

PDF to Music_chord

이 프로젝트는 PDF에 포함된 악보 정보를 추출하여 음악 코드로 변환하는 것을 목표로 합니다. 이를 통해 사용자들은 손쉽게 연주할 수 있는 코드 진행을 얻을 수 있습니다.



목차

- 1 프로젝트 개요
- 2 프로젝트 수행 절차 및 방법
- 3 프로젝트 수행 경과
- 4 자체 평가 의견



프로젝트 개요

1 프로젝트 주제 및 선정 배경

2 프로젝트 내용

3 활용 기술 및 재료



프로젝트 주제 및 선정 배경

작곡을 할 때 필요한 화성적 분석을

더 쉽고 빠르게 할 수 있는 방법은 없을까?

화성적 분석을 할 수 있는 프로그램을 만들어 보았습니다.

Mama

(Feat. 구윤희, 한예슬, 신종욱, 김나영)

uma' (2013.03) 수록곡

구윤희, 한예슬, 신종욱

Tempo

남)마 마 아련히 - 내 게 남아-버린 - 마 마 혼한이-름한-변 불러-보

한 평생 을못 난 자식 품 고살-았던--- 그 대생 각에- 오-- 난

마 마 철없던 그시 절 돌 이 켜보면 - 오 랜 시간우 린 많 이 변- 해 버렸죠

지 자로 태어 나 세월 에 주름 진다-는게 이 져알 아요- 얼마나외로운-지

매 일 같은자 리에 서 나 들바 라 보 며 상 처 뿐 인 마

눈 물 뿐 인 지 난 날 만 힘 들 때 는 뒤 들 들 아 향 상 내 가 서 있

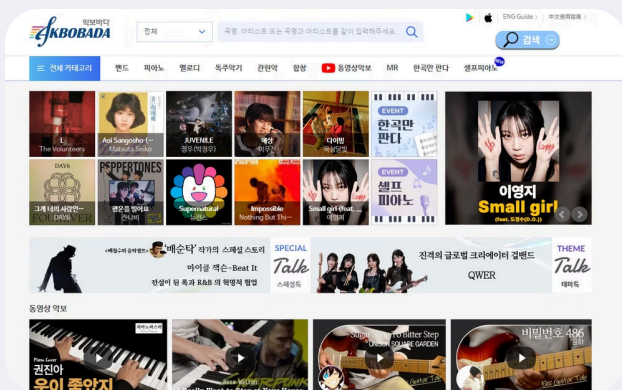
Made with Gamma

프로젝트 내용

- PDF를 텍스트로 읽어올 수 있는 방법
- 텍스트로 얻은 정보 중에 규칙과 데이터 정제 방법
- 정제된 데이터에서 코드의 단일화 할 수 있는 알고리즘
- 데이터의 정확성 검증

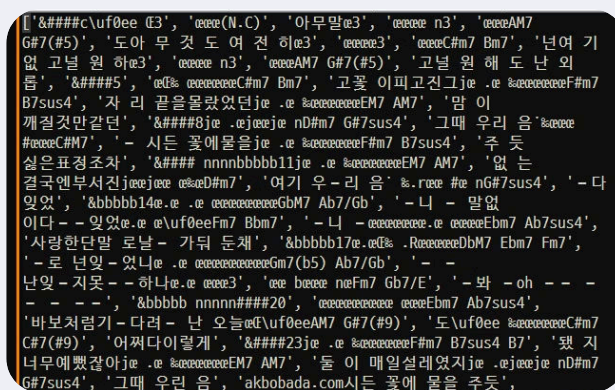


활용 기술 및 자료



PDF 악보 구하기

악보바다 데이터 활용



PDF to Text

PyPDF2, PyMuPDF

직접적으로 텍스트로 읽는 방법



PDF to Text

OpenCV, pdf2image

OCR을 통해서 구간을 나눠서 읽어오는 방식



프로젝트 수행 절차 및 방법

PDF 데이터 전처리

텍스트 변환 후 필요한 내용을 찾기 위한 알고리즘 설계

UI 개발

사용자가 편리하게 코드를 활용할 수 있도록 개발

1

2

3

패턴 분석과 변환

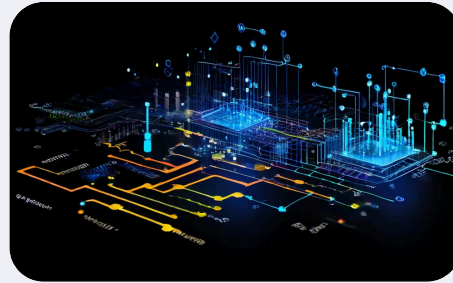
데이터들을 통해 패턴을 분석후 조표와 코드를 변환

프로젝트 수행 과정



PDF to Text 정보 및 모듈 선별

다양한 PDF to Text 변환 방식을 분석하여 최적의 솔루션을 선정하였습니다.



