COURSE PROJECT

A Survey Hand Detection and Finger Pose Estimation Applications in Human-Machine Communication

Students:

- Dương Thị An
- Phan Đình Anh Quân
- Phạm Gia Thông
- Hoàng Đức Nhật Minh

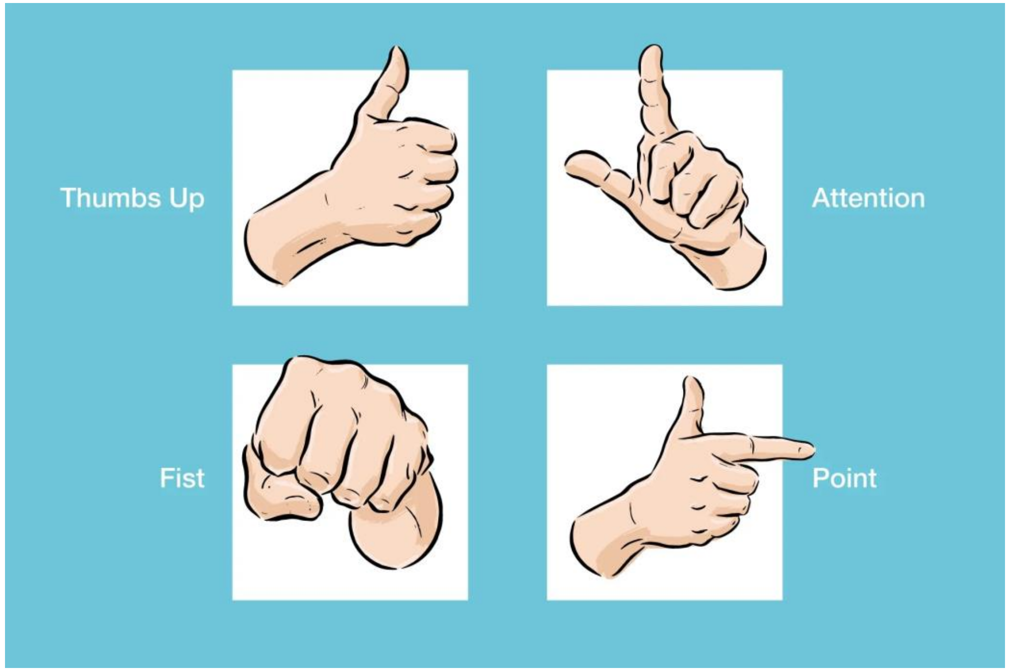
CONTENT

- Motivation
- Problem Statement
- Related Work

MOTIVATION

Motivation

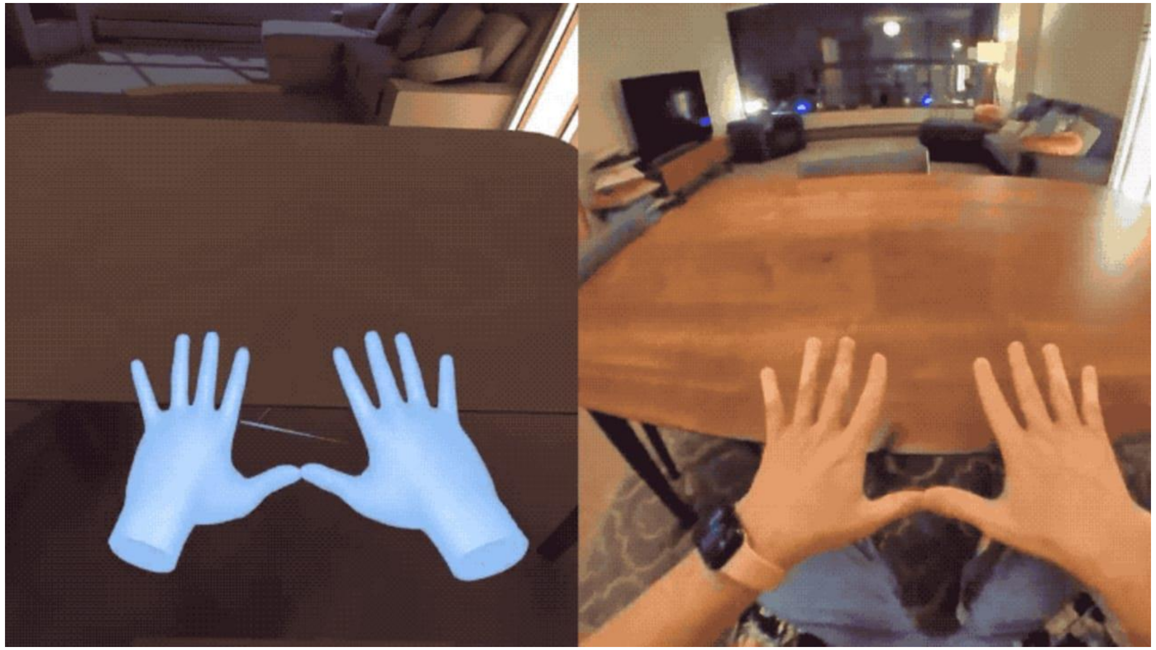
Humans often interact with objects or objects with their hands.
Hand gestures always appear in life.



