

## 1.해야할일들

- 음정, 셈여림 시간별 특징 추출 시 이상한점 리스팅
- 셈여림 특징 추출 시 빈 시간대 데이터 채우기 (반드시 악보에 나와있는 셈여림표를 참고해서 정답 데이터의 셈여림 수정하기. 악보의 노트나 음이 발생한 시간대를 보면 정답데이터의 셈여림을 수정할 수 있을듯함)
- 악보의 셈여림에 맞추어 다시 연주 (cubase 구매 후 악보 수정 필요)
- 0.1초에는 데이터가 있고 0.2초에는 데이터가 없고 이런식으로 반복되는 데 왜 그런건지와 비교 알고리즘 적용시에 이부분때문에 정확도가 떨어지지 않는지 확인 필요

음정, 셈여림 시간별 특징 추출 시 이상 한점 리스팅

# 음성, 셈여림 시간별 특징 추출 시 이상 한점 리스팅

1	sec	tick	bpm	msg_type	channel	note	velocity	dynamic
2	0	0	69.000017	[]	[]	[]	[]	ppp
3	0.001	1	69.000017	['note_on']	[0]	[60]	[61]	p
4	0.003	2	69.000017	['note_on']	[0]	[60]	[61]	p
5	0.005	3	69.000017	['note_on']	[0]	[60]	[61]	p
6	0.007	4	69.000017	['note_on']	[0]	[60]	[61]	p
7	0.009	5	69.000017	['note_on']	[0]	[60]	[61]	p
8	0.01	6	69.000017	['note_on']	[0]	[60]	[61]	p
9	0.012	7	69.000017	['note_on']	[0]	[60]	[61]	p
10	0.014	8	69.000017	['note_on']	[0]	[60]	[61]	p

데이터 추출 알고리즘 V4 파일로 추출한 결과 -> 1틱 기준으로 데이터 추출

	A	B	C	D	E	F
1	sec	msg_type	channel	note	velocity	dynamic
2	0	['note_on']	[0]	[60]	[61]	p
3	0.1	[]	[]	[]	[]	
4	0.2	['note_on']	[0]	[64]	[57]	p
5	0.3	[]	[]	[]	[]	
6	0.4	['note_on']	[0]	[67]	[56]	p
7	0.5	[]	[]	[]	[]	
8	0.6	['note_off',	[0, 0]	[67, 72]	[64, 60]	mp
9	0.7	[]	[]	[]	[]	
10	0.8	['note_off',	[0, 0]	[72, 73]	[64, 60]	mp

데이터 추출 알고리즘 V5 파일로 추출한 결과 -> 0.1초 간격으로 데이터 추출

## 문제점

- V4로 추출한 데이터에 있던 연속적인 셈여림 기호가, V5로 0.1초 간격으로 뽑았을 때 셈여림 데이터 자체가 손실됨.
  - note\_on 또는 note\_off가 발생했을 때만 이벤트가 발생해서, 이벤트가 발생한 경우에 대해서만 데이터가 추출되는 것으로 확인.

# 음성, 샘플링 시간별 특징 추출 시 이상 한점 리스팅

	A	B	C	D	E	F
1	sec	msg_type	channel	note	velocity	dynamic
2	0	['note_on']	[0]	[60]	[61]	p
3	0.1	[]	[]	[]	[]	
4	0.2	['note_on']	[0]	[64]	[57]	p
5	0.3	[]	[]	[]	[]	
6	0.4	['note_on']	[0]	[67]	[56]	p
7	0.5	[]	[]	[]	[]	
8	0.6	['note_off', 'note_on']	[0, 0]	[67, 72]	[64, 60]	mp
9	0.7	[]	[]	[]	[]	
10	0.8	['note_off', 'note_on']	[0, 0]	[72, 76]	[64, 63]	mp
11	0.9	[]	[]	[]	[]	
12	1	['note_off', 'note_on']	[0, 0]	[76, 67]	[64, 50]	p
13	1.1	[]	[]	[]	[]	
14	1.2	[]	[]	[]	[]	
15	1.3	['note_off', 'note_on']	[0, 0]	[67, 72]	[64, 47]	p
16	1.4	[]	[]	[]	[]	
17	1.5	['note_off', 'note_on']	[0, 0]	[72, 76]	[64, 47]	p
18	1.6	[]	[]	[]	[]	
19	1.7	['note_off', 'note_off', 'note_off', 'note_on']	[0, 0, 0, 0]	[60, 64, 76]	[64, 64, 64]	mp
20	1.8	[]	[]	[]	[]	

## 문제점

1. 피아노 건반이 눌린 상태를 데이터가 반영하지 못함. -> 예를 들면, 그림 Sec 0일 때 note (60)이 note\_on이 되었다면, 0.1초에서도 note\_on인 상태를 반영해야 하는데, 0.1초에는 아무 데이터가 없음.
  - a. 이유: 이벤트가 발생할 때만 note\_on 또는 note\_off를 찍어주는 것으로 확인

# 음성, 썸여림 시간별 특징 추출 시 이상 한점 리스팅

썸여림(dynamic)	정의	평균 velocity 범위
ppp(피아니시시모)	아주 여리게	0~35
pp(피아니시모)	매우 여리게	36~48
p(피아노)	여리게	49~61
mp(메조 피아노)	조금 여리게	62~74
mf(메조 포르테)	조금 세게	75~87
f(포르테)	세게	88~100
ff(포르티시모)	매우 세게	101~113
fff(포르티시시모)	아주 세게	114~127

8	0.6	['note_off', 'note_on']	[0, 0]	[67, 72]	[64, 60]	mp
---	-----	-------------------------	--------	----------	----------	----

