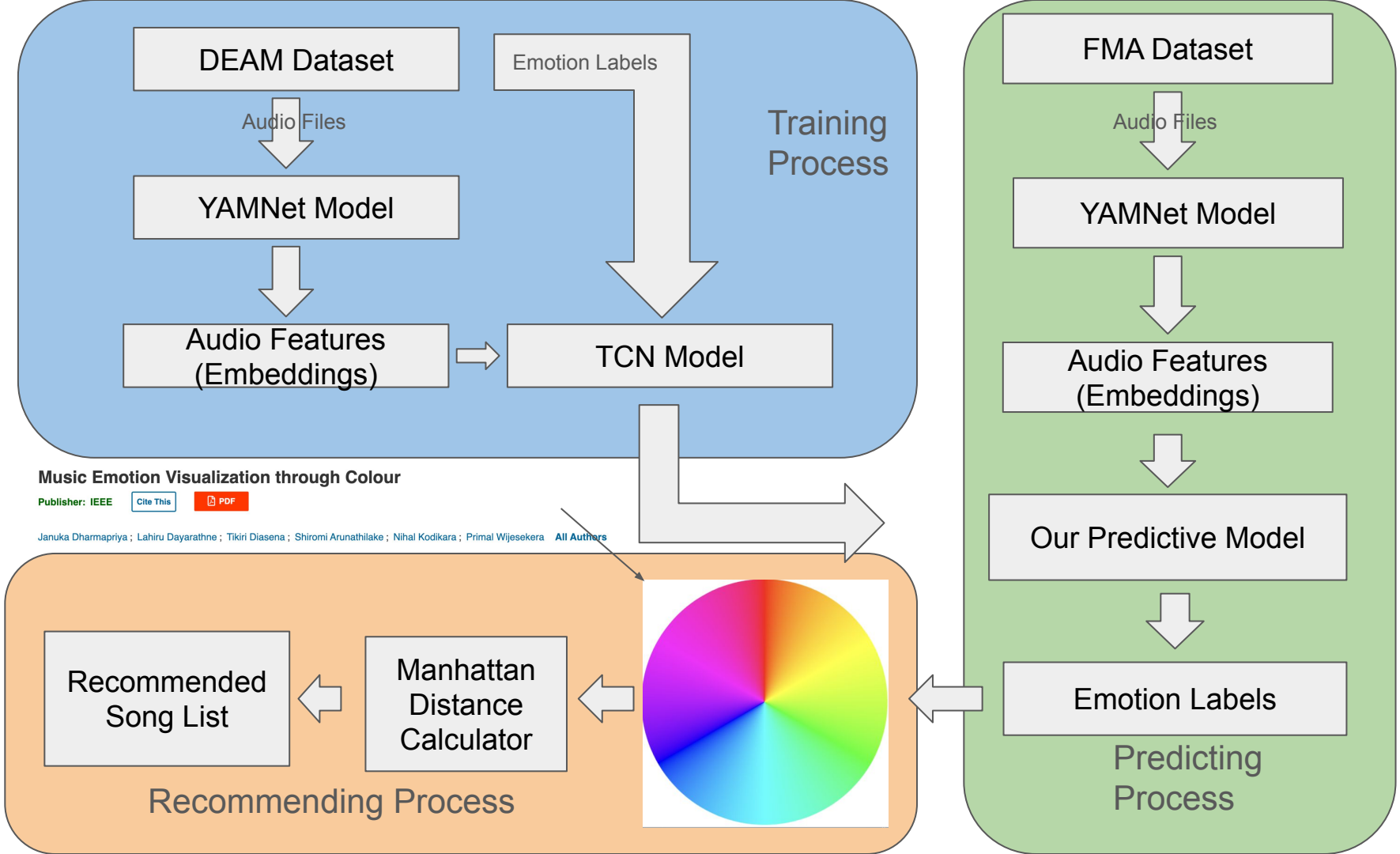
Music Recommendation System Based on Color-Based Emotions

Group 2 : Duanning Wang, Kaiyang Zhao & Zhinuo Li



Music Emotion Visualization through Colour

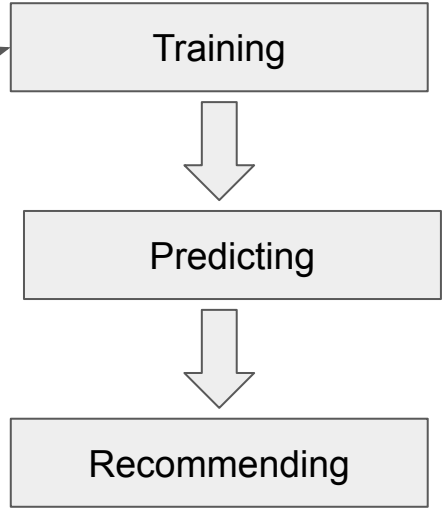
Publisher: IEEE Cite This PDF

Januka Dharmapriya ; Lahiru Dayarathne ; Tikiri Diasena ; Shiromi Arunathilake ; Nihal Kodikara ; Primal Wijesekera All Authors

