



# MUSIC SCORE IMAGE GUITAR PRO (GPX) CONVERTER

2018038015 홍석균

2018038037 이정렬

2019038054 김경민



COURSE: ARTIFICIAL INTELLIGENCE

Chungbuk National University

# CONTENTS

01. Motivation

---

02. Research Object

---

03. Material (Dataset)

---

04. Model

---

05. Mid-Term Discussion

---

06. Improve Performance

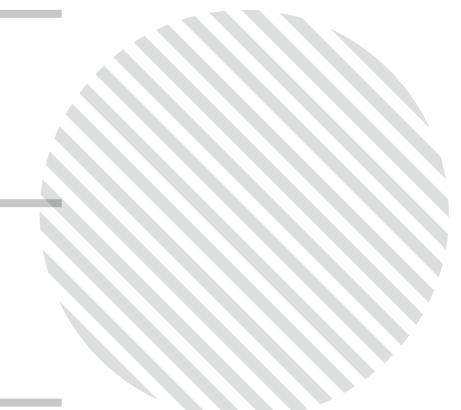
---

07. Discussion

---

08. Result

---



# MOTIVATION

Pno.

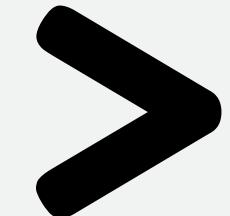
2  
15

19

21

To Coda

The image shows three staves of piano sheet music. The top staff starts at measure 15, featuring eighth-note chords in common time. The middle staff starts at measure 19, showing sixteenth-note patterns. The bottom staff starts at measure 21, with sixteenth-note patterns and a dynamic instruction 'To Coda'.

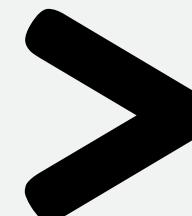
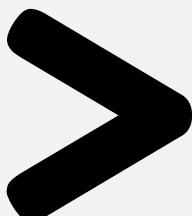
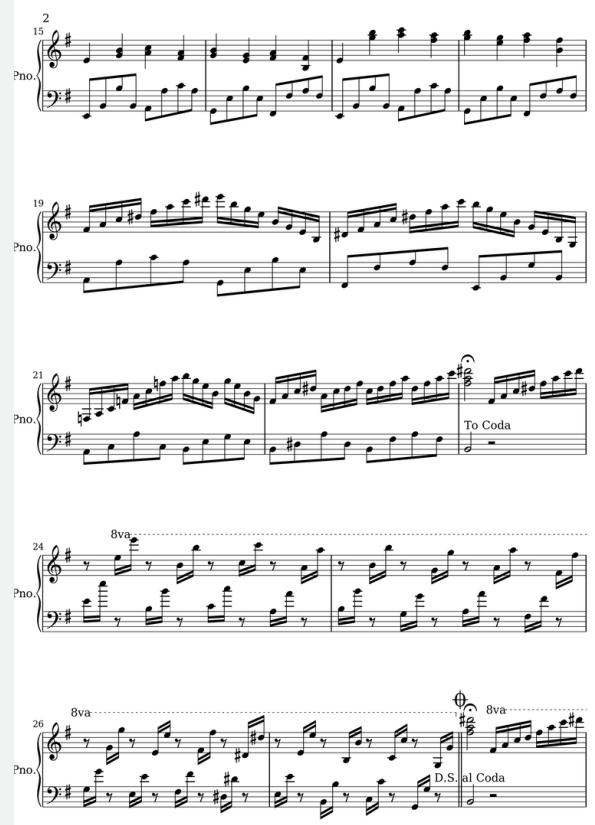


11 12 9 11 12 9

The image shows a screenshot of a guitar pro program. It displays the piano sheet music from the previous image, now converted into a guitar tablature format. The tablature shows six strings and includes numerical fret numbers (11, 12, 9) and a 'To Coda' instruction. The software's toolbar and menu bar are visible at the top.

Identifying music symbols and entering them into a guitar pro program  
→ Music Scores Image Guitar Pro converter

