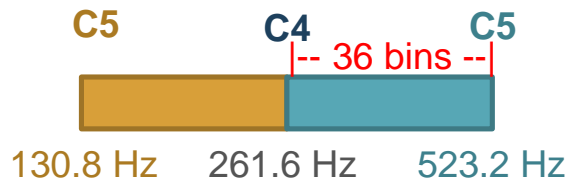
STFT




CQT

## Methodology - Data Preprocessing (2/2)

- To enhance the efficiency of CQT, the audio signal is first down sampled from **44.1 kHz**  $\rightarrow$  **16 kHz**.
- The hop size is **512**, so we have  $1/(16000 / 512) = 0.032$  second per frame.
- **88** notes  $\times$  **3** bins = **264** dimensional input vector for each frame.



	processedMaestro
類型:	檔案資料夾
位置:	D:\
大小:	287 GB (308,166,029,133 位元組)
磁碟大小:	287 GB (308,170,878,976 位元組)



## Methodology – Hybrid structure

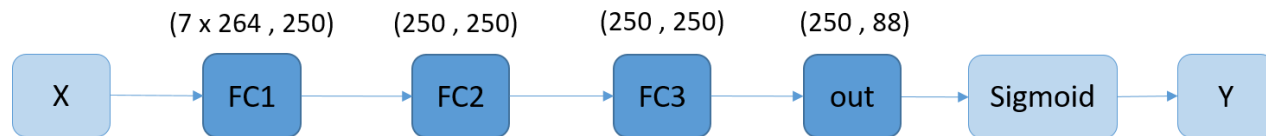


