# Week 18

## Mediapipe Hands: On-device Real-time Hand Tracking

Hand Tracking 기술은 기계의 제스쳐 제어, 수화 이해 등의 기술을 위한 중요 구성 요소이다. VR/AR 분야에서도 핵심적인 기술이다. 기존의 Hand Tracking 모델들은 손에 장비를 착용하거나, 특수한 카메라를 사용하거나 하는 하드웨어에 의존적인 방식이었지만 본 모델은 이를 극복하고

- 카메라 이외의 추가적 하드웨어가 없음
- 2개 이상의 손도 탐지 가능하고, 손의 일부가 가려지더라도 탐지됨
- 모바일 환경에서 실시간 연산이 가능

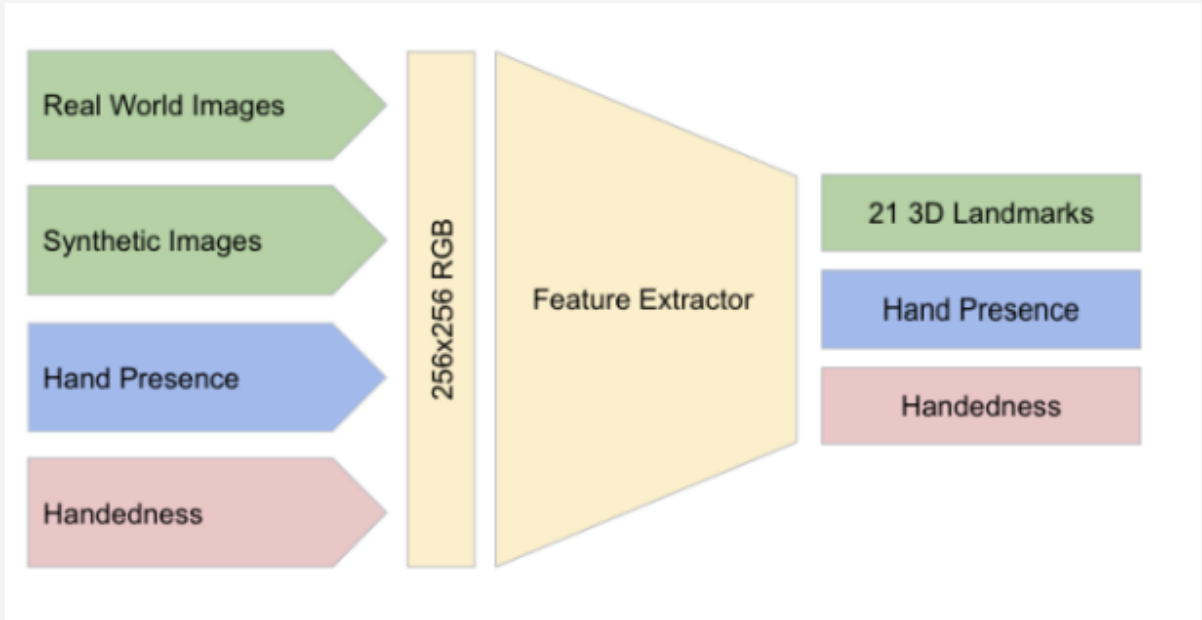
이라는 성과를 냈다.

이 모델은 2 stage Detector 구조를 이루고 있다.

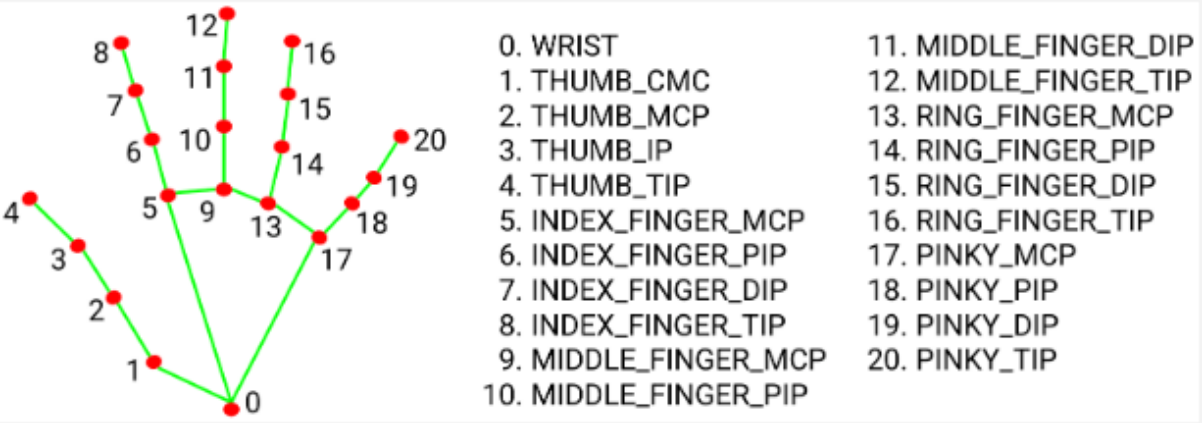
1. Bounding Box를 찾는 손바닥 detector
2. 각 bounding box(손) 별로 21개의 key points 탐지

