3D hand pose estimation using ResNet-50

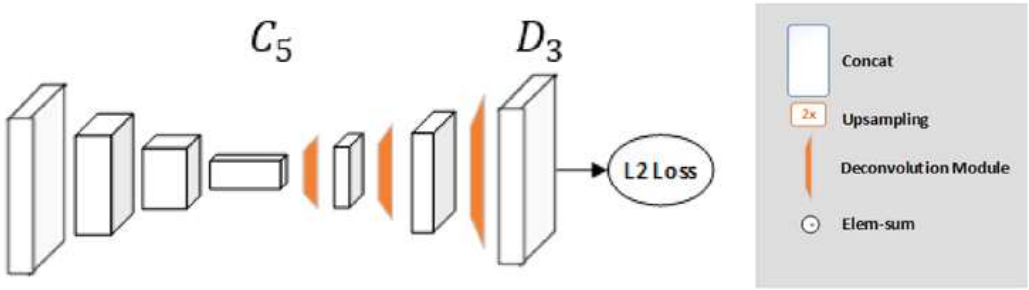


그림 1. SimpleBaseline using ResNet

Human Pose Estimation을 연구하던 마이크로소프트의 인턴 Haiping Wu는 “기술이 많이 발전했는데 현재의 간단한 모델은 얼마나 성능이 좋을까?”라는 생각으로 위와 같은 아주 간단한 encoder-decoder 구조를 설계

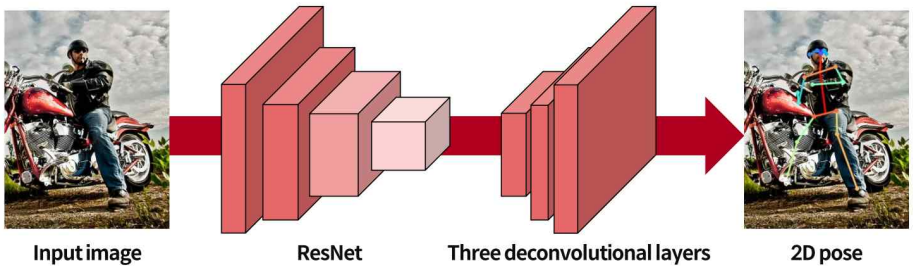


그림 2. ResNet으로 2d pose estimation

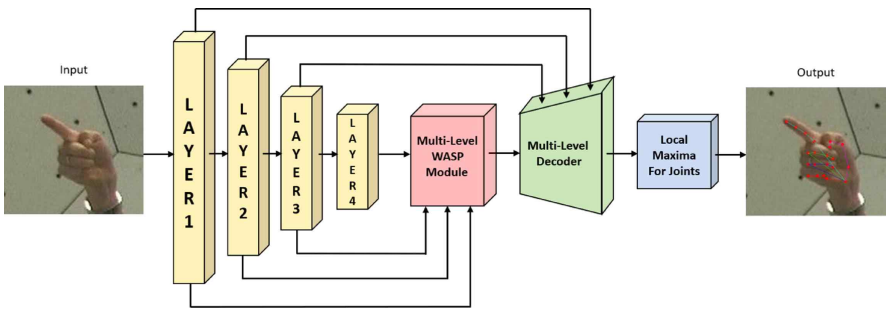


그림 3. HandyPose
https://www.sciencedirect.com/science/article/pii/S0031320322001558

1. **PoseResnet** ~ ResNet50 기반인가?

Pose ResNet을 설명하는 포스팅 글에 나와 있던 아래의 표가 바로 1 page, figure 1에서 설명한 실험 결과인데, ResNet-50, ResNet-101 그리고 ResNet-152만을 사용한 간단한 구조가 SOTA(state-of-the-art) 모델들보다 성능이 좋다는 것을 밝힘. 즉 ResNet-50 기반이라고 볼 수 있음

Method	Backbone	Input Size	OHKM	*AP*
8-stage Hourglass	-	256 × 192	✗	66.9
8-stage Hourglass	-	256 × 256	✗	67.1
CPN	ResNet-50	256 × 192	✗	68.6
CPN	ResNet-50	384 × 288	✗	70.6
CPN	ResNet-50	256 × 192	✓	69.4
CPN	ResNet-50	384 × 288	✓	71.6
Ours	ResNet-50	256 × 192	✗	70.4
Ours	ResNet-50	384 × 288	✗	72.2