# RESEARCH OBJECT



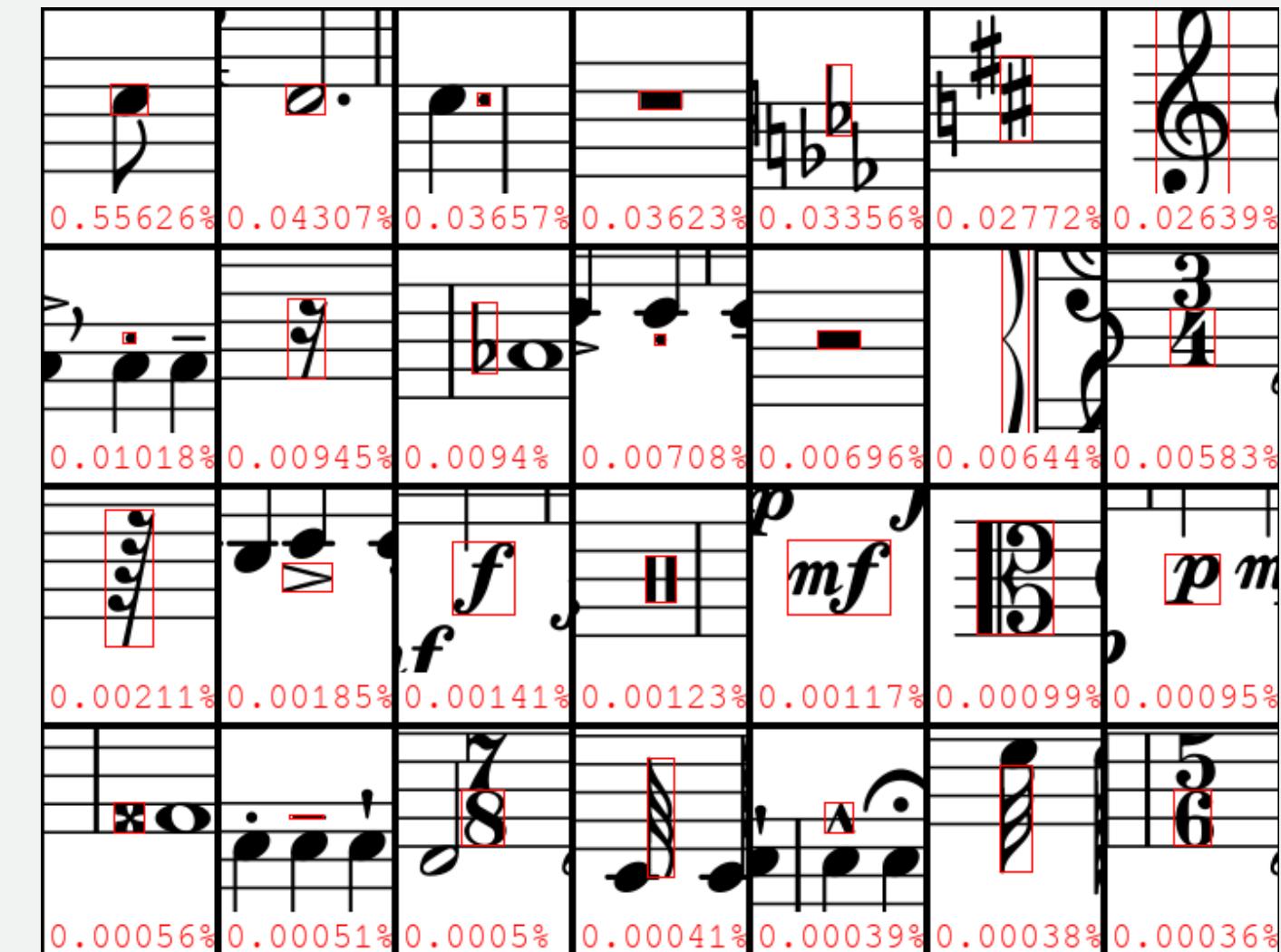
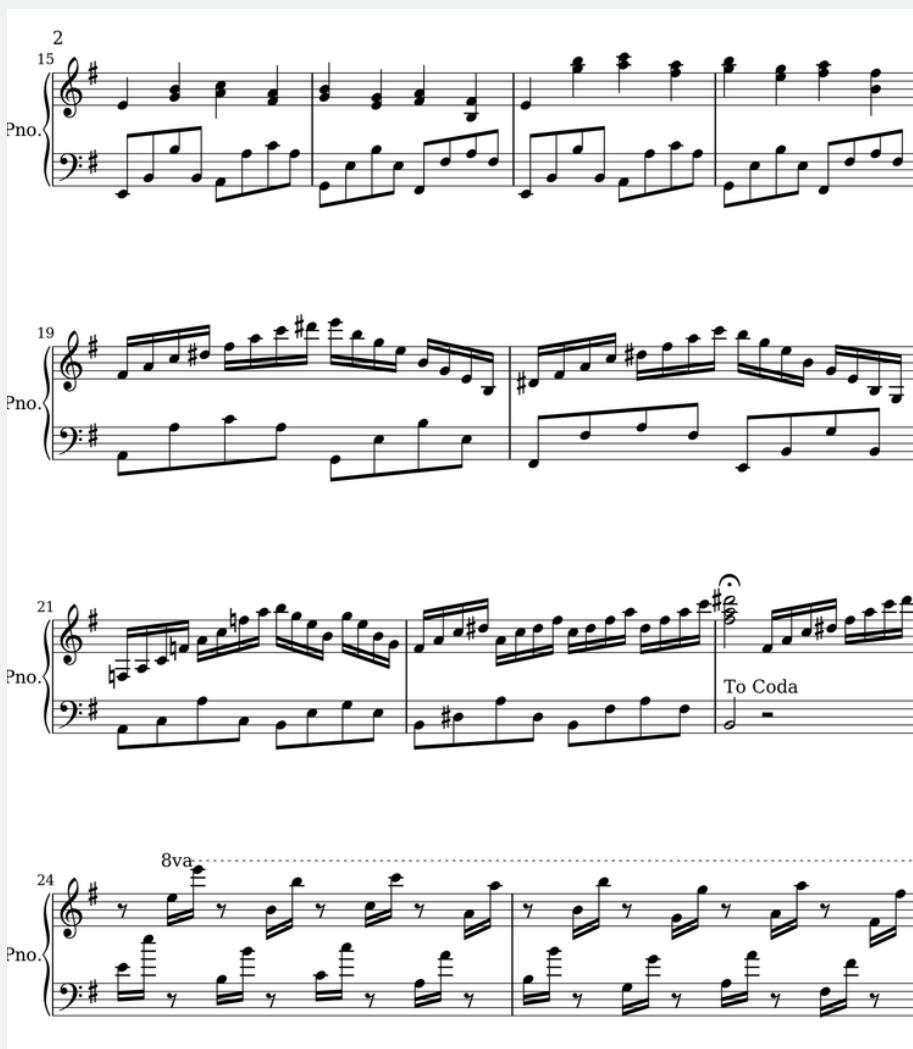
Music Scores Image

