

```
import os
```

```
os.chdir("../")
```

```
from tension_analyzer.tension_visualizer import TensionVisualizer, quick_load_and_visualize
```

```
viz = TensionVisualizer()
```

```
viz.load_tension_data("tension_analyzer/outputs/tension_data/tension_급_유턴_20250610_155915.jsc
```

✅ TensionVisualizer 초기화 완료

얼굴 이미지 디렉토리: video\_analyzer/preprocessed\_data/debug\_faces/chimchakman

✅ 텐션 데이터 로드 완료

영상명: f\_004\_급\_유턴\_1.0\_24.0

지속시간: 22.9초

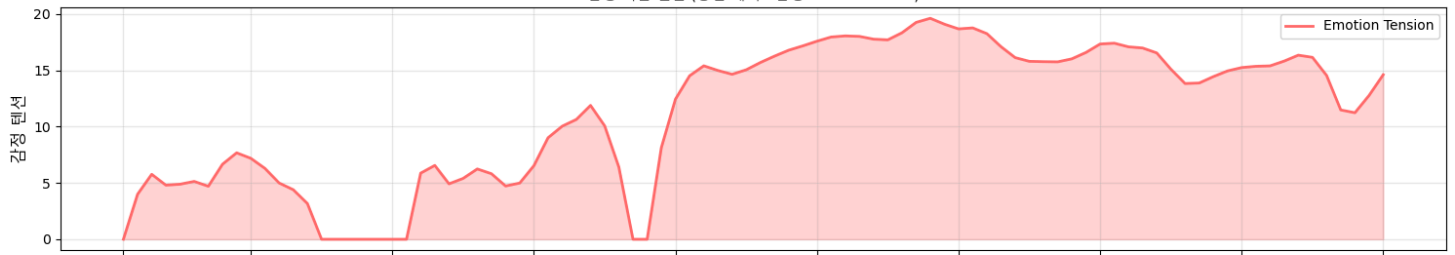
텐션 포인트: 90개

True

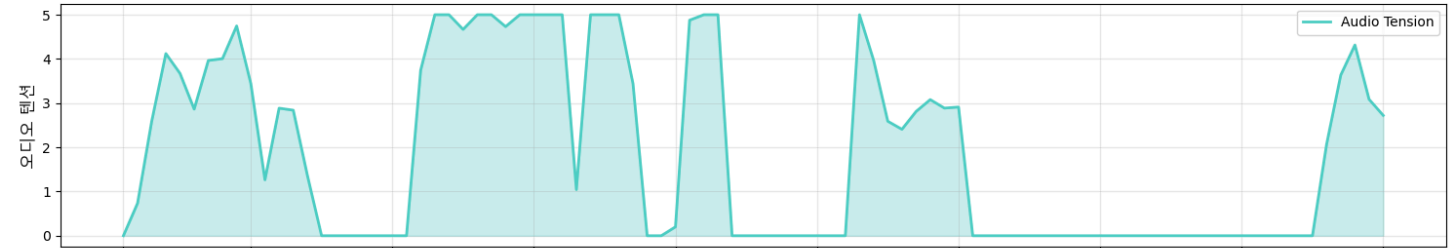
```
viz.plot_tension_curves()
```

## 텐션 분석 결과: f\_004\_급\_유턴\_1.0\_24.0

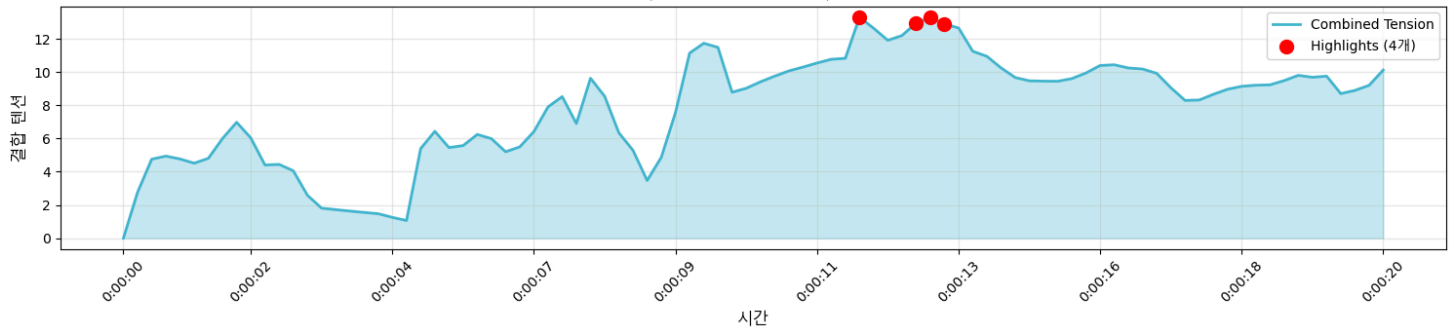
감정 기반 텐션 (중립 제외 7감정 + Arousal×10)