main | 2 Branches | 0 Tags | Go to file | Add file | Code

KaiyangZhao0603 Update_0501_2 299f655 · 1 hour ago 16 Commits

DEAM_Dataset	Update_1	3 days ago
FMA_Excerpt	Update_0430	12 hours ago
__pycache__	Update_0501	1 hour ago
.DS_Store	Update_0501	1 hour ago
Linear_RGB_color_wheel_2.png	Add files via upload	3 days ago
Predicting.ipynb	Update_0501	1 hour ago
README.md	Initial commit	last week
Recommending.ipynb	Update_0501_2	1 hour ago
Training.ipynb	Update_0501	1 hour ago
emotionPredict.h5	Add files via upload	4 days ago
emotionPredict2.h5	Add files via upload	20 hours ago
emotion_values.txt	Update_0501	1 hour ago
gui.py	Update_0501	1 hour ago
models.py	Add files via upload	4 days ago
predicted_emotions.csv	Update_0430_2	11 hours ago
utils.py	Update_0430	12 hours ago



The Datasets that we used

Search this file

1	song_id	valence_mean	valence_std	arousal_mean	arousal_std
2	2	3.1	0.94	3	0.63
3	3	3.5	1.75	3.3	1.62
4	4	5.7	1.42	5.5	1.63
5	5	4.4	2.01	5.3	1.85
6	7	5.8	1.47	6.4	1.69
7	8	3.2	1.4	4.8	1.54
8	10	4	1.67	4.7	1.85
9	12	5.5	1.91	5.8	1.89
10	13	3.2	1.4	4	1.67
11	17	4.4	1.8	6	2.05
12	18	4.8	1.72	3.9	1.37
13	19	5.9	1.64	4.3	1.9

1.DEAM Dataset (For training, evaluation and testing):

Comprehensive genres

Annotated with emotion labels(Arousal & Valence)

Provides audio files with the same length (30 seconds)

2.FMA Dataset (For music recommendation):

Large dataset scale

Comprehensive metadate

Data Preprocess

1. Alignment

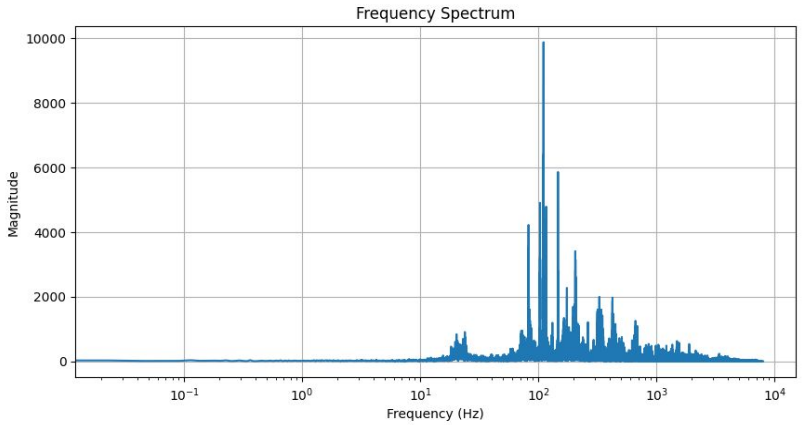
song_file_path

song_id	valence_mean	valence_std	arousal_mean	arousal_std
2	3.1	0.94	3	0.63
3	3.5	1.75	3.3	1.62
4	5.7	1.42	5.5	1.63
5	4.4	2.01	5.3	1.85
7	5.8	1.47	6.4	1.69

2. Augmentation (5406 in total)

- Pitch shift
- Lowpass filter (ladder filter)

Cutoff frequency = 1200Hz



3. Data split

20% test (1082)
25% validation (1081)