Acoustic model

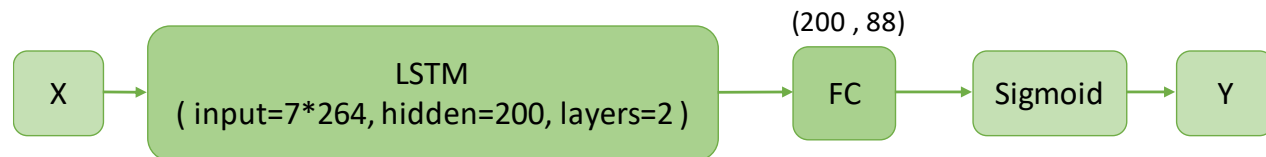
Language model

# Methodology – Acoustic models

- All models use **Binary Cross Entropy** as their loss function
- DNN

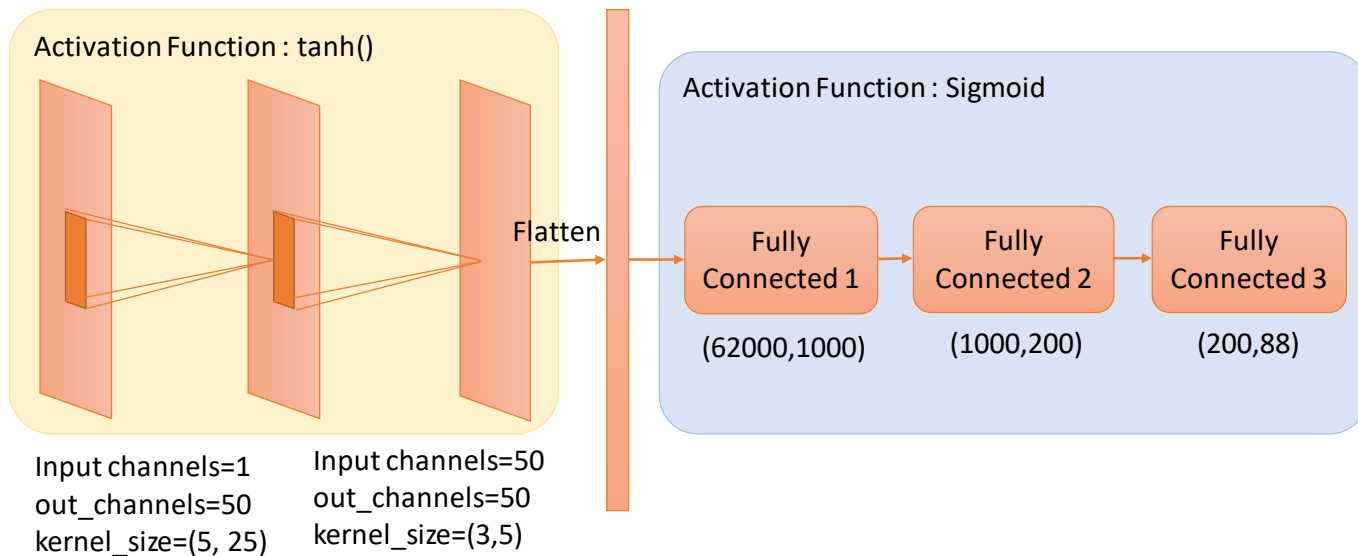


- LSTM

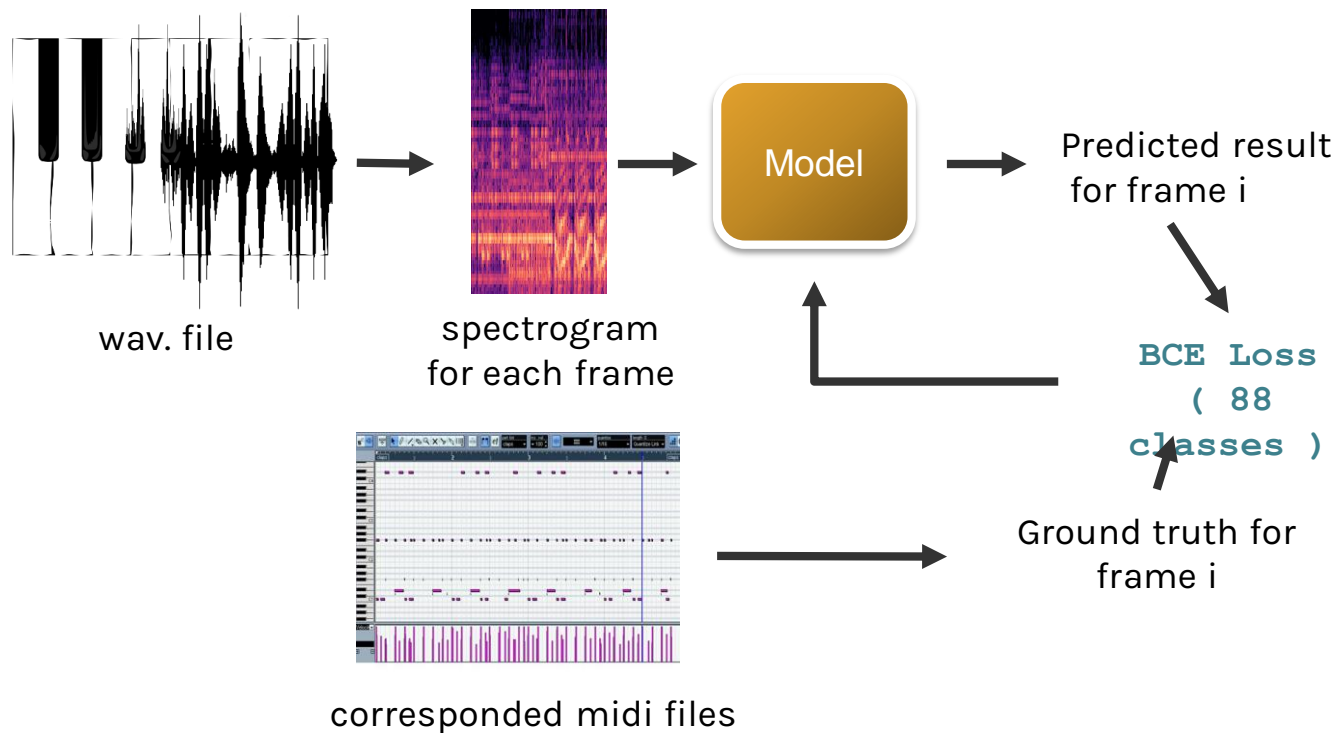


# Methodology – Acoustic models

- CNN



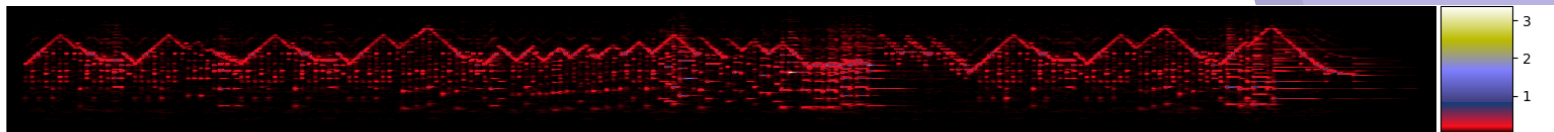
# Methodology - Training



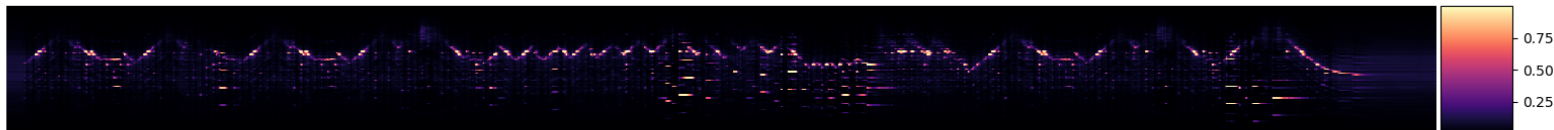
# Acoustic model outputs

2017\MIDI-Unprocessed\_070\_PIANO070\_MID--AUDIO-split\_07-08-17\_Piano-e\_1-02\_wav--2