Object Detection

Create Guitar Pro file (.gpx)

# MATERIAL (DATASET)

## What is DeepScoresV2?



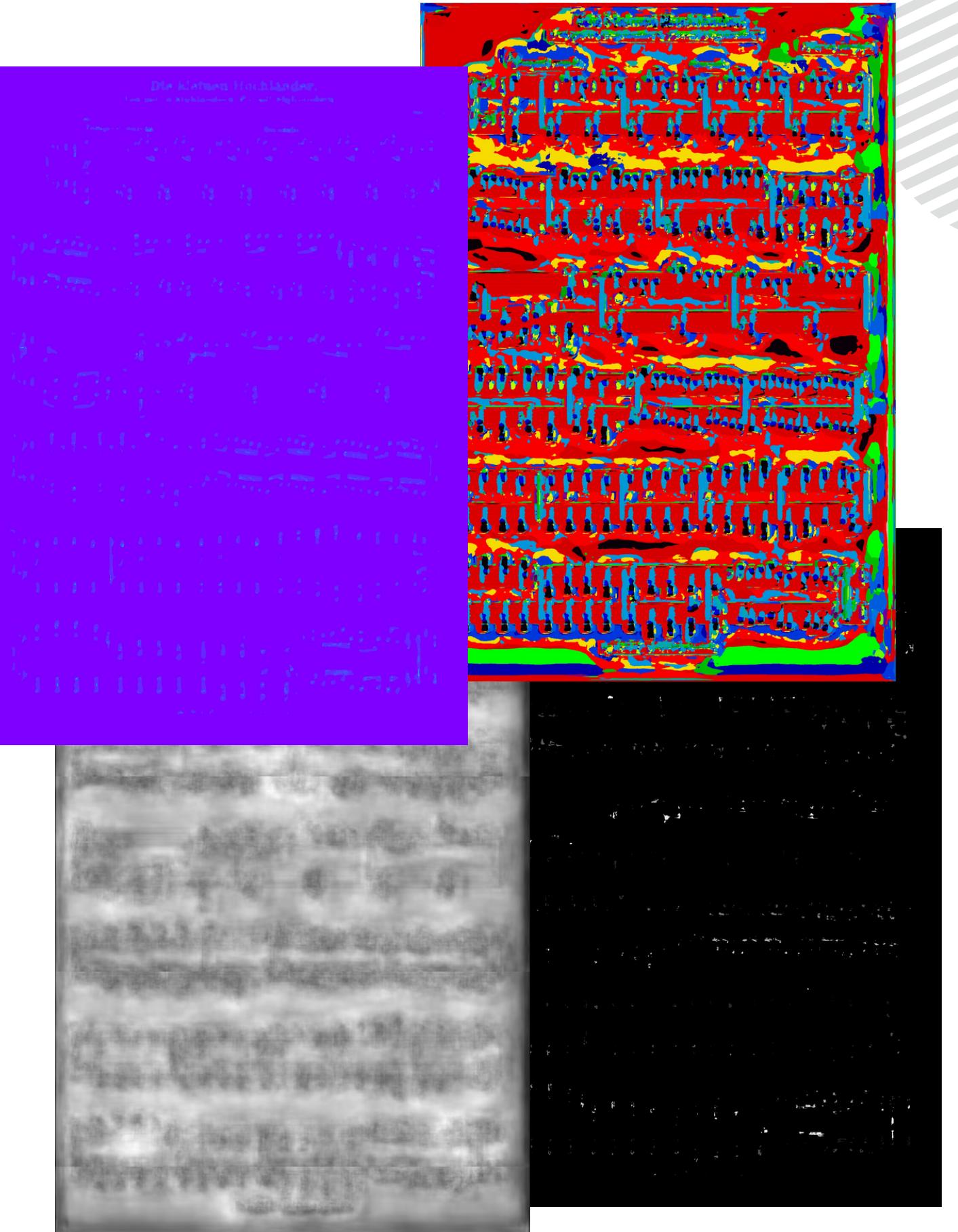
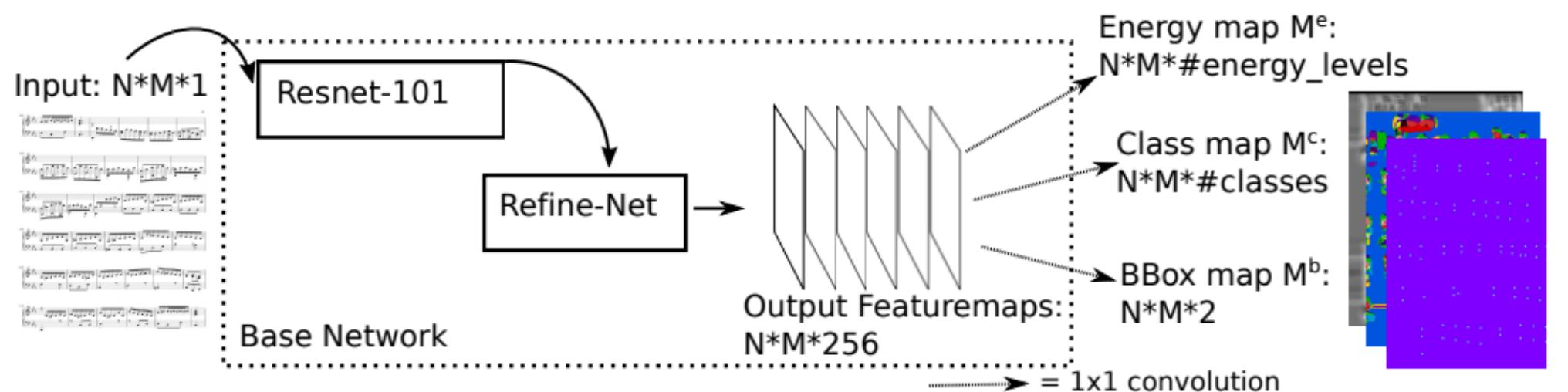
Music score image dataset Symbols of various shapes and sizes in Coco data format  
(clef, key signature, note, rest, beat, scale) included