Palm Detector는 모바일 환경에서 실시간 얼굴 탐지에 사용되던 BlazeFace 모델과 비슷하게 설계했다. 그림을 보면 convolution 중간중간 feature map을 사용하는 FPN(Feature Pyramid Network)구조와 유사하다는 것을 알 수 있다. 그런데 왜 Hand Detector가 아닌 Palm Detector를 사용했을까? 손의 위치를 찾는 것이 생각보다 쉽지 않기 때문이다. 얼굴의 경우 눈. 코, 입 등 뚜렷한 특징들이 있는 반면 손에서는 그렇다 할 특징이 존재하지 않는다. 따라서 손이 아닌 손바닥(손가락 뺀), Palm Detector를 사용했으며 다음과 같은 특징이 존재한다.

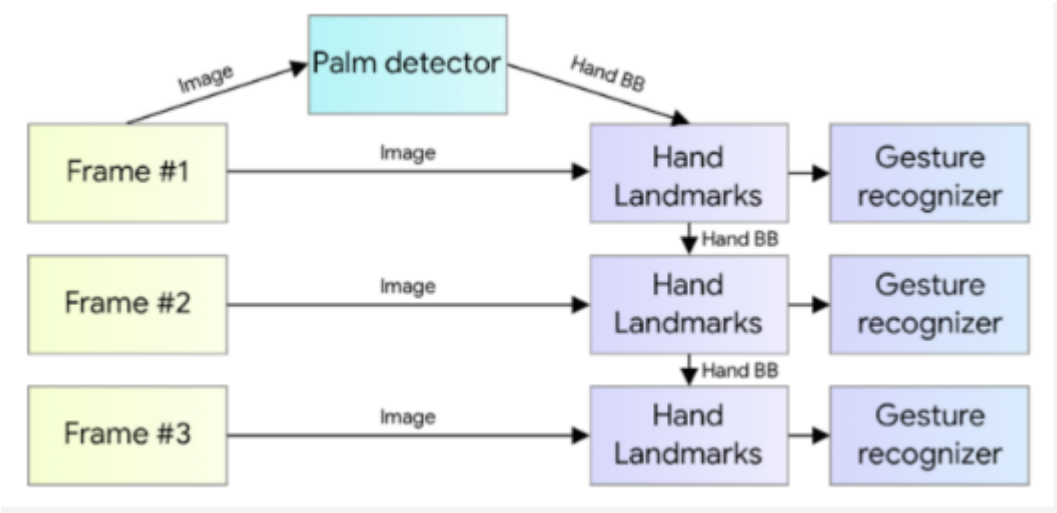
- 손에비해 NMS(Non-Max Suppression)가 잘 먹힌다.
    - 손가락 같이 삐쭉삐쭉하면 Region Proposal이 쉽지 않다.
    - Object Detection 배경지식 참고
- 손바닥의 bounding box 모양은 정사각형만 고려해도 된다.
    - 일반적인 Region Proposal은 가로:세로 비율이 1:1, 1:2, 2:1인 경우를 고려한다.
- focal loss를 이용한다.
    - object detection에서 사용하는 loss function.