Method	Backbone	Input Size	*AP*	AP_{50}	AP_{75}	AP_m	AP_l	*AR*
CMU-Pose [5]	-	-	61.8	84.9	67.5	57.1	68.2	66.5
Mask-RCNN [12]	ResNet-50-FPN	-	63.1	87.3	68.7	57.8	71.4	-
G-RMI [24]	ResNet-101	353 × 257	64.9	85.5	71.3	62.3	70.0	69.7
CPN [6]	ResNet-Inception	384 × 288	72.1	91.4	80.0	68.7	77.2	78.5
FAIR* [9]	ResNeXt-101-FPN	-	69.2	90.4	77.0	64.9	76.3	75.2
G-RMI* [9]	ResNet-152	353 × 257	71.0	87.9	77.7	69.0	75.2	75.8
oks* [9]	-	-	72.0	90.3	79.7	67.6	78.4	77.1
bangbangren*+ [9]	ResNet-101	-	72.8	89.4	79.6	68.6	**80.0**	78.7
CPN+ [6,9]	ResNet-Inception	384 × 288	73.0	**91.7**	80.9	69.5	78.1	**79.0**
Ours	ResNet-152	384 × 288	**73.7**	**91.9**	**81.1**	**70.3**	**80.0**	**79.0**

Base Code: https://github.com/Microsoft/human-pose-estimation.pytorch

2. 3d hand pose estimation으로 응용 개발이 가능한가?

관련 연구 1.

<< Large-scale Multiview 3D Hand Pose DataSet >>

(https://arxiv.org/pdf/1707.03742.pdf)

5.2. Hand Pose Regression

The hand pose is computed by a modified ResNet50. The ResNet50 deep learning architecture is currently the state of the art CNN on image recognition, achieving a top-1 error (22.85) on the ImageNet validation split. This architecture introduces the 'residual' term, which consists of the aggregation of the input image to the output image of a convolution block. As a result, the output of a convolution block can be seen as the input image where the features activated by the filters are highlighted. In contrast, the output of a convolution layer in a default convolutional neural network is only the result of the neuron activation. If a neuron is not triggered on a certain region of the input image, the output remains with lower activation values. When the network computes the weights update in the backpropagation stage, the values on non-activated regions lead to very low upgrades, eventually even resulting in no upgrade at all, which causes the learning to stall. This issue is known as the vanishing gradient problem. The inclusion of the 'residual' term helps fight the vanishing gradient problem and allows the design of even deeper architectures. Currently, the best performer on several tasks of the ImageNet challenge is based on the 'residual' approach introduced by ResNet.

관련 연구 2.

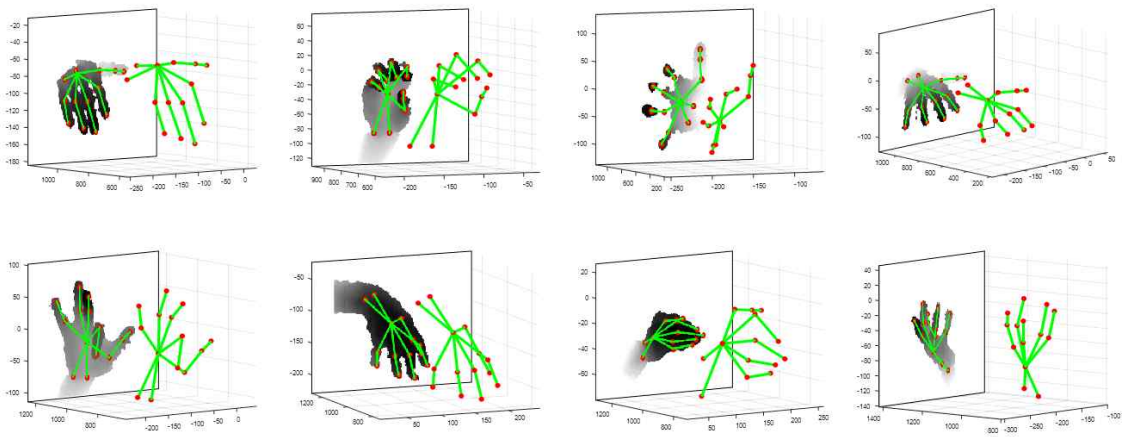
A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation from a Single Depth Image (https://arxiv.org/abs/1908.09999)