# MODEL

## DEEP WATERSHED DETECTOR

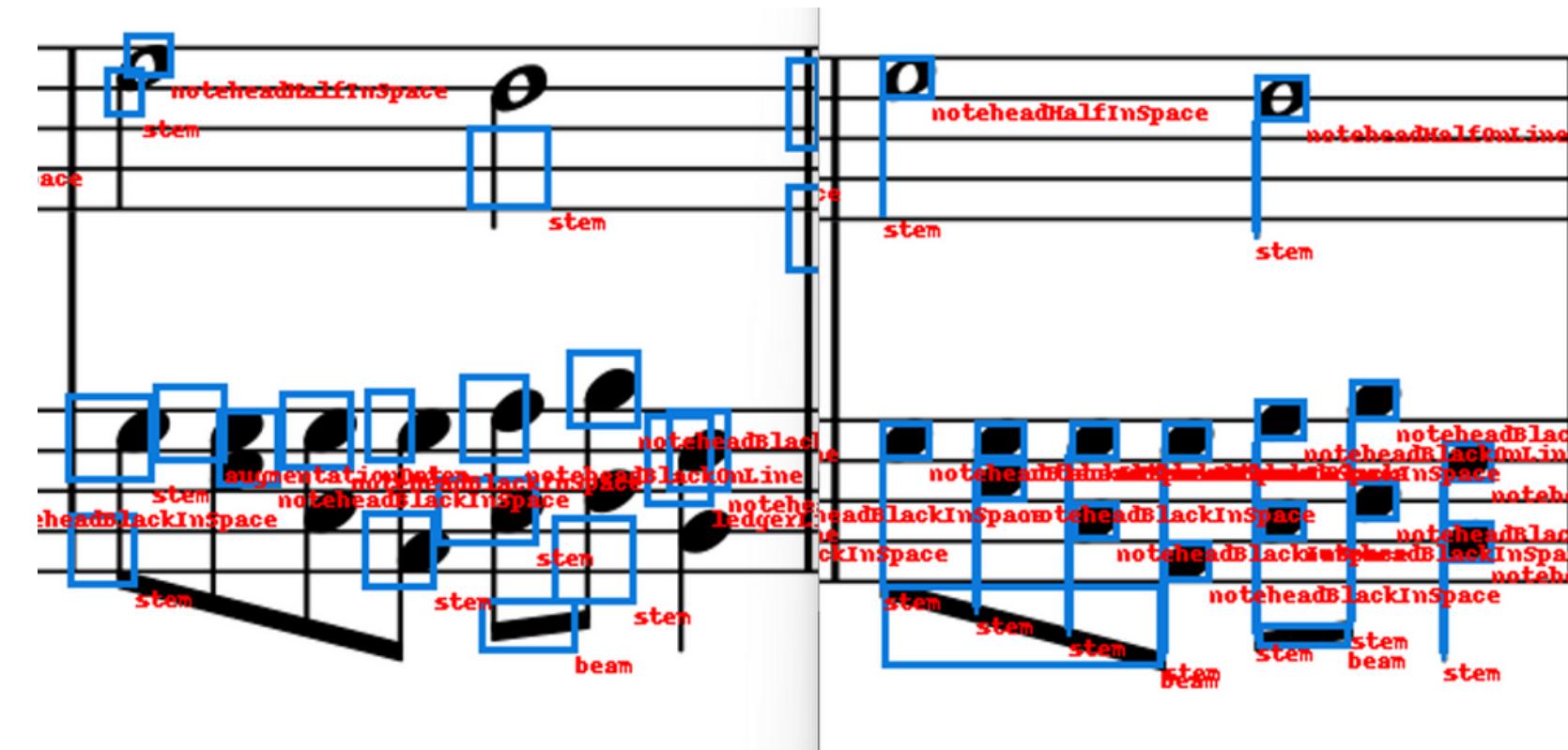
Watershed algorithm, one of the image segmentation algorithms, combined with deep learning network