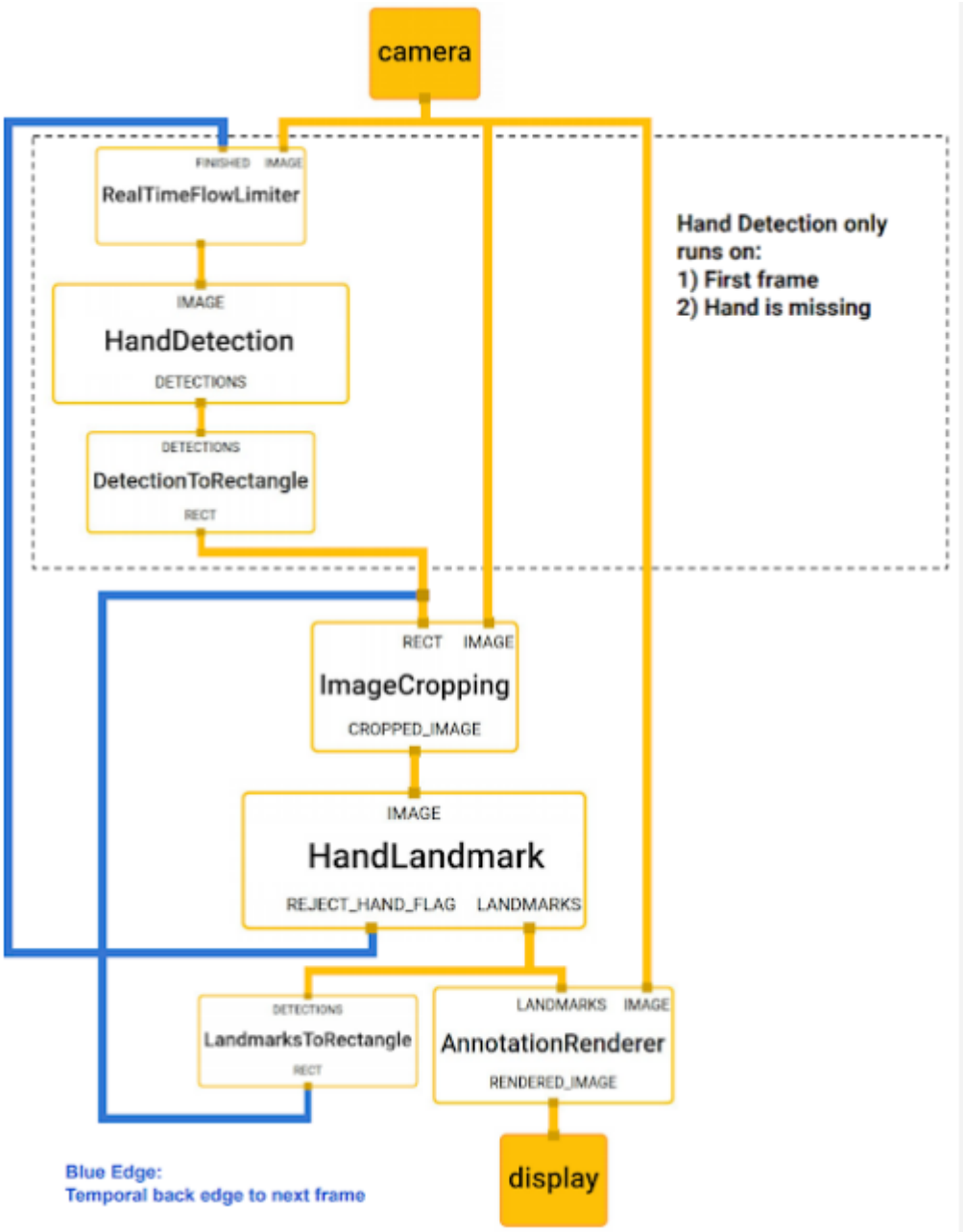

- **Input Data**
  - Real world, Synthetic Images : 아래 datasets 목차에서 다룸
  - Hand Presence : '왼손/오른손' 라벨링
  - Handedness : '이미지에 손이 존재하는가?' 라벨링




- 대략적인 손 모양이 정해져있음
- keypoint만 구하면 3D 손 모양을 비교적 정확하게 렌더링 가능
- 상대적 깊이 차원을 구할 수 있음




현재 프레임에서 LandMarks를 계산할 때 Hand bounding box를 놓쳤는지 여부를 flag로 설정한다(Handedness 값). 만약 손을 놓치지 않았다면 Palm detecting을 실행하지 않고 현재 프레임의 keypoints에서 후속 프레임의 손 위치를 추정한다. 이를 그래프로 표기하면 다음과 같다.




## In-the-wild datasetPermalink

일반적인 이미지 6천 장

- 다양한 배경
- 다양한 손의 크기
- Negative Sample (손 없음)

## In-house collected gesture datasetPermalink

디테일 중심의 이미지 1만 장

- 배경과 손의 크기는 제한
- 손의 각도와 가능한 제스처에 집중

# Learning Deep Features for Discriminative Localization

## 문제와 해결책

딥러닝 기술이 점점 정확도가 높아져가며 성능이 좋아졌지만 그것이 왜 그런 결과를 내는지는 알기 어려웠다. 이러한 것을 두고 딥러닝은 black box와 같다고 묘사를 하곤 한다. 그 안에서 무슨 일이 일어나는지 왜 그런 판단을 하는지 어렵기 때문이다. 이 어두컴컴한 상자에 빛을 내려준 기술 중 하나가 이 논문에서 제시한 **Class Activation Maps (CAM)**이다. 기존의 CNN에서 사용이 되는 모델은 Convolution Layer가 여러 겹 쌓여있고 마지막에 Fully-connected layer로 이어져 분류를 시행하게 된다. 반면, CAM 방법은 **Convolution-Global average pooling-Softmax의** 간단한 구조로 바꾸어 학습하게 된다. 이 방법을 사용하면 모델이 어떤 판단을 내려 output을 낼 때 어디에 집중하여 보았는지 볼 수 있기 때문에 Explainable 한 결과를 낼 수 있다.