## 문제점

1. 위 표에 대하여 데이터 추출 알고리즘 상에 velocity의 값에 따라서 썸여림 기호를 추출해주고 있는데, 행을 보면, note\_off와 note\_on 두 개의 이벤트가 발생하고 있는데, 이에 대한 썸여림 기호는 하나만 추출해주고 있다.
  - a. 이유 : 썸여림에 대한 이해도 부족?
2. 찾아보니 피아노 건반을 누를 때와 건반에서 손을 뗄 때, 각각 note\_on과 note\_off에 대한 이벤트가 발생을 하는데, 이때 발생한 이벤트에 대한 velocity 값이 각각 추출이 된다. 따라서 썸여림에 대한 정의가 필요해 보인다. (피아노 건반을 칠 때에만 썸여림을 반영을 할 것인지? 아니면 둘 다 반영을 할 것인지?)

셈여림 특징 추출 시 빈 시간대 데이터 채우기

# 셈여림 특징 추출 시 빈 시간대 데이터 채우기

앞서 말씀 드렸던 문제점에 대해 다음과 같이 해결했습니다.

## 1. 피아노의 건반 눌린 상태를 유지하기.

a. note\_on 이벤트가 발생한 note에 대해

```
input_file_note_status = [{ 'note': x}, { 'status': False}, { 'velocity' : 0}] for x in range(128)]  
target_file_note_status = [{ 'note': x}, { 'status': False}, { 'velocity' : 0}] for x in range(128)]
```

위 그림과 같이 note의 상태를 만들어 note\_on이 되면, 'status' : True, note\_off면 'status' : False로 0.1초마다 상태를 추출해 주었습니다.

## 2. velocity 계산하기

- 위와 같이 note\_on, note\_off의 상태를 기록해, note\_on이 되어있는 note에 대해 velocity 값을 리스트로 만들어, 그 중 최대값(velocity)에 기반해 셈여림 기호를 추출해 주었습니다.

# 셈여림 특징 추출 시 빈 시간대 데이터 채우기 - 결과

sec	msg_type	channel	note	velocity	dynamic
0	['note_on']	[0]	[60]	[61]	p
0.1	[]	[]	[]	[]	
0.2	['note_on']	[0]	[64]	[57]	p
0.3	[]	[]	[]	[]	
0.4	['note_on']	[0]	[67]	[56]	p
0.5	[]	[]	[]	[]	
0.6	['note_off', 'note_on']	[0, 0]	[67, 72]	[64, 60]	mp
0.7	[]	[]	[]	[]	
0.8	['note_off', 'note_on']	[0, 0]	[72, 76]	[64, 63]	mp
0.9	[]	[]	[]	[]	
1	['note_off', 'note_on']	[0, 0]	[76, 67]	[64, 50]	p
1.1	[]	[]	[]	[]	



	sec	msg_type	channel	note	velocity	new_dynamic	dynamic	accent	count	main_vol	depth	pedal	pan
0	0.0	['note_on']	[0]	[60]	[61]	p	p	0	1	[100, 100, 100]	[127, 127]	127	[64, 64]
1	0.1	[]	[]	[]	[]	p	NaN	0	0	0	0	0	0
2	0.2	['note_on']	[0]	[64]	[57]	p	p	0	1	0	0	0	0
3	0.3	[]	[]	[]	[]	p	NaN	0	0	0	0	0	0
4	0.4	['note_on']	[0]	[67]	[56]	p	p	0	1	0	0	0	0

기존 데이터 추출 알고리즘을 통해 추출한 엑셀 파일을 불러와 따로 전처리 코드를 만들어 빈 곳에 대한 셈여림 기호를 추가해 열(new\_dynamic)을 새로 만들어 주었습니다.



0.1초에는 데이터가 있고 0.2초에는 데이터가 없고 이런식으로 반복되는데 왜 그런건지와 비교 알고리즘 적용시에 이부분때문에 정확도가 떨어지지 않는지 확인 필요

# 빈 썸여림 데이터를 추가한 후 비교 알고리즘 성능 평가

기존 데이터로 평가시

```
[2]: 썸여림 유사도: 7.35%
```

새로운 데이터로 평가시

```
In [16]: dynamic_similarity = calculate_similarity_dynamic(input_file, target_file)
print(f'썸여림 유사도: {dynamic_similarity}')
```

썸여림 유사도: 35.87%

이유? : 썸여림을 평가하는 알고리즘은 정확도를 측정하는 것이 아닌 서로 다른 두 데이터에 대해 얼마나 유사한지를 따지는 것이기 때문에, 유사도 정확도에 차이가 없을 것이라 예상했지만, 평가에 차이가 생겼다.

아마도 빈 곳에 데이터가 채워지면서, 곡에 대한 시간 별 썸여림 추출이 정확해져, 조금 더 정확한 유사도 평가를 할 수 있던 것 같다.

셈여림 특징 추출 시 빈 시간대 데이터 채우기 (반드시 악보에 나와있는 셈여림표를 참고해서 정답 데이터의 셈여림 수정하기. 악보의 노트나 음이 발생한 시간대를 보면 정답데이터의 셈여림을 수정할 수 있을듯함)

1.악보에 나와 있는 셈여림표에 따른 데이터 추출 방법을 알아보고 적용시켜봐야 할 것 같습니다!

- 만약 Pro 버전에서 추출한 데이터에 셈여림 기호를 추출 할 수 없다면?

기존 데이터 추출 알고리즘을 사용해 데이터를 추출한 뒤, 빈 곳의 셈여림 기호를 채워주고, 셈여림이 변화하는 구간을 체크 해, 악보에 반영해주어야 할 것 같음.