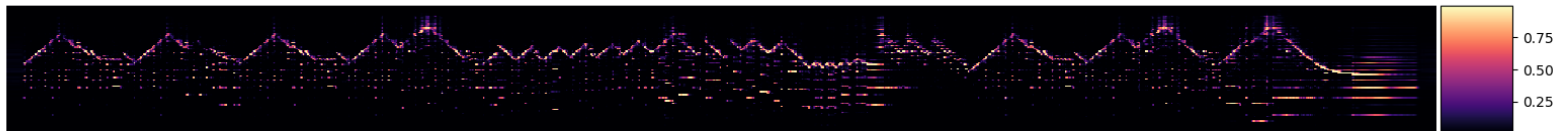
Wav



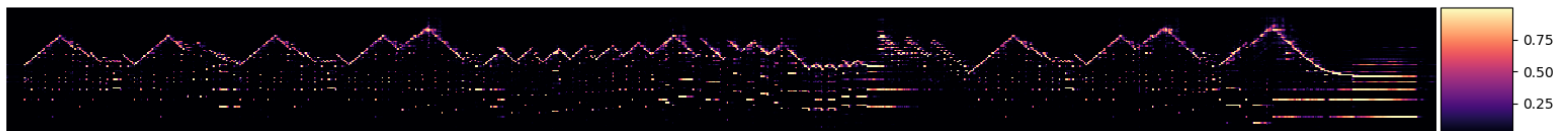
DNN



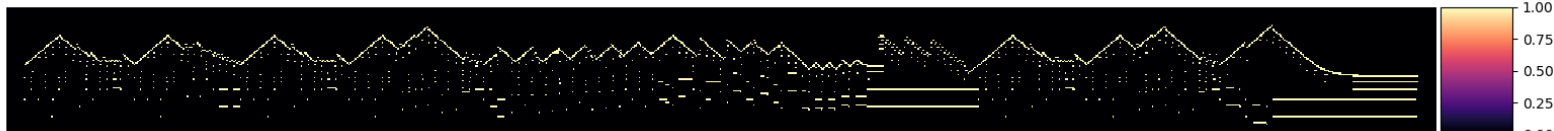
RNN



CNN



Truth



# Methodology – Music Language Model

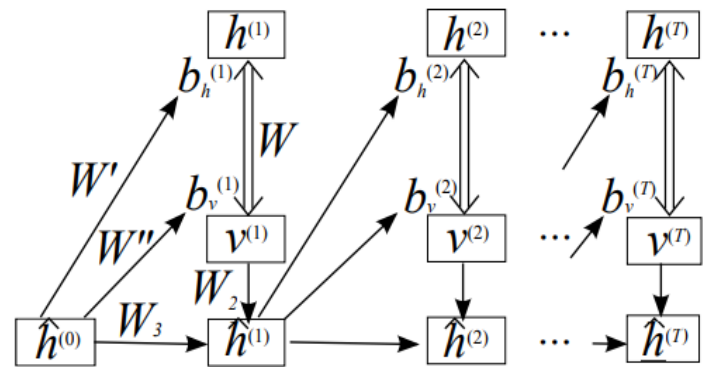
- Given a sequence  $y = y_0^t$ , we use the **MLM** to define a prior probability distribution  $P(y)$ ,  $y_t$  is a high-dimensional binary vector that represents the notes being played at  $t$
- In order to learn high dimensional, temporal distributions for the MLM, we used the **RNN-RBM** as proposed in [2]
- We compute the **RBM biases** as a linear transformation of the **RNN hidden**

$$b_h^t = b_h + W' \hat{h}_t$$

$$b_v^t = b_v + W'' \hat{h}_t$$

# Methodology – RNN-RBM

1. Propagate the current values of the hidden units  $\hat{h}^{(t)}$  in the RNN portion of the graph using (11),
2. Calculate the RBM parameters that depend on the  $\hat{h}^{(t)}$  (eq. 8 and 9) and generate the negative particles  $v^{(t)*}$  using  $k$ -step block Gibbs sampling,
3. Use  $CD_k$  to estimate the log-likelihood gradient (eq. 6) with respect to  $W$ ,  $b_v^{(t)}$  and  $b_h^{(t)}$ ,
4. Propagate the estimated gradient with respect to  $b_v^{(t)}$ ,  $b_h^{(t)}$  backward through time (BPTT) (Rumelhart et al., 1986) to obtain the estimated gradient with respect to the RNN parameters.