## Main Contribution : Class Activation Maps (CAM)

논문의 가장 중요한 핵심은 CAM이다. 이 CAM이 이루어지는 방식은 상당히 간단하다. 먼저 위에서 설명한 바와 같이 Convolution층 바로 다음 Global Average Pooling(GAP)을 붙이고 softmax를 붙이는 모델 구조를 만든다.


## Weakly-supervised Object Localization

Table 1. Classification error on the ILSVRC validation set.

| Networks | top-1 val. error | top-5 val. error |
|---|---|---|
| VGGnet-GAP | 33.4 | 12.2 |
| GoogLeNet-GAP | 35.0 | 13.2 |
| AlexNet*-GAP | 44.9 | 20.9 |
| AlexNet-GAP | 51.1 | 26.3 |
| GoogLeNet | 31.9 | 11.3 |
| VGGnet | 31.2 | 11.4 |
| AlexNet | 42.6 | 19.5 |
| NIN | 41.9 | 19.6 |
| GoogLeNet-GMP | 35.6 | 13.9 |

Table 2. Localization error on the ILSVRC validation set. *Back-prop* refers to using [23] for localization instead of CAM.

| Method | top-1 val.error | top-5 val. error |
|---|---|---|
| GoogLeNet-GAP | **56.40** | **43.00** |
| VGGnet-GAP | 57.20 | 45.14 |
| GoogLeNet | 60.09 | 49.34 |
| AlexNet*-GAP | 63.75 | 49.53 |
| AlexNet-GAP | 67.19 | 52.16 |
| NIN | 65.47 | 54.19 |
| Backprop on GoogLeNet | 61.31 | 50.55 |
| Backprop on VGGnet | 61.12 | 51.46 |
| Backprop on AlexNet | 65.17 | 52.64 |
| GoogLeNet-GMP | 57.78 | 45.26 |

논문에서는 단지 Explainable한 모델을 제시한 것에서 그치는 것이 아니라 object localization을 한 실험도 제시한다. 먼저 CAM 방법으로 classification을 학습시킨 모델은 GAP를 사용하니 성능이 조금 저하되긴 했지만 경쟁력 있는 결과를 보여주었다. classification을 학습시킨 모델로 thresholding 방법을 사용하여 bounding box를 만드는 실험을 하였는데 놀랍게도 굉장히 좋은 성능을 보였다. table 2의 결과를 보면 기존의

bounding box를 annotation 하여 backpropagation 시킨 방법보다 더 좋은 성능을 보인다. 따로 학습을 하지 않았는데 이런 결과를 보여 재미있는 결과였다.

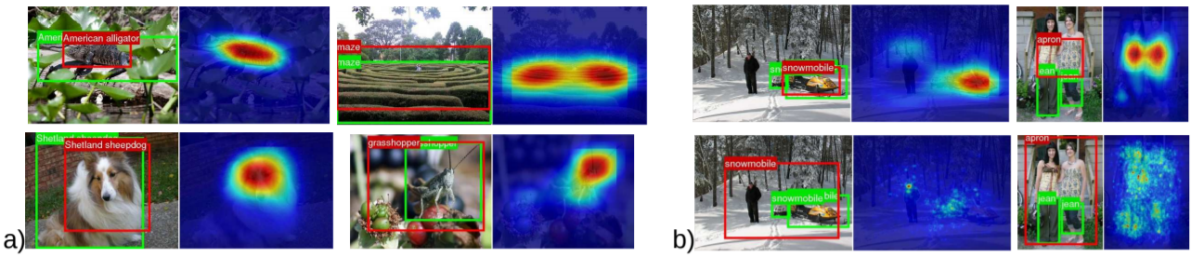



Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

## Experimental results

이 외에도 논문에서는 정말 많은 실험 결과들을 제시한다. Fine-grained Recognition으로 200종의 새를 인식하는 문제에서 full image를 thresholding으로 crop하니 더 성능이 잘 나온다는 결과, Pattern Discovery로 장면에서 informative한 물체들을 인식하는 것에서 어디를 보고 판단했는지 알 수 있다는 결과 등 흥미로운 결과들이 있었다. 또 visual question answering에서 predictor가 문제에 대한 답을 무엇을 보고 맞혔는지 나타내는 것도 흥미로웠다. 이에 대한 결과는 Figure 12에 나와있다.


Figure 12. Examples of highlighted image regions for the predicted answer class in the visual question answering.