

# MediaPipe를 활용한 얼굴 랜드마크 비디오 데이터셋 구축

소프트웨어학부

20215214

최재원

## 과제 목표

본 과제는 MediaPipe를 이용해 비디오에서 인물의 얼굴 영역을 추출하는 방식으로 얼굴 랜드마크 비디오 데이터셋을 구축하는 과정을 구현합니다.

## 과제 구성

구성으로는 비디오 원시 데이터(raw data) 수집, 비디오 시간영역 자르기(slicing), 비디오 공간 영역 자르기(cropping), 비디오 얼굴 랜드마크 검출로 이루어져 있습니다.

### 1. 비디오 원시 데이터(raw data) 수집

- 비디오 원시 데이터를 수집하기 위해 yt-dlp를 사용해 metadata.csv파일 내의 video\_id 열에 저장되어 있는 youtube 링크를 받아와 해당 링크의 영상을 받아오는 방식입니다.
- 아래 명령어를 통해 video\_id를 받아와 output\_folder에 video\_id.mp4의 형태로 저장되게 됩니다.

```
- command = [  
-     "yt-dlp",  
-     f"https://www.youtube.com/watch?v={video_id}",  
-     "-o", f"{output_folder}/{video_id}s.%(ext)s",  
-     "--recode-video", "mp4"  
- ]
```

- 실행 : python download.py -i {csv\_file} -o {output\_folder}  
인자를 줄 때 파일 이름을 입력하지 않을 경우 metadata.csv를 기본값으로 사용하게 됩니다. Output\_folder도 입력하지 않을 경우 source\_videos를 기본 폴더로 사용하게 됩니다.

### 2. 비디오 시간영역 자르기(slicing)

- 얼굴이 검출되는 영상만을 따로 저장하기 위한 절차입니다. Mediapipe를 이용하여 얼굴이 검출되는 시간을 측정하여 최소 측정시간 이상으로 검출 될 경우 그 시간 영역을 잘라 저장하는 절차입니다.
- 카메라를 이용한 방법과 비디오만을 이용하는 방법을 구현하였습니다. 카메라만을 이용하는 방식은 카메라에 얼굴이 검출되는 시간이 최소 측정 시간 이상이면 그 영상을 따로 잘라 저장합니다. 비디오만을 이용할 경우 영상에서 얼굴이 검출되는 시간이 최소 측정 시간 이상이면 그 영상을 따로 잘라 저장합니다.
- 스크립트를 실행할 때 인자는 카메라를 사용할 때의 인자와 비디오만을 이용할 때의 인자가 다릅니다.

우선 카메라를 사용할 때의 경우 `python slicer.py -camera` 로 `-camera`만 입력하면 카메라를 사용하게 됩니다.

- 실행 :  
비디오만을 이용 할 경우 `python slicer.py -i {input_video.mp4 or input_path} -o {output_path}` 로 input으로는 경로 혹은 파일을 주면 되고, output으로는 경로를 입력해야 합니다.

### 3. 비디오 공간 영역 자르기(cropping)

- Slicing을 거쳐 저장된 영상에서 bounding box를 통해 얼굴을 검출하며 bounding box가 검출되는 영역을 구하고, 모든 bounding box를 포함하는 super bounding box로 정해 인물 영역을 구하여 해당 공간 영역을 자르는 절차입니다.
- Bounding box의 위치를 검출될 때마다 업데이트 후, 비율이 맞지 않는 문제를 super bounding box의 width와 height의 긴 축에 맞춰 정사각형 비율로 맞춰주었습니다.
- 실행 : `python crop.py -i {input_video.mp4 or input_path} -o {output_path}`  
input은 비디오 파일 혹은 경로를 줄 수 있고, output으로는 경로를 입력해야 합니다. Output\_path를 주지 않을 경우 `cropped_video`파일로 저장됩니다.

### 4. 비디오 얼굴 랜드마크 검출

- Cropping 한 영상을 원하는 버전에 맞게 랜드마크를 영상 위에 그려서 저장을 하는 절차입니다.
- 버전 정보  
버전 1 ( `-v 1` ): MediaPipe의 기본 얼굴 랜드마크 그리기 방식  
버전 2 ( `-v 2` ): TESSELATION 없이 그리는 방식  
버전 3 ( `-v 3` ): FACE\_OVAL 없이 그리는 방식  
버전 4 ( `-v 4` ): 흑백 이미지에 FACE\_OVAL 없이 그리는 방식

- 실행: `python video_landmark.py -i {input_video.mp4 or input_path} -v {version_number} -o {output_path}`  
input은 비디오 파일 혹은 폴더 경로를 줄 수 있고, 버전은 1~4로 주지 않을 경우 MediaPipe의 기본 얼굴 랜드마크 그리기 방식인 버전 1로 주어집니다. Output은 경로를 주면 해당 경로에 소스 영상에 랜드마크가 그려진 영상이 저장되게 됩니다.

## 5. 전체 자동화 프로세스

- Python sequential.py 를 실행을 하면 metadata.csv에 저장되어 있는 video\_id를 읽어 전부 다운로드 후, 소스 동영상 파일들을 slicing작업을 하여 저장되고, slicing된 영상 파일들을 cropping 작업을 하여 저장된 후, 각 cropping 된 영상 파일들을 landmark를 입력해 metadata파일과 함께 저장된다.
- Slicing, cropping, landmark 단계를 거치면서 전부 비디오별로 정리되어 저장됩니다.
- 모든 프로세스를 진행하게 되면 source\_videos, slice\_videos, cropped\_videos, landmark\_videos 폴더가 현재 폴더 내에 저장되게 됩니다.

## 결론

본 과제를 통해 구현된 프로그램은 비디오 다운로드부터 시간 및 공간 영역의 자르기, 얼굴 랜드마크 검출 및 저장까지 일련의 과정을 효율적으로 수행할 수 있음을 확인하였습니다.

이 프로그램은 영상 데이터 수집 및 전처리의 자동화를 목표로 하며, 특히 얼굴이 포함된 구간만을 효과적으로 추출하고 처리하는 기능을 통해, 영상 데이터를 활용한 다양한 연구와 작업의 준비 단계를 크게 단축할 수 있습니다.

프로그램은 MediaPipe의 정확성과 성능을 활용하여 데이터의 품질을 보장하며, yt-dlp, ffmpeg, Slicing, Cropping 등 다양한 기술을 조합하여 데이터 처리 파이프라인을 지원합니다.

이를 통해 대규모 비디오 데이터셋을 구축하거나, 기계 학습 및 딥러닝 모델의 학습용 데이터를 준비하는 데 있어 유용한 도구로 활용될 가능성을 보여주었습니다.