Text 정보 분리하는 알고리즘 설계

추출된 텍스트 데이터에서 필요한 정보를 효과적으로 분리하기 위한 알고리즘을 설계하였습니다.



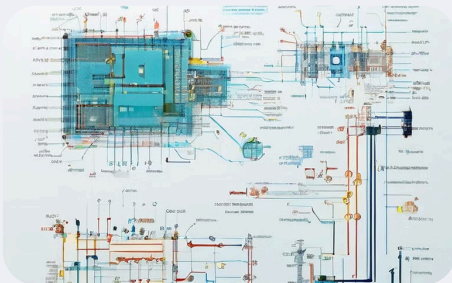
조표와 코드 변환의 필요한 알고리즘 설계

음악 이론에 기반하여 악보 데이터를 코드 정보로 변환할 수 있는 알고리즘을 개발하였습니다.



Ui 개발 및 서비스 구상

사용자가 편리하게 활용할 수 있는 UI를 개발하고, 전체적인 서비스를 구상하였습니다.



오류 수정 및 서비스 개발

지속적인 버그 색출을 토대로 서비스 안정화 및 서비스 업데이트

PDF to Text 정보 및 모듈 선별

텍스트

PyPDF2, PyMuPDF 같은 모듈

장점

1. 텍스트 자체를 가져오기 때문에 **높은 정확도**
2. 반복되는 패턴 형성
3. 빠른 텍스트화

단점

1. 전체를 가져오기 때문에 정보의 차별화가 필요
2. 분리를 위해 많은 알고리즘과 무거워짐

OCR

OpenCV를 통해 pdf2image 실행

장점

1. 원하는 텍스트의 위치를 가져와서 분리가 가능
2. 적은 알고리즘과 가벼움

단점

1. 이미지를 한번 거친 후 작업하기 때문에 다소 느림
2. OCR 방식으로 인해 낮은 정확도

Text 정보 분리

제목 및 정보 제거

Key 분리

노래 기호 제거

코드 분리

```
def text_setting(file_path):
    # try:
    #     folder_path = "C:/Python/Syntex/working/개인/악보" # 악보 폴더 위치
    #     title = input("곡 제목을 말해주세요 :")
    #     file_path = os.path.join(folder_path, f"{title}.pdf")
    # except:
    #     print("현재 폴더에 악보가 없습니다. 다시 확인 해주세요.")
    title = os.path.basename(file_path)
    reader = PdfReader(file_path)
    pages = reader.pages
    parts = []
    # 제목부분 및 필요없는 부분 구간설정
    def visitor_body(text, cm, tm, fontDict, fontSize):
        y = tm[5]
        if y > 665:
            parts.append(text)
    # 페이지0번째 텍스트 제외구간 설정
    pages[0].extract_text(visitor_text=visitor_body)
    text_body_page_0 = "".join(parts)
    A = text_body_page_0.split('\n')

    # 전체 페이지 정보
    text = ""
    for page in pages:
        text += page.extract_text()

    lines = text.split('\n')

    # 전체에서 제외시킬 내용을 찾아 삭제
    filtered_lines = [line for line in lines if line not in A]

    try:
        Key_name = pd.read_csv("Key_signature.csv", index_col='KEY')
```

[illegible]

정제 후

```
'AM7 G#7(#5)', 'C#m7 Bm7', 'AM7 G#7(#5)', 'C#m7 Bm7',  
'F#m7 B7sus4', 'EM7 AM7', 'D#m7 G#7sus4', 'C#M7',  
'F#m7 B7sus4', 'EM7 AM7', 'D#m7', 'G#7sus4', 'GbM7  
Ab7/Gb', 'Fm7 Bbm7', 'Ebm7 Ab7sus4', 'DbM7 Ebm7 Fm7',  
'Gm7(b5) Ab7/Gb', 'Fm7 Gb7/E', 'Ebm7 Ab7sus4', 'AM7  
G#7(#9)', 'C#m7 C#7(#9)', 'F#m7 B7sus4 B7', 'EM7 AM7',  
'D#m7 G#7sus4', 'C#M7', 'F#m7 B7sus4 B7', 'EM7 AM7',  
'D#m7', 'G#7sus4', 'GbM7 Ab7/Gb', 'Fm7 Bbm7', 'Ebm7  
Ab7sus4 Ab7', 'DbM7 Abm7 Db7', 'Gm7(b5) Gb7(b5)', 'Fm7  
F/A Bbm7 E7', 'Ebm7 Fm7', 'Gbm7 Abm7', 'Ab7sus4',  
'GbM7 Ab7/Gb', 'Fm7 F/A Bbm7 Ddim7', 'Ebm7 Cm7(b5)  
F/A', 'Bbm7 Abm7 Db7', 'Gm7(b5) C7(#9) C/E', 'Fm7 F7/A  
Bbm7 Ddim7', 'Ebm7', 'Ab7sus4', 'AM7 G#7(#5)', 'C#m7 Bm7', 'AM7  
G#7(#5)', 'DM7', 'C#m']
```