# Methodology – BPTT For RNN-RBM

In order to capture better temporal dependencies, we initialize the  $W_2, W_3, b_{\hat{h}}, W'', b_v, \hat{h}^{(0)}$  parameters of the RNN-RBM from an RNN trained with the cross-entropy cost:

$$L(\{v^{(t)}\}) = \frac{1}{T} \sum_{t=1}^T \sum_{j=1}^{n_v} -v_j^{(t)} \log y_j^{(t)} - (1-v_j^{(t)}) \log(1-y_j^{(t)})$$

where  $y^{(t)} = \sigma(b_v^{(t)})$



## Methodology – BPTT For RNN-RBM

Suppose we want to minimize the negative log-likelihood cost  $C \equiv -\log P(\{v^{(t)}\})$ . The gradient of  $C$  with respect to the parameters of the conditional RBMs can be estimated by CD using equations

$$\frac{\partial C}{\partial b_v^{(t)}} \simeq v^{(t)*} - v^{(t)}$$

# Methodology – BPTT For RNN-RBM

$$\frac{\partial C}{\partial W} \simeq \sum_{t=1}^T \sigma(Wv^{(t)*} - b_h^{(t)})v^{(t)*T} - \sigma(Wv^{(t)} - b_h^{(t)})v^{(t)T} \quad (14)$$

$$\frac{\partial C}{\partial b_h^{(t)}} \simeq \sigma(Wv^{(t)*} - b_h^{(t)}) - \sigma(Wv^{(t)} - b_h^{(t)}). \quad (15)$$

The gradient then back-propagates through the hidden-to-bias parameters (eq. 8 and 9):

$$\frac{\partial C}{\partial W'} = \sum_{t=1}^T \frac{\partial C}{\partial b_h^{(t)}} \hat{h}^{(t-1)T} \quad (16)$$

$$\frac{\partial C}{\partial W''} = \sum_{t=1}^T \frac{\partial C}{\partial b_v^{(t)}} \hat{h}^{(t-1)T} \quad (17)$$

$$\frac{\partial C}{\partial b_h} = \sum_{t=1}^T \frac{\partial C}{\partial b_h^{(t)}} \text{ and } \frac{\partial C}{\partial b_v} = \sum_{t=1}^T \frac{\partial C}{\partial b_v^{(t)}}. \quad (18)$$

For the single-layer RNN-RBM, the BPTT recurrence relation follows from (11):

$$\begin{aligned} \frac{\partial C}{\partial \hat{h}^{(t)}} &= W_3 \frac{\partial C}{\partial \hat{h}^{(t+1)}} \hat{h}^{(t+1)} (1 - \hat{h}^{(t+1)}) \\ &\quad + W' \frac{\partial C}{\partial b_h^{(t+1)}} + W'' \frac{\partial C}{\partial b_v^{(t+1)}} \end{aligned} \quad (19)$$

for  $0 \leq t < T$  ( $\hat{h}^{(0)}$  being a parameter of the model) and  $\partial C / \partial \hat{h}^{(T)} = 0$ . Formulas for the remaining RNN-RBM parameters are:

$$\frac{\partial C}{\partial b_h} = \sum_{t=1}^T \frac{\partial C}{\partial \hat{h}^{(t)}} \hat{h}^{(t)} (1 - \hat{h}^{(t)}) \quad (20)$$

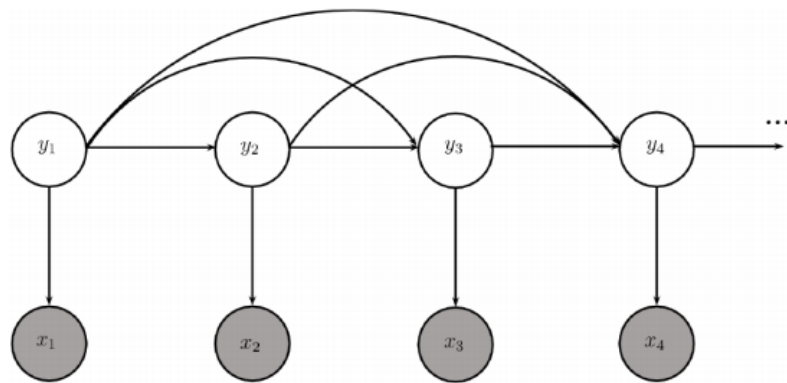
$$\frac{\partial C}{\partial W_3} = \sum_{t=1}^T \frac{\partial C}{\partial \hat{h}^{(t)}} \hat{h}^{(t)} (1 - \hat{h}^{(t)}) \hat{h}^{(t-1)T} \quad (21)$$

$$\frac{\partial C}{\partial W_2} = \sum_{t=1}^T \frac{\partial C}{\partial \hat{h}^{(t)}} \hat{h}^{(t)} (1 - \hat{h}^{(t)}) v^{(t)T}. \quad (22)$$

# Inference

- The hybrid RNN is a graphical model that combines the predictions of any *arbitrary* frame level acoustic model with an **RNN-based language** model :
- Let  $x = x_0^T$  be a sequence of be the corresponding transcriptions. The joint probability of  $y = y_0^T, y, x$ , can be factorized as follows:

$$\begin{aligned} P(y, x) &= P(y_0 \dots y_T, x_0 \dots x_T) \\ &= P(y_0)P(x_0|y_0) \prod_{t=1}^T P(y_t|y_0^{t-1})P(x_t|y_t). \end{aligned}$$



Graphical Model of the Hybrid Architecture.