Deep learning network **trains music symbols by DeepScoresV2 dataset** → Apply watershed transformation to **perform segmentation**



# MID-TERM DISCUSSION

# IMPROVED PERFORMANCE



# **prediction**

## ground truth

# Incorrect Bounding Box

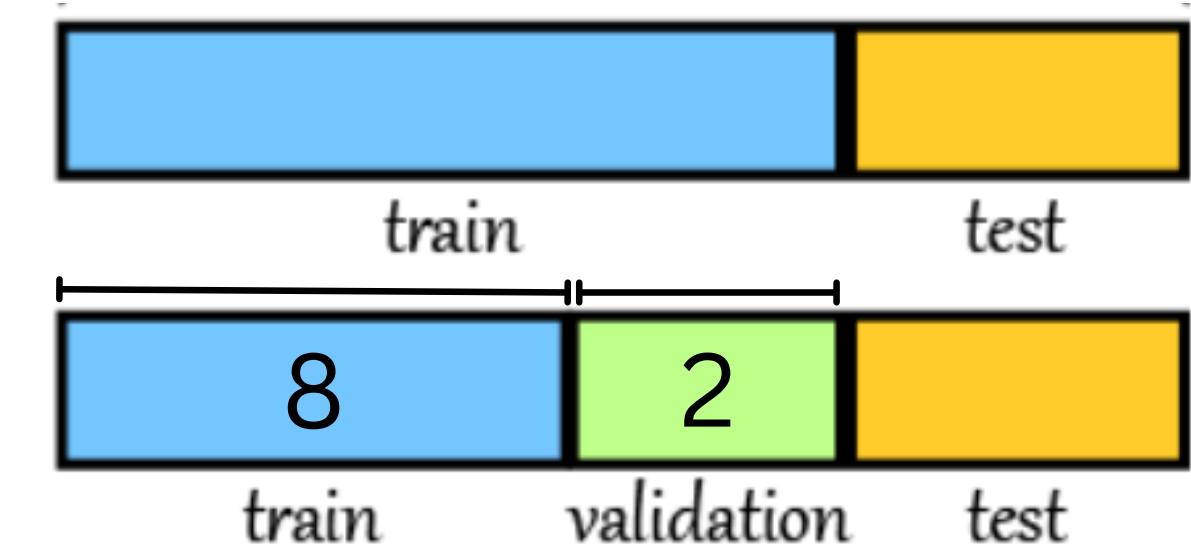
# IMPROVE PERFORMANCE

## HYPERPARAMETER TUNING

### Hyperparameters

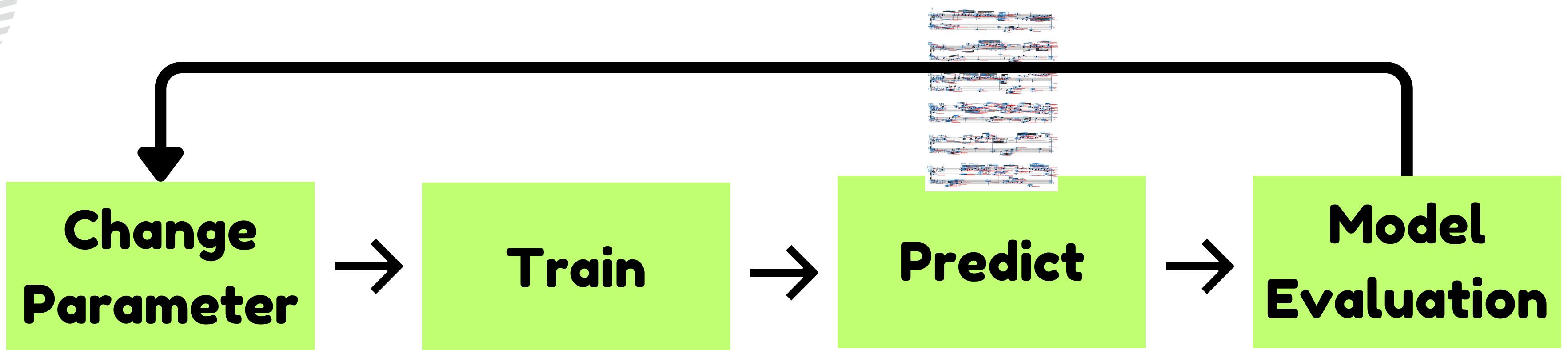
1. model : Resnet-101
2. optimizer : RMSProp
3. learning rate :  $1e-4 = 0.0001$
4. iteration : 2200