Model

- Yamnet: feature extraction
- TCN model

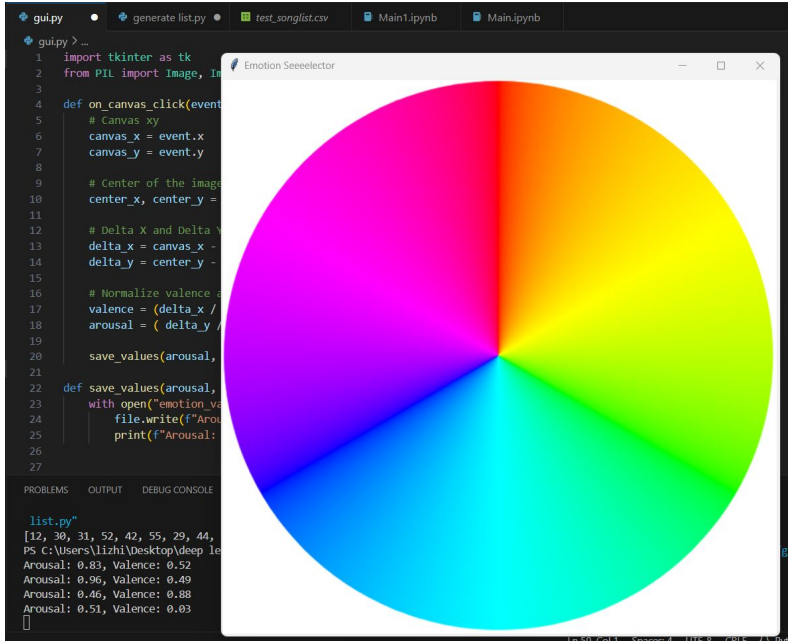
```
def build_model(input_shape):
    model = Sequential([
        TCN(input_shape=input_shape, nb_filters=64, kernel_size=6, nb_stacks=1,
            dilations=[1, 2, 4, 8, 16], padding='causal', use_skip_connections=True,
            dropout_rate=0.2, return_sequences=False, activation='relu', kernel_initializer='he_normal'),
        Dense(2, activation='linear', kernel_regularizer=l2(0.01))
    ])
    BatchNormalization(),
    optimizer = Adam(learning_rate=0.005)
    model.compile(optimizer=optimizer, loss='mean_squared_error', metrics=['mae'])
    return model
```

Test Loss: 0.922179639339447, Test MAE: 0.7563273906707764

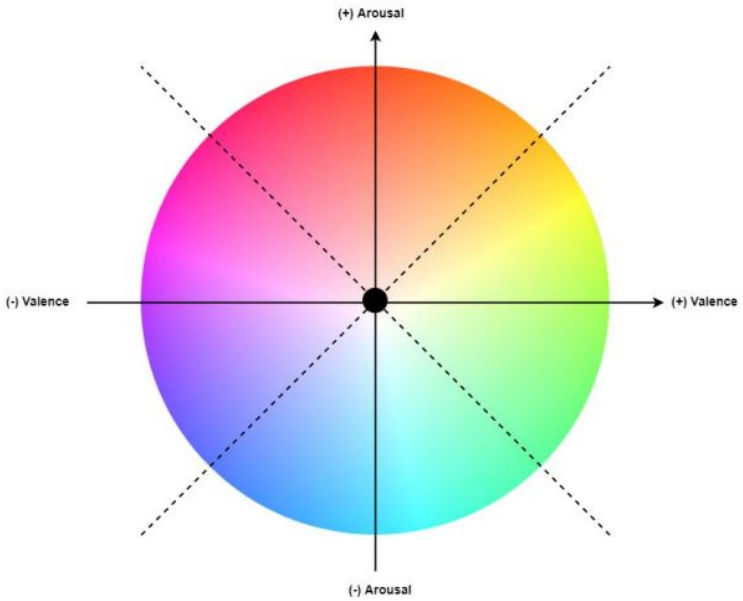
UI

“Music Emotion Visualization through Colour”, by Januka et al.

Emotion Selector



Mapping of the Itten’s colour system to Russell’s circumplex model of affect



UI Demo

musicId	Arousal(m	Valence(mean)
1	0.4	0.575
4	0.2625	0.2875
5	0.15	0.2
6	0.5125	0.35
7	0.7	0.725
8	0.3875	0.225
9	0.45	0.2875
10	0.4375	0.425
12	0.6875	0.575
13	0.825	0.5875
14	0.5625	0.575
15	0.6125	0.6625
16	0.5375	0.575
17	0.825	0.675
18	0.8	0.8375
19	0.7625	0.75
20	0.55	0.6375
21	0.8625	0.775
23	0.625	0.675
25	0.761364	0.784091
26	0.825	0.7625
28	0.4875	0.575
29	0.7	0.675
30	0.6125	0.5375
31	0.625	0.4375
32	0.575	0.3875