Motivation

BASICS OF AMERICAN SIGN LANGUAGE
ASL is the primary language used by people in North America who are hard of hearing or deaf. It is a complete and natural language and shares linguistic properties with spoken languages.
HELLO
GOODBYE
PLEASE
THANK YOU
YES
NO
YOU'RE WELCOME
SORRY
I LOVE YOU
Sources: Dummies.com. handspeak.com. Graphic reporting by Noah Danesh, Daily Bruin contributor. Graphic by Farrah Au-Yeung, assistant Graphics editor.

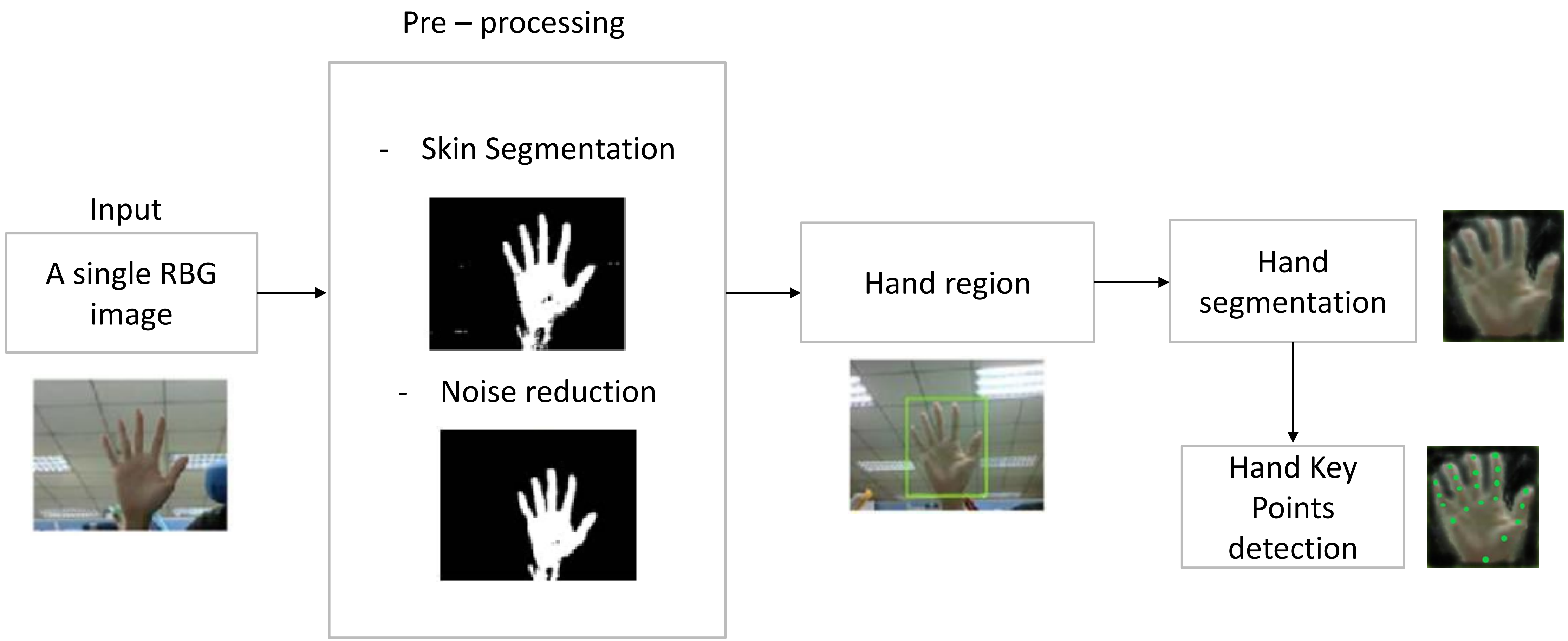
uture
bot and human
ector illustration



PROBLEM STATEMENT

Problem Statement

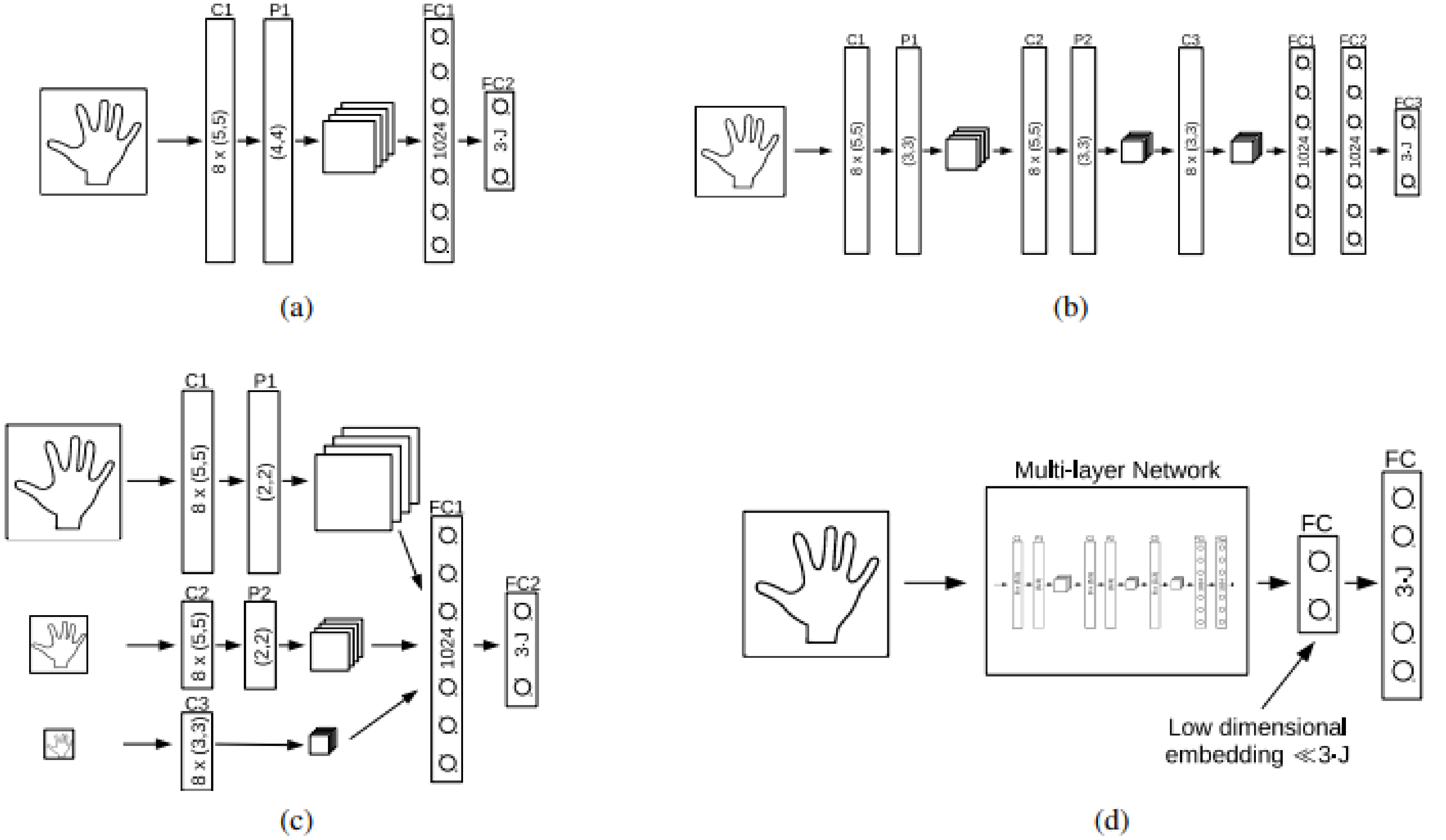
The problem is to detect the hand, the anchor point on the hand from that result predicts the hand gesture.