### Dataset



※ Red letters are default parameter values

# PROCESS



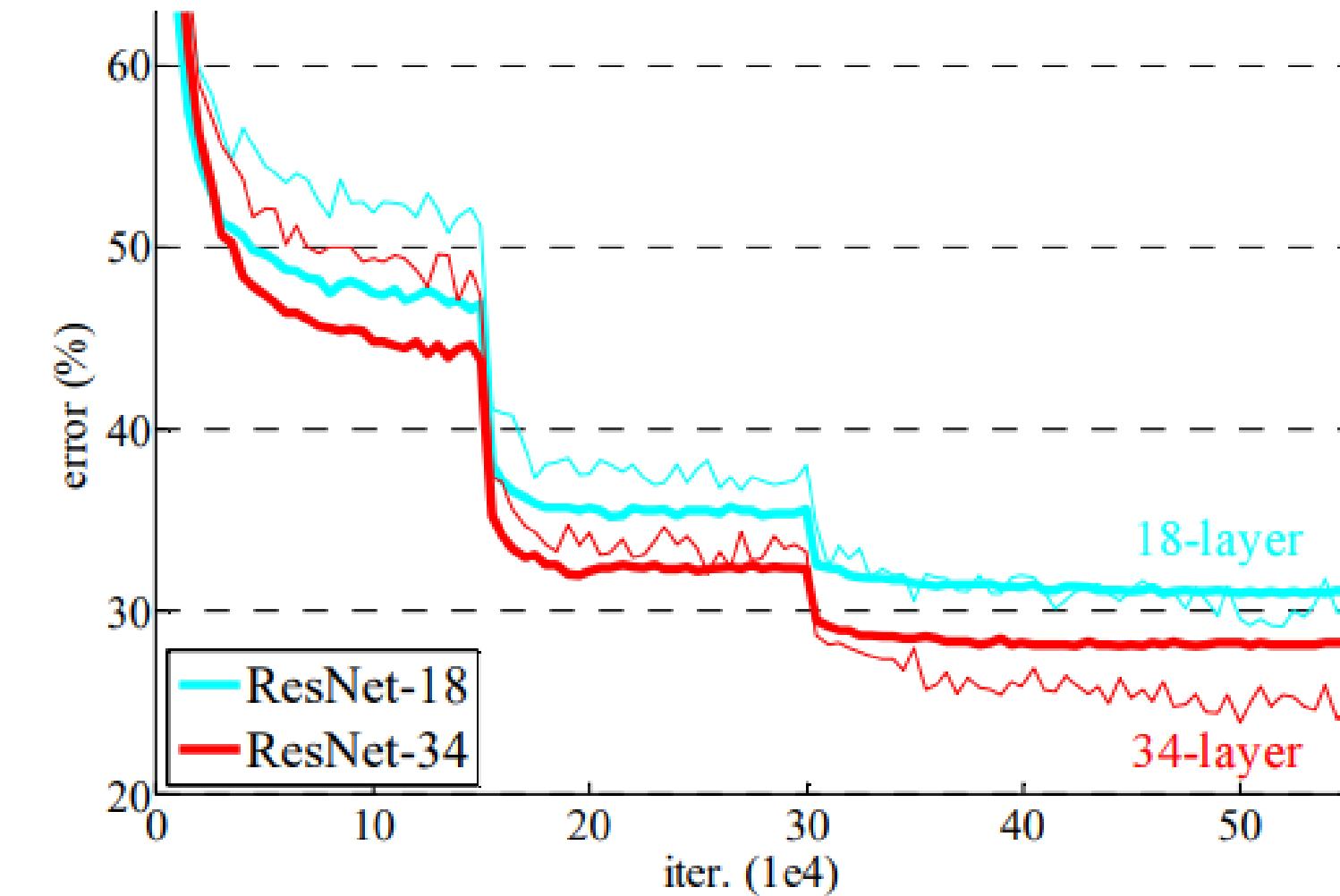