음성 기반 텐션 (VAD 필터링된 Voice RMS)



결합 텐션 (감정 70% + 오디오 30%) + 하이라이트



텐션 통계:

평균: 7.85

최대: 13.31

최소: 0.00

표준편차: 3.26

음성 활동 비율: 41.2%

`viz.plot_emotion_pie_chart()`

✓ 감정 데이터 로드: (92, 10)

Anger: 전체 81개, 양수 69개, 평균 2.0893

Contempt: 전체 81개, 양수 8개, 평균 1.5905

Disgust: 전체 81개, 양수 61개, 평균 2.5349

Fear: 전체 81개, 양수 55개, 평균 2.8216

Happiness: 전체 81개, 양수 32개, 평균 1.6166

Neutral: 전체 81개, 양수 24개, 평균 1.6409

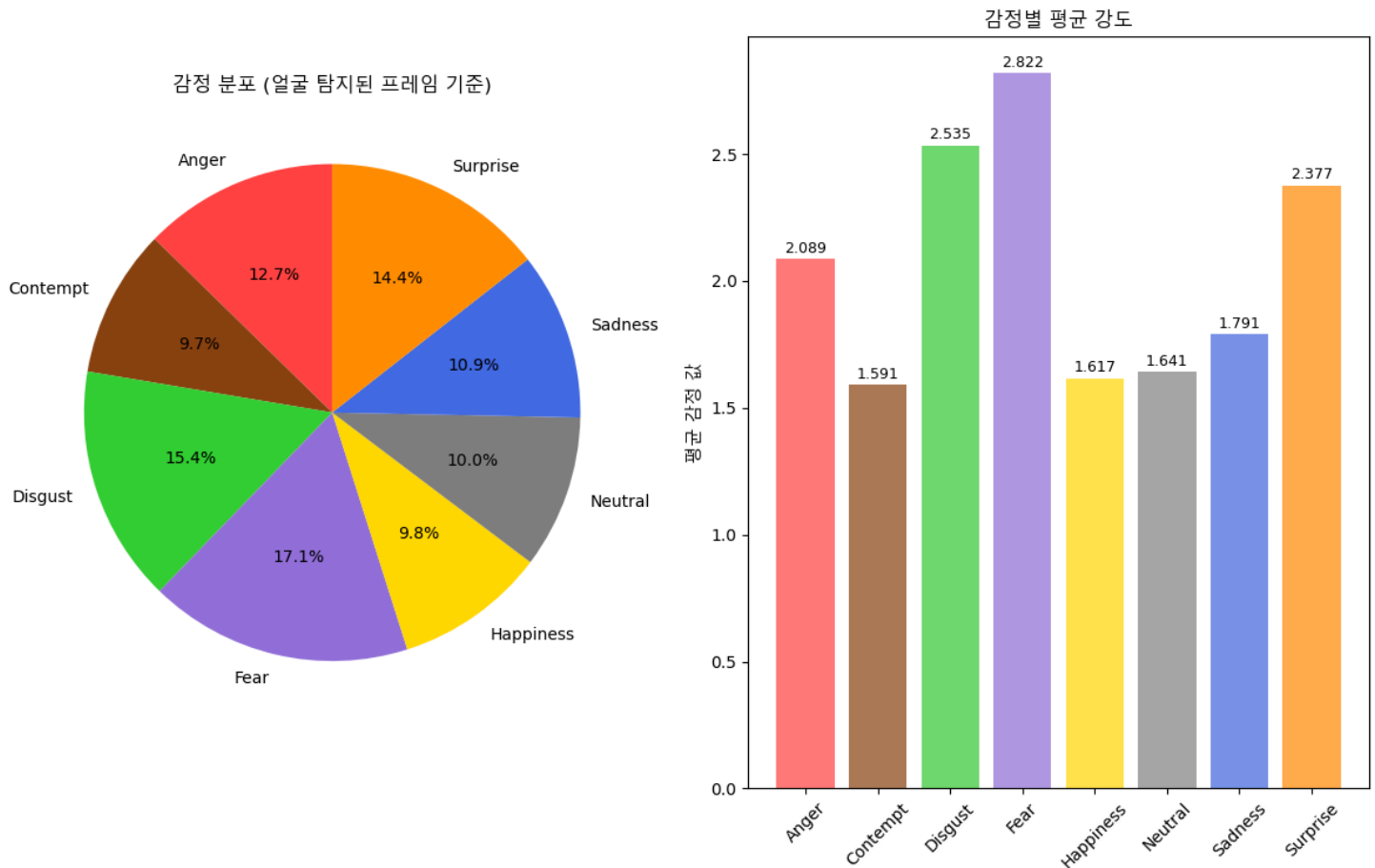
Sadness: 전체 81개, 양수 58개, 평균 1.7906

Surprise: 전체 81개, 양수 28개, 평균 2.3768

양수값만 필터링한 감정 평균값: [2.08933633 1.59051093 2.53490231 2.82163737 1.61179064074 2.37679974]

데이터 범위: 최소 1.591, 최대 2.822

감정 분석: f\_004\_급\_유턴\_1.0\_24.0



주요 감정:

1. Fear: 2.822

2. Disgust: 2.535

3. Surprise: 2.377

```
viz.debug_faces_dir = "video_analyzer/preprocessed_data/chimchakman_f_004_급_유턴_10_240_2025061  
viz.show_emotion_peak_faces(top_n=1)
```

- ✓ 감정 데이터 로드: (92, 10)
- ✓ HDF5에서 얼굴 폴더 자동 감지: video\_analyzer\preprocessed\_data\debug\_faces\chimo

### 감정별 피크 순간의 실제 얼굴: f\_004\_급\_유턴\_1.0\_24.0

Anger  
0:00:14  
값: 4.704



Contempt  
0:00:06  
값: 2.411



Disgust  
0:00:18  
값: 5.217



Fear  
0:00:20  
값: 4.280



Happiness  
0:00:08  
값: 7.260



Sadness  
0:00:20  
값: 3.299



Surprise  
0:00:00  
값: 3.646



Valence  
0:00:08  
값: 0.855



Arousal  
0:00:15  
값: 0.943



🎭 감정 피크 표시 완료 (9개: 7감정 + VA)

```
print("\n📄 주피터랩 사용 예시:")
print("`python`")
print("from tension_visualizer import TensionVisualizer, quick_load_and_visualize")
print("")
print("# 방법 1: 단계별 시각화 (자동 얼굴 폴더 감지)")