RELATED WORK

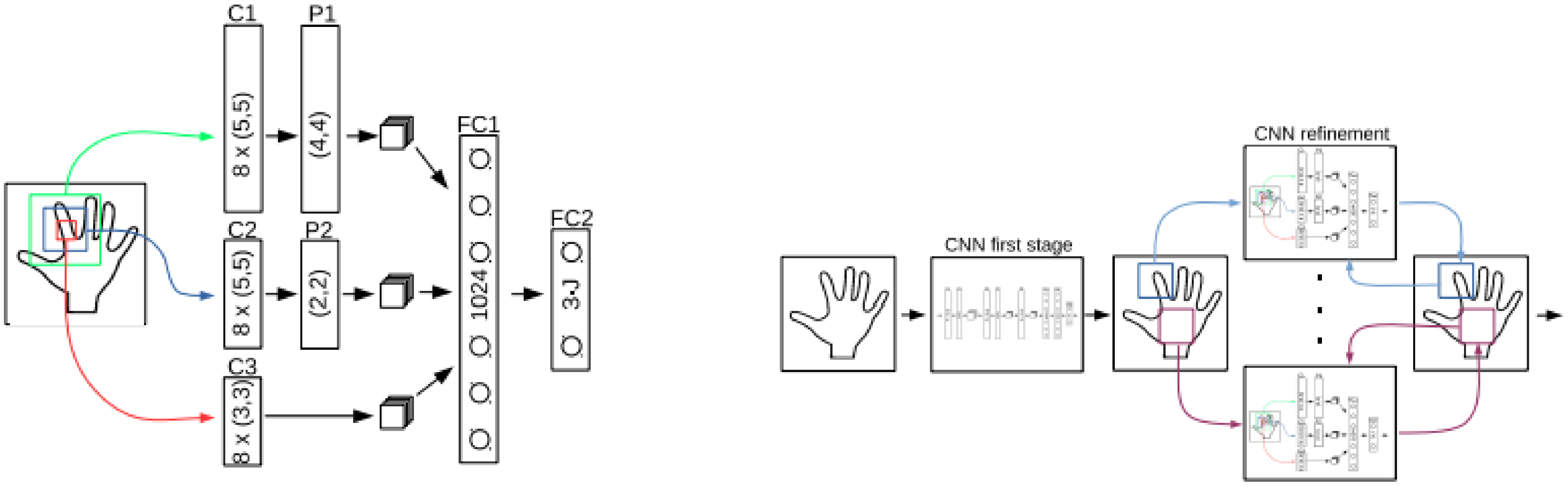
Hands Deep in Deep Learning for Hand Pose Estimation

Hand Detection



Hands Deep in Deep Learning for Hand Pose Estimation

Predict the position of the joints.



(a) (b)

Hands Deep in Deep Learning for Hand Pose Estimation

Metric:

Euclide distance between the predicted result that fits the hand and the actual measurement result.

Pros:

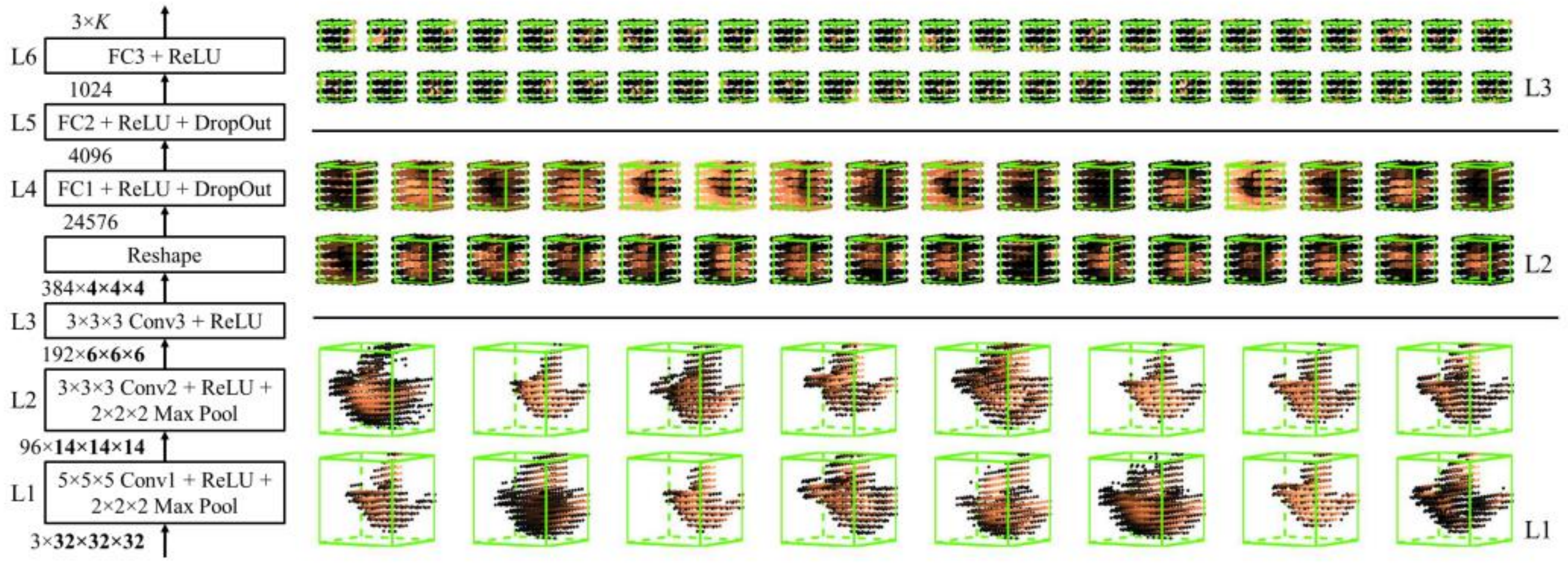
The execution time is fast and the accuracy is high.