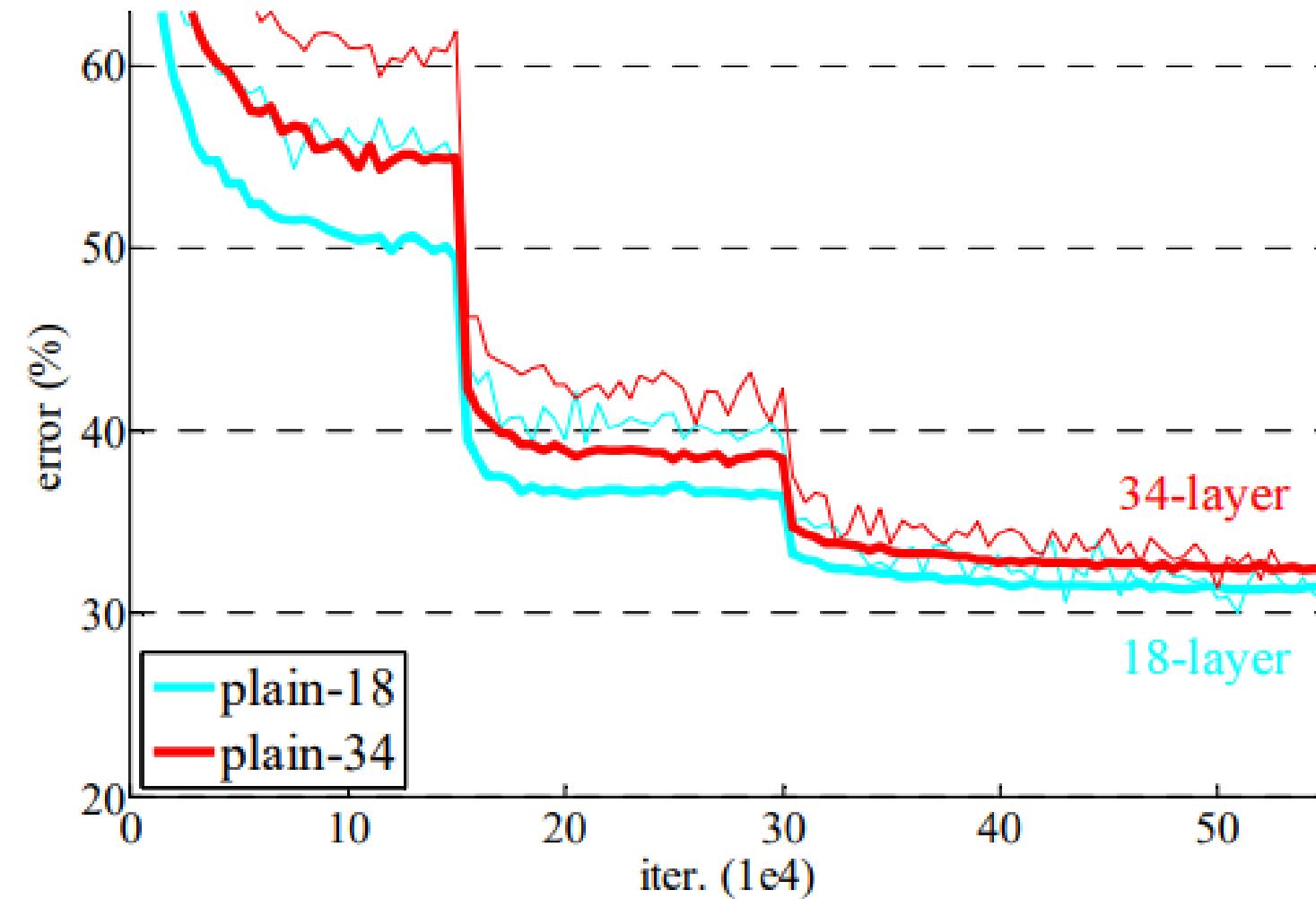
- model
- optimizer
- learning rate
- iteration