test_songlist

	A	B	C	D
27	32	0.575	0.3875	
28	33	0.6375	0.75	
29	34	0.6875	0.8375	
30	35	0.3375	0.625	
31	37	0.8	0.75	
32	38	0.8875	0.8375	
33	39	0.875	0.590909	
34	40	0.375	0.6625	
35	42	0.7625	0.6	
36	43	0.8625	0.7875	
37	44	0.6875	0.675	
38	46	0.8375	0.575	
39	47	0.4	0.4625	
40	48	0.725	0.8375	
41	49	0.7875	0.7125	
42	50	0.8375	0.5875	
43	51	0.825	0.7625	
44	52	0.5875	0.4625	
45	54	0.875	0.775	
46	55	0.625	0.4	
47	56	0.4125	0.275	
48	57	0.525	0.3625	
49	58	0.2625	0.25	
50	59	0.6	0.6	
51	60	0.4204545	0.40909090909100004	
52				
53				

test_songlist

Arousal: 0.7
Valence: 0.5

Manhattan Distance

The sum of absolute differences between points across all the dimensions

```
 list.py"
[12, 30, 31, 52, 42, 55, 29, 44, 59, 14]
PS C:\Users\lizhi\Desktop\deep learning>
```

Songs recommended after you click on the blue color

Recommending.ipynb — Group2_Final-Project

EXPLORER

GROUP2_FINAL-PROJECT

- __pycache__
- DEAM_Dataset
- FMA_Excerpt
- FMA_Metadata
- README.txt U
- tracks.csv U
- emotion_values.txt
- emotionPredict.h5
- emotionPredict2.h5
- gui.py
- Linear_RGB_color_wheel_2.png
- models.py
- predicted_emotions.csv
- Predicting.ipynb
- README.md
- Recommending.ipynb
- Training.ipynb
- utils.py

OUTLINE

TIMELINE

Recommending.ipynb

Recommending.ipynb > # 读取CSV的前三行用于生成列名

+ Code + Markdown Run All Restart Clear All Outputs ··· dl4m-hw3 (Python 3.9.19)

```python
import pandas as pd
from gui import get_values

df = pd.read_csv('predicted_emotions.csv')

ui_arousal, ui_valence = get_values()

# Example UI values for arousal and valence


# Function to calculate Manhattan distance
def calculate_manhattan_distance(x, y, valence, arousal):
    return abs(x - valence) + abs(y - arousal)

# Add a new column to the DataFrame for the Manhattan distance
df['distance'] = df.apply(lambda row: calculate_manhattan_distance(row['Valence'], row['Arousal

# Sort the DataFrame based on the 'distance' column and select the top 10 entries
closest_songs = df.sort_values(by='distance').head(10)

# closest matches
closest_song_ids = closest_songs['Song_ID'].tolist()

# Print  list
print(closest_song_ids)
```

[1] ✓ 2.1s Python

```
[31389.0, 5936.0, 14318.0, 21087.0, 615.0, 12486.0, 7388.0, 3761.0, 31391.0, 16158.0]
```

main* ⊗ 0 ⚠ 0 Live Share Ln 5, Col 1

Songs recommended after you click on the red color

Recommending.ipynb — Group2_Final-Project

EXPLORER

GROUP2_FINAL-PROJECT
- __pycache__
- DEAM_Dataset
- FMA_Excerpt
- FMA_Metadata
  - README.txt U
  - tracks.csv U
- emotion_values.txt
- emotionPredict.h5
- emotionPredict2.h5
- gui.py
- Linear_RGB_color_wheel_2.png
- models.py
- predicted_emotions.csv
- Predicting.ipynb
- README.md
- Recommending.ipynb
- Training.ipynb
- utils.py

OUTLINE

TIMELINE

Recommending.ipynb ●

Recommending.ipynb > import os

+ Code + Markdown | Run All Restart Clear All Outputs ··· dl4m-hw3 (Python 3.9.19)

```python
import pandas as pd
from gui import get_values

df = pd.read_csv('predicted_emotions.csv')

ui_arousal, ui_valence = get_values()

# Example UI values for arousal and valence


# Function to calculate Manhattan distance
def calculate_manhattan_distance(x, y, valence, arousal):
    return abs(x - valence) + abs(y - arousal)

# Add a new column to the DataFrame for the Manhattan distance
df['distance'] = df.apply(lambda row: calculate_manhattan_distance(row['Valence'], row['Arousal

# Sort the DataFrame based on the 'distance' column and select the top 10 entries
closest_songs = df.sort_values(by='distance').head(10)

# closest matches
closest_song_ids = closest_songs['Song_ID'].tolist()

# Print  list
print(closest_song_ids)
```

[1] ✓ 2.2s Python

```
[31389.0, 5936.0, 12486.0, 14318.0, 615.0, 21087.0, 3761.0, 31391.0, 16158.0, 7388.0]
```

main* ⊗ 0 ⚠ 0 Live Share Ln 5, Col 1

Thank you!