# Inference

- Using Bayes's rule, the conditional distribution can be written as follows:

$$P(y|x) \propto P(y_0|x_0) \prod_{t=1}^T P(y_t|y_0^{t-1})P(y_t|x_t),$$

- The priors  $P(y_t|y_0^{t-1})$  are obtained from the **RNN-RBM MLM**, while the posterior distributions  $P(x_t|y_t)$  are obtained from the acoustic models

# Inference

- At test time, we would like to find the mode of the conditional output distribution:

$$y^* = \operatorname{argmax}_y P(y|x)$$

- We are interested in solutions that globally optimize  $P(y|x)$ . But exhaustively searching for the best sequence is intractable since the number of possible configurations of  $y^t$  is exponential in the number of output pitches ( $2^n$  for  $n$  pitches).

# Beam search

- A graph search algorithm that is commonly used to decode the conditional outputs of an RNN.
- When the space of candidate solutions is **large**, the algorithm can be constrained to consider only  **$K$  new candidates** for each partial solution in the beam.
- In [3], the authors, propose forming a pool of  $K$  candidates by **drawing random samples** from the conditional output distribution.

# Beam search

Find the most likely sequence  $y$  given  $x$  with a beam width  $w$  and branching factor  $K$ .

$beam \leftarrow$  new beam object

$beam.insert(0, \{\})$

**for**  $t = 1$  to  $T$  **do**

$new\_beam \leftarrow$  new beam object  
        (min-priority queue of capacity  $w * K$ )

**while**  $new\_beam.len() < w$  **do**

**for**  $l, s, m_a, m_l$  **in**  $beam$  **do**

**for**  $k = 1$  to  $K$  **do**

$y' = m_a.next\_most\_probable()$

$l' = \log P_l(y'|s)P_a(y'|x_t) - \log P(y')$

$m'_l \leftarrow m_l$  with  $y_t := y'$

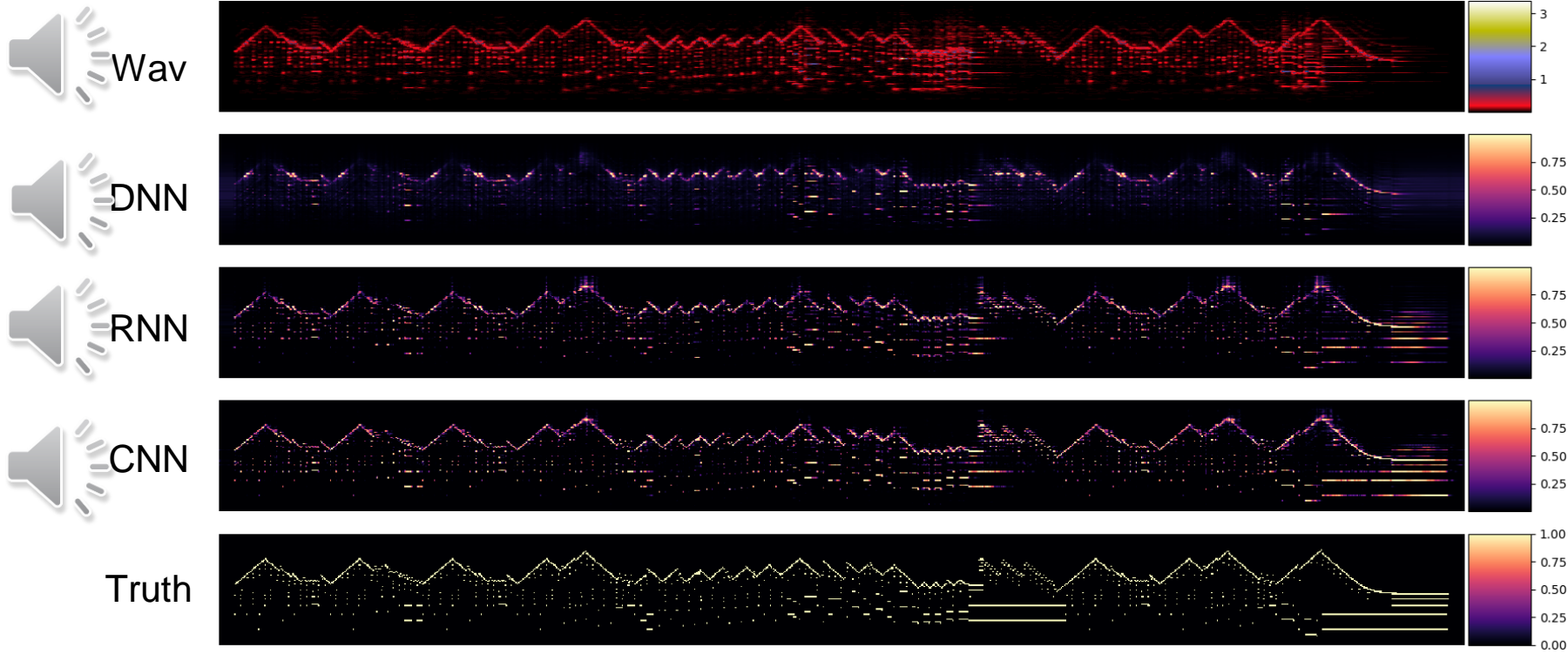
$m'_a \leftarrow m_a$  with  $x := x_{t+1}$