Draw  
bounding box

- AP at 0.5
- F1-Score

# 1. CHANGE MODEL

Why Change to ResNet-152?



In Resnet, **the deeper** the network layer, **the lower** the error rate.

# 1. CHANGE MODEL

Resnet-101 → Resnet-152

	Current Experiment	Mid-Term Experiment
f1-score	2.853e-5	8.229e-6
precision	1.085e-4	2.800e-05
recall	1.642e-05	4.823e-06

Current experiment shows better performance then Mid-Term Experiment

## 2. CHANGE LEARNING RATE

Why Change Learning Rate?



Data deviates randomly  
and may cause overshooting

Very long learning time,  
may not reach the lowest point

## 2. CHANGE LEARNING RATE

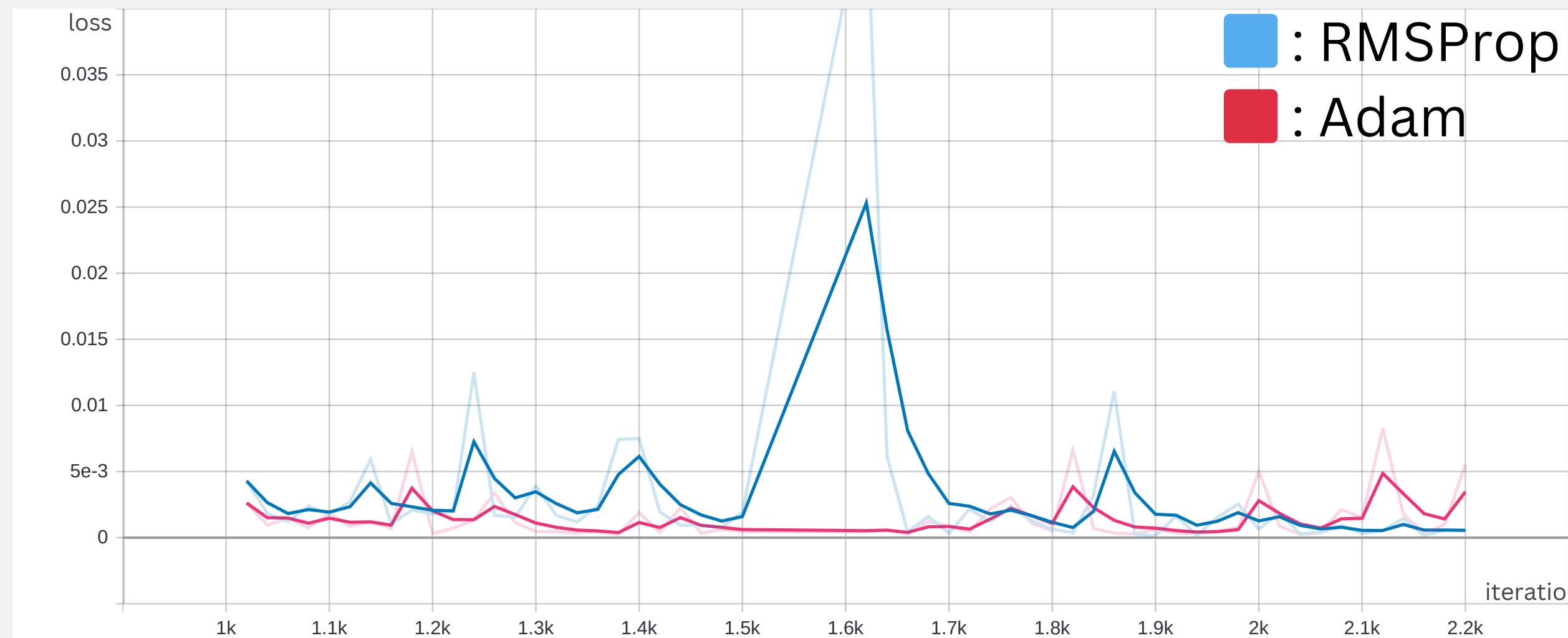
Learning Rate  $1e-4 \rightarrow 1e-2$

	Current Experiment	Experiment with Res-152
f1-score	X	2.853e-5
precision	X	1.085e-4
recall	X	1.642e-5

Learning rate  $1e-2$  caused overshooting.  
As a result of learning by finding the optimal learning rate,  $1e-4$  was optimal.

# 3. CHANGE OPTIMIZER

Why Change the Optimizer?



Optimizer is used to minimizes loss functions.

We tried ‘Adam’ Optimizer that is most frequently used for deep learning.

### 3. CHANGE OPTIMIZER

RMSProp → Adam

	Current Experiment	Experiment with RMSProp
f1-score	7.3972e-6	2.853e-5
precision	2.06e-4	1.085e-4
recall	3.7662e-6	1.642e-5

As a result of training, 'Adam' performs worse than 'RMSProp'.

# 4. CHANGE ITERATION

2200 → 7500

	Current Experiment	Experiment with 2200 itr
f1-score	7.0927e-7	2.853e-5
precision	3.0183e-6	1.085e-4
recall	4.0185e-7	1.642e-5

As a result of training, We expected the performance to improve when we increased iteration to 7500, but actual result was not.

# DISCUSSION

## Why is there little difference?

Nr	class	No. Occurrences	AP at 0.5
52	fermataAbove	105	0.001
13	timeSig3	103	0.001
35	flag64thDown	28	0.001
27	stem	25194	0
38	accidentalNatural	706	0
45	articStaccatoAbove	654	0
37	accidentalFlat	499	0
39	accidentalSharp	430	0
25	noteheadWholeInSpace	391	0
43	articAccentAbove	270	0

F1-score fell due to the presence of undetected classes.  
We decided to **compare the final result as the AP** for each class.

# RESULT

## Optimal Hyperparameters

1. model : Resnet-152
2. optimizer : RMSProp
3. learning rate : 1e-4
4. iteration : 2200

※ Red letters are default parameter values

# RESULT

Nr	class	No. Occurrences	AP at 0.5
2	clefF	534	0.004
5	noteheadHalfOnLine	957	0.002
3	noteheadBlackOnLine	13845	0.002
4	noteheadBlackInSpace	13635	0.001
21	restQuarter	736	0.001
24	dynamicP	289	0.001
22	rest8th	818	0.001
6	noteheadHalfInSpace	1045	0.001
20	restWhole	753	0.001
26	dynamicF	622	0.001



clefF

Nr	class	No. Occurrences	AP at 0.5
73	ornamentMordent	29	0.144
7	clefF	534	0.112
2	segno	25	0.017
72	ornamentTurnInverted	23	0.017
90	dynamicCrescendoHairpin	108	0.015
5	clefCAlto	123	0.008
64	dynamicP	289	0.008
24	noteheadWholeOnLine	389	0.007
71	ornamentTurn	9	0.007
14	timeSig4	413	0.005

mid-term prediction

final prediction

# RESULT

# mid-term prediction



# final prediction



# ground truth

**THANK YOU**