코드 변환

Chord 정보

Index	Type	Size	Value
0	str	11	AM7 G#7(#5)
1	str	8	C#m7 Bm7
2	str	11	AM7 G#7(#5)
3	str	8	C#m7 Bm7
4	str	11	F#m7 B7sus4
5	str	7	EM7 AM7
6	str	12	D#m7 G#7sus4
7	str	4	C#M7
8	str	11	F#m7 B7sus4
9	str	7	EM7 AM7
10	str	4	D#m7
11	str	7	G#7sus4

정제한 코드 정보

Key, 코드, 로마기호의 정보

KEY	I	ii	iii	IV	V	vi	vii
C_key	C	D	E	F	G	A	B
F_key	F	G	A	Bb	C	D	E
Bb_key	Bb	C	D	Eb	F	G	A
Eb_key	Eb	F	G	Ab	Bb	C	D
Ab_key	Ab	Bb	C	Db	Eb	F	G
Db_key	Db	Eb	F	Gb	Ab	Bb	C
Gb_key	Gb	Ab	Bb	B	Db	Eb	F
B_key	B	C#	D#	E	F#	G#	A#
E_key	E	F#	G#	A	B	C#	D#
A_key	A	B	C#	D	E	F#	G#
D_key	D	E	F#	G	A	B	C#
G_key	G	A	B	C	D	E	F#

Key를 기준으로 코드 → 로마

로마 기호로 데이터 변환

Index	Type	Size	Value
0	str	10	IVM7 iii7(#5)
1	str	7	vIm7 Vm7
2	str	10	IVM7 iii7(#5)
3	str	7	vIm7 Vm7
4	str	10	iIm7 V7sus4
5	str	7	IM7 NM7
6	str	10	vIm7 iii7sus4
7	str	3	viM7
8	str	10	iIm7 V7sus4
9	str	7	IM7 NM7
10	str	3	vIm7
11	str	6	iii7sus4

화성 분석이 가능한 상태로 변환

프로젝트 수행 경과

PDF 데이터 추출

PDF 파일에서 악보 이미지를 성공적으로 추출하였습니다.

개선 사항으로는 영어 가사의 차별화가 필요합니다.

코드 생성 알고리즘

추출된 악보 데이터를 분석하여 코드 진행을 파악하는 알고리즘을 개발하였습니다.

현재 정확도와 효율성을 향상 시키기 위한 작업을 진행 중입니다.

UI 구현

사용자가 쉽게 데이터를 얻을 수 있도록 개발 하였습니다.

현재 피드백을 반영하여 인터페이스를 개선하고 있습니다.

코드 진행 파악

데이터에서 코드 진행을 얻은 뒤 여러 데이터와 확인해서 코드 진행 수집

현재 더 정확한 진행을 찾기 위해 알고리즘 개발 중입니다.

자체 평가 의견

작업물의 필요성

PDF 내 악보 정보를 자동으로 추출하고 음악 코드로 변환하는 기술은 작곡의 많은 도움을 줄 것 입니다.

보완할 점

여러 문제점들을 보완하고 더 가볍고 나은 데이터를 추출 할 수 있도록 지속적인 개발을 할 것 입니다.

배우고 느낀 점

기존에 알던 지식을 기반으로 자동화를 시키기 위해 목표 설정 및 진행 방향을 세워보고 실현 시키기 위해 필요로 하는 기술을 찾으면서 더 다양한 지식을 얻었습니다.

또한 프로그램으로 만들어보며 개발 할 때 어떤 점을 신경 써야 되는지 어떻게 만들어야 효율이 올라갈지 생각 할 수 있는 시간이 되었습니다.





프로젝트 결과 요약



PDF 데이터 추출

PDF 파일에서 악보 정보를 성공적으로 추출하였습니다.



음악 코드 생성

추출된 악보 데이터를 분석하여 음악 코드로 변환하였습니다.



사용자 친화적 UI

사용자가 쉽게 데이터를 얻을 수 있게 인터페이스를 구현하였습니다.



향후 발전 방향

1

정확성 향상

다양한 악보 형식에 대한 인식 능력을 향상 시키고 코드 생성 알고리즘을 개선하겠습니다.

2

코드 진행 추천

여러 악보 데이터를 기반으로 코드 진행을 분석 한 뒤 장르의 어울리는 코드를 추천 하는 프로그램 개발 하겠습니다.

3

서비스 화

서비스 화를 진행 시켜서 여러 피드백을 받아 더 나은 프로그램으로 발전 시키겠습니다.