$new\_beam.insert(l + l', \{s, y'\}, m_a, m_l)$

$beam \leftarrow new\_beam$

# Result

2017\MIDI-Unprocessed\_070\_PIANO070\_MID--AUDIO-split\_07-08-17\_Piano-e\_1-02\_wav--2





# Result – Other songs not in testing set

- V.K. - 1 Billion Lightyear of Distance



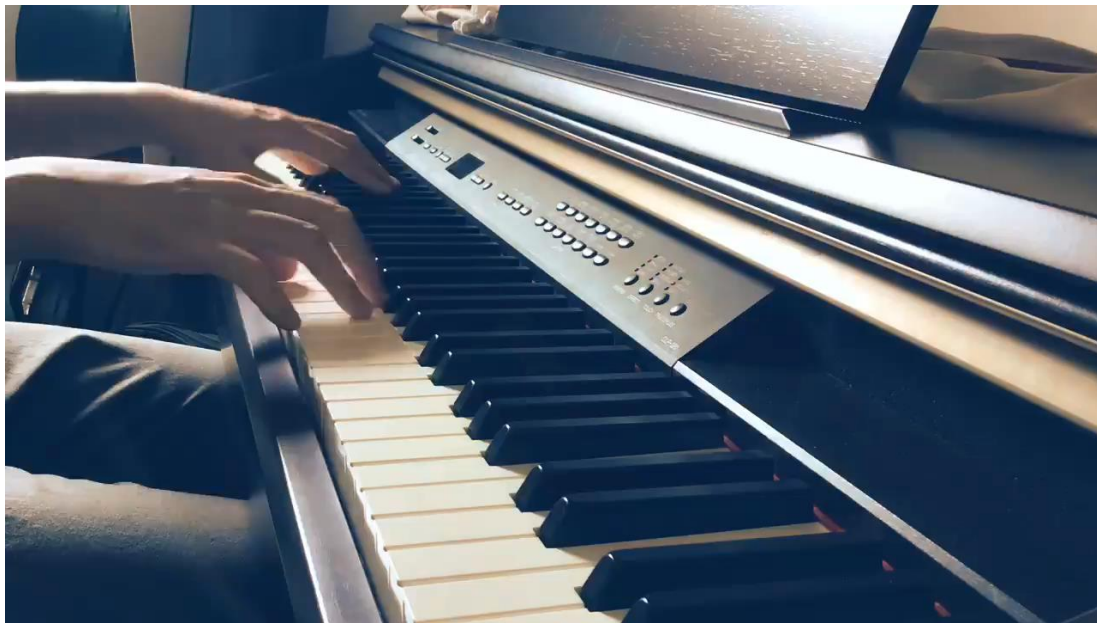
- Kreisler-Rachmaninoff: Liebesleid



- Marasy8 - Senbonzakura Piano ver.



## Result – Recording video



Transcription result:



# Result – Metrics for acoustic models

- **F-1 measure** for acoustic model(Threshold = 0.5) DNN
  - DNN: 0.2548693
  - RNN: 0.6302122
  - **CNN: 0.7372564**

		PREDICTED	
		Positive	Negative
ACTUAL	Positive	True Positive TP	False Negative FN
	Negative	False Positive FP	True Negative TN