깊이 기반 이미지, 백본 네트워크로 ResNet-50을 사용하였고 이 코드를 GitHub에도 공개 중

GitHub

- https://github.com/zhangboshen/A2J
- https://github.com/zhangboshen/A2J/tree/master/src_train

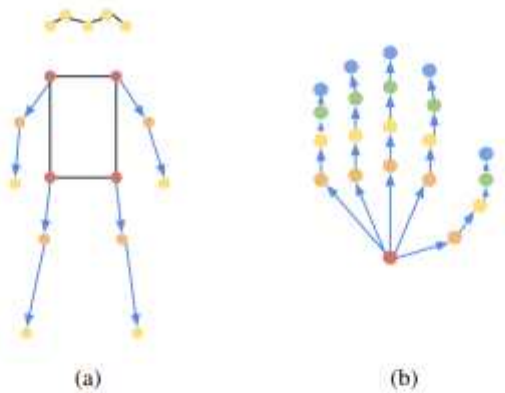
Dataset: NYU hand pose dataset(https://jonathantompson.github.io/NYU_Hand_Pose_Dataset.htm)



관련 연구 3.

<< **Analyzing and Diagnosing Pose Estimation with Attributions Supplementary Material** >> ~ Simple Baseline ResNet50을 사용해서 estimation

https://openaccess.thecvf.com/content/CVPR2023/supplemental/He_Analyzing_and_Diagnosing_CVPR_2023_supplemental.pdf



(a) (b)

Figure D. Groups of (a) human joints and (b) hand joints. The groups noted with color are (a) trunk joints (red), branch joints (orange), and leaf joints (yellow); (b) wrist (red), MCP (orange), PIP (yellow), DIP (green), and TIP (blue). The blue arrow indicates the relationship between the child and parent along the kinematic chain.

Joints	ResNet50	ResNet101	HRNet-W32	Transpose	Integral	RLE
trunk	1.961	1.906	1.849	1.863	1.865	1.754
branch	1.986	1.949	1.926	1.907	1.907	1.795
leaf	2.028	1.999	1.978	1.990	1.946	1.853

Table C. FI of different kinds of joints in each pose estimation model. Progressing down from the trunk joint to the leaf joint through the branch joint, the mean value of FI gets larger.

A.4. Model Randomization

We use Simple Baseline ResNet50 [22] as the example to conduct the model randomization test. Specifically, there are 54 convolutional layers in the model, and we successively randomize the parameters of these layers. After randomizing every nine layers, we compute the PoseIG attribution map of that randomized model. Apart from conducting the test qualitatively like previous methods [1], we also verify it quantitatively with the numerical indices. As Tab. B shows, the attribution maps have less FI and LI and higher DI, indicating that the attribution tends to change more with more corrupted parameters. Therefore, PoseIG is sensitive to the parameters in the model, so it can be used to diagnose the model. We also visualize this as shown in Fig. C.

3. 예상되는 문제점 및 대안

ResNet이 backbone이 되면 input image를 32x downsampling(ex: 256x256 → 8x8) → 저해상도 feature map을 초래하는 high-to-low resolution 네트워크로써 정확도가 떨어질 수 있음(대안: single person에 대해서만 estimation 할 때, 네트워크 내내 high resolution을 유지할 수 있는 HRNet-pose 활용)

4. 활용할 수 있을 것 같은 추가 자료

1. ResNet-based hand pose recognition edge computing project

https://www.hackster.io/aaronhuang/resnet-based-hand-pose-recognition-edge-computing-project-64fc7f

2. **mano hand model**: 변형 가능한 3d hand mesh model

- https://mano.is.tue.mpg.de/
- github: https://github.com/otaheri/MANO