print("viz = TensionVisualizer()")
print("viz.load_tension_data('your_filename_pattern')")
print("viz.auto_find_faces_dir()          # HDF5에서 자동 감지")
print("viz.plot_tension_curves()          # 셀 1")
print("viz.plot_emotion_pie_chart()      # 셀 2")
print("viz.show_emotion_peak_faces()     # 셀 3")
print("")
print("# 방법 2: 빠른 로드 (자동 감지 포함)")
print("viz = quick_load_and_visualize('your_filename_pattern')")
print("viz.plot_tension_curves()")
print("")
print("# 방법 3: 모든 플롯 한번에 (자동 감지 포함)")
print("show_all_plots('your_filename_pattern')")
print("")
print("# 방법 4: 수동 폴더 지정")
print("viz = TensionVisualizer(debug_faces_dir='custom_path/chimchakman')")
print("viz.load_tension_data('your_filename_pattern')")
print("")
print("# 📁 타임스탬프 파일명 형식: timestamp_015750_face0_chimchakman_sim0.748.jpg")
```

 주피터랩 사용 예시:

```
```python
from tension_visualizer import TensionVisualizer, quick_load_and_visualize

# 방법 1: 단계별 시각화 (자동 얼굴 폴더 감지)
viz = TensionVisualizer()
viz.load_tension_data('your_filename_pattern')
viz.auto_find_faces_dir()          # HDF5에서 자동 감지
viz.plot_tension_curves()          # 셀 1
viz.plot_emotion_pie_chart()       # 셀 2
viz.show_emotion_peak_faces()      # 셀 3

# 방법 2: 빠른 로드 (자동 감지 포함)
viz = quick_load_and_visualize('your_filename_pattern')
viz.plot_tension_curves()

# 방법 3: 모든 플롯 한번에 (자동 감지 포함)
show_all_plots('your_filename_pattern')

# 방법 4: 수동 폴더 지정
viz = TensionVisualizer(debug_faces_dir='custom_path/chimchakman')
viz.load_tension_data('your_filename_pattern')

# 📁 타임스탬프 파일명 형식: timestamp_015750_face0_chimchakman_sim0.748.jpg
```