believe

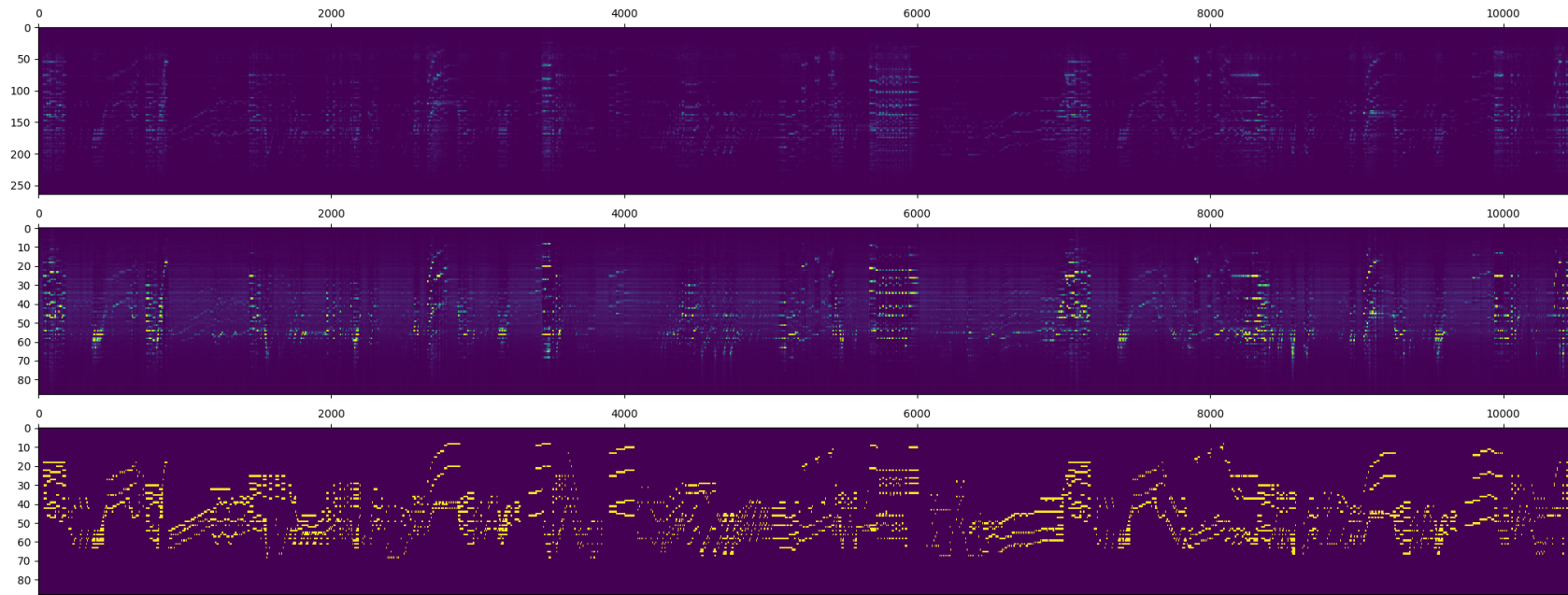
$$\mathcal{P} = \sum_{t=1}^T \frac{TP[t]}{TP[t] + FP[t]}$$

$$\mathcal{R} = \sum_{t=1}^T \frac{TP[t]}{TP[t] + FN[t]}$$

$$\mathcal{A} = \sum_{t=1}^T \frac{TP[t]}{TP[t] + FP[t] + FN[t]}$$

$$\mathcal{F} = \frac{2 * \mathcal{P} * \mathcal{R}}{\mathcal{P} + \mathcal{R}}$$

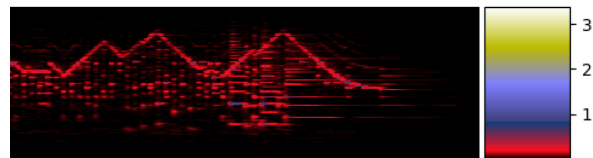
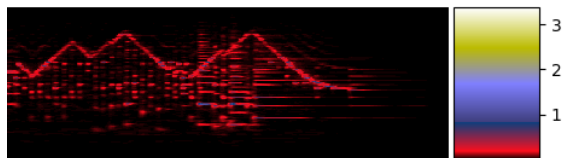
# Result – DNN Failed



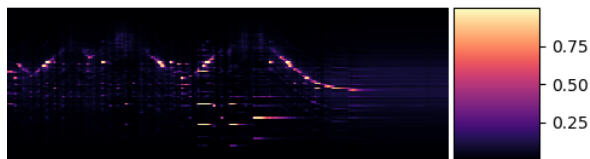
Model output

Using threshold value 0.5

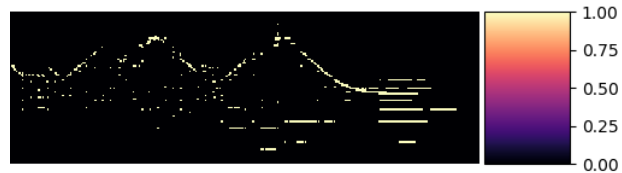
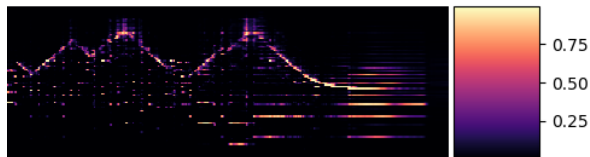
Wav



