**MediaPipe를 활용한 실시간 비전 인식**

30501 김도현

## **1페이지**

서론 - 실시간 비전 AI의 시작, MediaPipe  
 이 노트북은 구글이 개발한 오픈소스 프레임워크인 MediaPipe를 활용하여, 복잡한 컴퓨터 비전 작업을 얼마나 손쉽게 구현할 수 있는지를 보여준다. 특히 MediaPipe Hands 솔루션은 별도의 고성능 GPU 없이도 웹캠 영상만으로 실시간 21개 손가락 관절(Landmark) 추적이 가능하도록 최적화되어 있다. 첫 단계로, cv2.VideoCapture(0)을 통해 컴퓨터에 연결된 기본 웹캠을 활성화하고, mp.solutions.hands.Hands()를 호출하여 손 탐지 모델을 메모리에 로드했다. 이 초기화 과정은 앞으로 수행될 모든 손 인식 작업의 기반을 마련하는 중요한 단계이다.

## **2페이지**

손 탐지와 랜드마크의 시각화 과정  
 웹캠으로부터 while 루프를 통해 지속적으로 이미지 프레임을 읽어온다. OpenCV는 기본적으로 이미지를 BGR(Blue, Green, Red) 색상 순서로 처리하지만, MediaPipe 모델은 RGB 순서를 표준으로 사용하므로 cv2.cvtColor 함수를 통해 색상 공간 변환을 수행해야 했다. 변환된 이미지는 hands.process() 메소드에 입력되어 실제 손 탐지와 랜드마크 계산을 수행한다. 그 결과는 results.multi\_hand\_landmarks에 저장되며, 영상에서 손이 감지되었을 경우에만 이 객체가 값을 가진다. 감지된 각 손의 랜드마크 정보(21개의 x, y, z 좌표)를 mp.solutions.drawing\_utils.draw\_landmarks 함수에 전달하여 원본 프레임 위에 관절 위치와 연결선을 시각적으로 덧그리는 과정을 구현했다.

## **3페이지**

특정 관절(Landmark) 좌표의 정밀 추적  
 MediaPipe는 21개의 랜드마크에 0부터 20까지 고유 ID를 부여한다. 이를 활용하면 전체 손 모양뿐만 아니라 특정 손가락 끝과 같은 특정 부위의 움직임을 정밀하게 추적할 수 있다. 노트북에서는 검지 손가락 끝에 해당하는 ID 8번 랜드마크를 타겟팅했다. landmark.x와 landmark.y 값은 이미지의 전체 너비와 높이에 대한 상대적인 비율(0.0 ~ 1.0)로 제공되므로, img.shape에서 얻은 실제 이미지의 너비(w)와 높이(h)를 곱하여 픽셀 단위의 절대 좌표(cx, cy)로 변환했다. 이 좌표를 중심으로 cv2.circle 함수를 사용해 시각적인 표식을 그림으로써, 특정 관절의 위치를 실시간으로 하이라이트하는 기능을 성공적으로 구현했다.

## **4페이지**

코드의 재사용성을 위한 객체 지향 프로그래밍  
 기능이 복잡해짐에 따라 코드의 구조화 및 재사용성에 대한 필요성이 대두된다. 이를 해결하기 위해 손 탐지와 관련된 모든 기능을 handDetector라는 파이썬 클래스로 캡슐화했다. \_\_init\_\_ 생성자에서는 손 탐지 모드, 최대 손 개수, 탐지 신뢰도 임계값 등 MediaPipe Hands 모델의 세부 파라미터를 설정할 수 있도록 설계했다. findHands 메소드는 이미지에서 손을 찾아 랜드마크를 그리는 기존의 기능을 수행하고, findPosition 메소드는 특정 손의 모든 랜드마크 ID와 픽셀 좌표를 리스트 형태로 반환하는 새로운 기능을 추가했다. 이러한 객체 지향적 접근은 코드의 가독성을 높이고 향후 다른 프로젝트에서의 재사용을 용이하게 한다.

## **5페이지**

랜드마크 좌표를 이용한 제스처 인식 - 손가락 카운팅  
 handDetector 클래스를 활용하여 한 단계 더 나아간 응용 프로그램, 즉 펴진 손가락의 개수를 세는 기능을 구현했다. 각 손가락 끝 랜드마크(ID 4, 8, 12, 16, 20)의 y 좌표를 그 아래에 위치한 관절의 y 좌표와 비교하는 간단한 논리를 사용했다. 만약 손가락 끝의 y 좌표가 그 아래 관절의 y 좌표보다 작다면, 해당 손가락은 펴진 것으로 간주했다. 엄지손가락은 다른 손가락과 움직임 축이 다르므로, x 좌표를 기준으로 펴짐 여부를 판단하는 예외 처리를 추가했다. 이렇게 계산된 펴진 손가락의 총 개수를 cv2.putText을 사용하여 화면에 숫자로 출력함으로써, 간단한 형태의 손 제스처 인식 시스템을 완성했다.

## **6페이지**

MediaPipe의 확장성 - 얼굴 인식 적용  
 MediaPipe의 강력함은 손 인식에만 국한되지 않는다. 프레임워크는 얼굴, 자세, 객체 등 다양한 인식 솔루션을 모듈 형태로 제공한다. 노트북의 마지막 부분에서는 mp.solutions.face\_detection 모듈을 사용하여 실시간 얼굴 인식을 시연했다. 손 인식과 유사한 파이프라인으로, FaceDetection 모델을 초기화하고 웹캠 프레임을 입력하여 얼굴 위치를 탐지한다. detection.location\_data.relative\_bounding\_box에서 얻은 정보를 바탕으로 mp.solutions.drawing\_utils.draw\_detection 함수가 감지된 얼굴 주변에 사각형(bounding box)을 그려준다. 이를 통해 MediaPipe가 얼마나 유연하고 확장 가능한 컴퓨터 비전 도구인지를 명확히 보여주었다.

## **7페이지**

결론 - MediaPipe를 통한 비전 AI의 민주화  
 본 mediapipe.ipynb는 MediaPipe와 OpenCV를 결합하여 복잡한 실시간 비전 애플리케이션을 얼마나 빠르고 효율적으로 개발할 수 있는지를 증명했다. 단순 랜드마크 시각화에서 시작하여, 특정 좌표 추적, 코드의 객체 지향적 리팩토링, 그리고 랜드마크 좌표를 해석하여 의미 있는 정보(펴진 손가락 개수)를 추출하는 제스처 인식까지, 점진적으로 심화되는 개발 과정을 체계적으로 보여주었다. 이는 MediaPipe가 전문가가 아닌 개발자나 학생들에게도 최신 AI 비전 기술의 접근성을 크게 높여주는 '기술의 민주화'에 기여하고 있음을 시사한다.