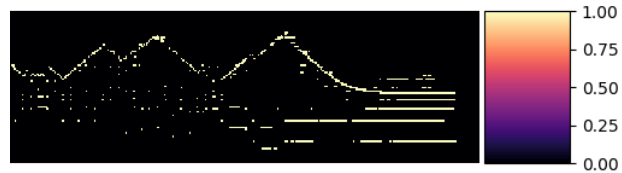
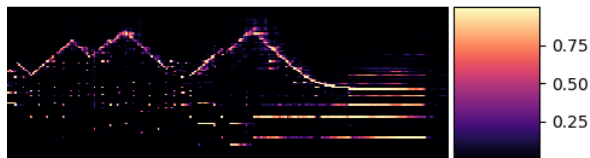
DNN



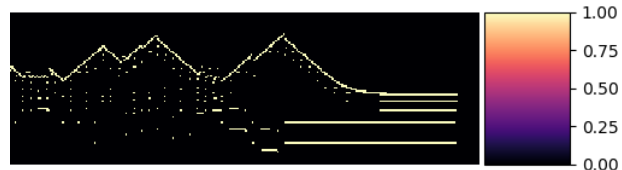
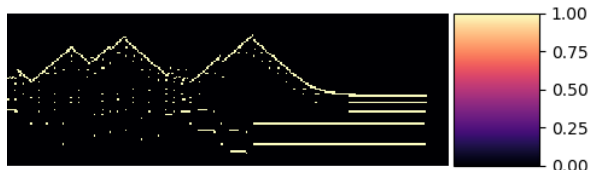
RNN



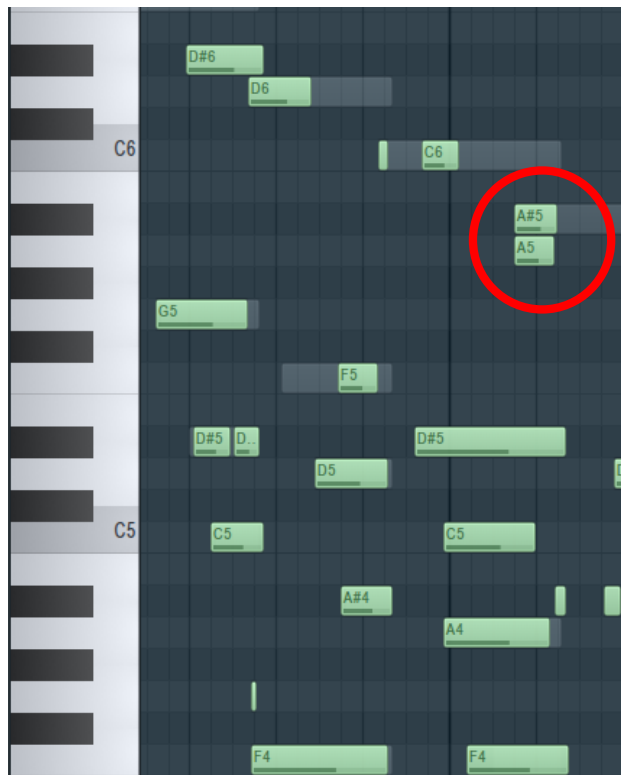
CNN



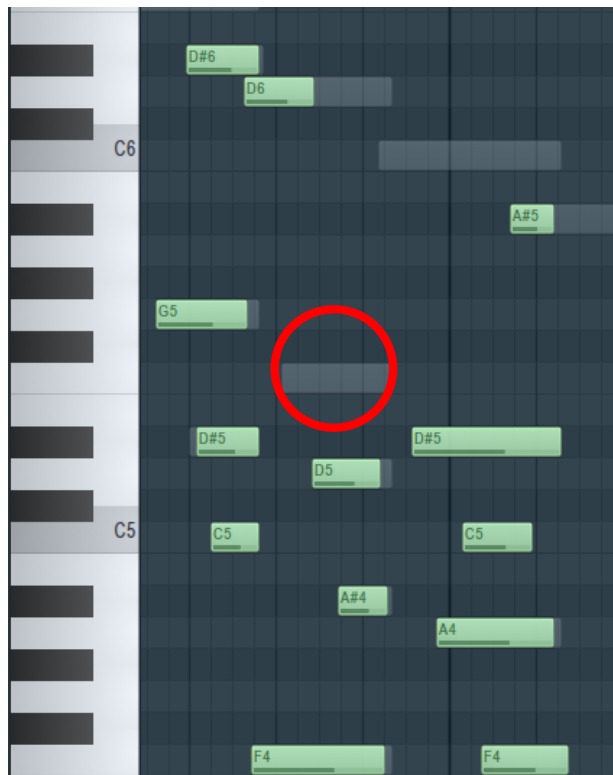
Truth



## Acoustic Model outputs



## Hybrid-structure



# Conclusion

- Combined with acoustic model and music language model, we can get good results in music transcription, and even apply to piano songs that are not in the test set, such as from albums or recordings.
- However, our network will be affected by the audio noise and the recording environment for bad estimating. The network needs to collect more information and other optimization techniques to increase the capability.



**Thanks for listening**