Cons:

Require labeled dataset and 3D locations for training.

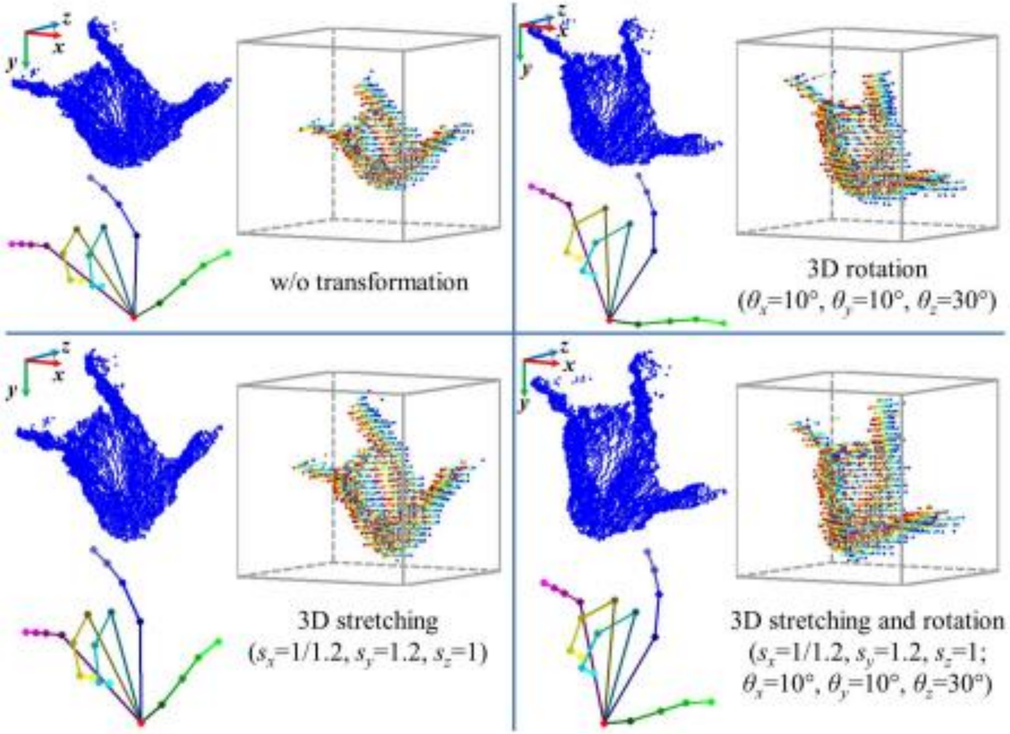
When the pixel in the image is not available due to the brightness effect, the result will be inaccurate.

3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation f-rom Single Depth Images

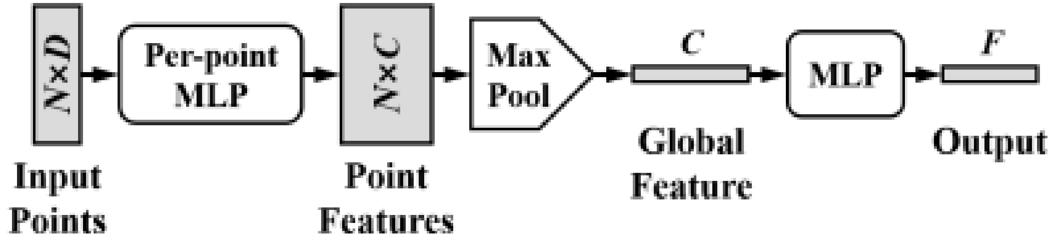


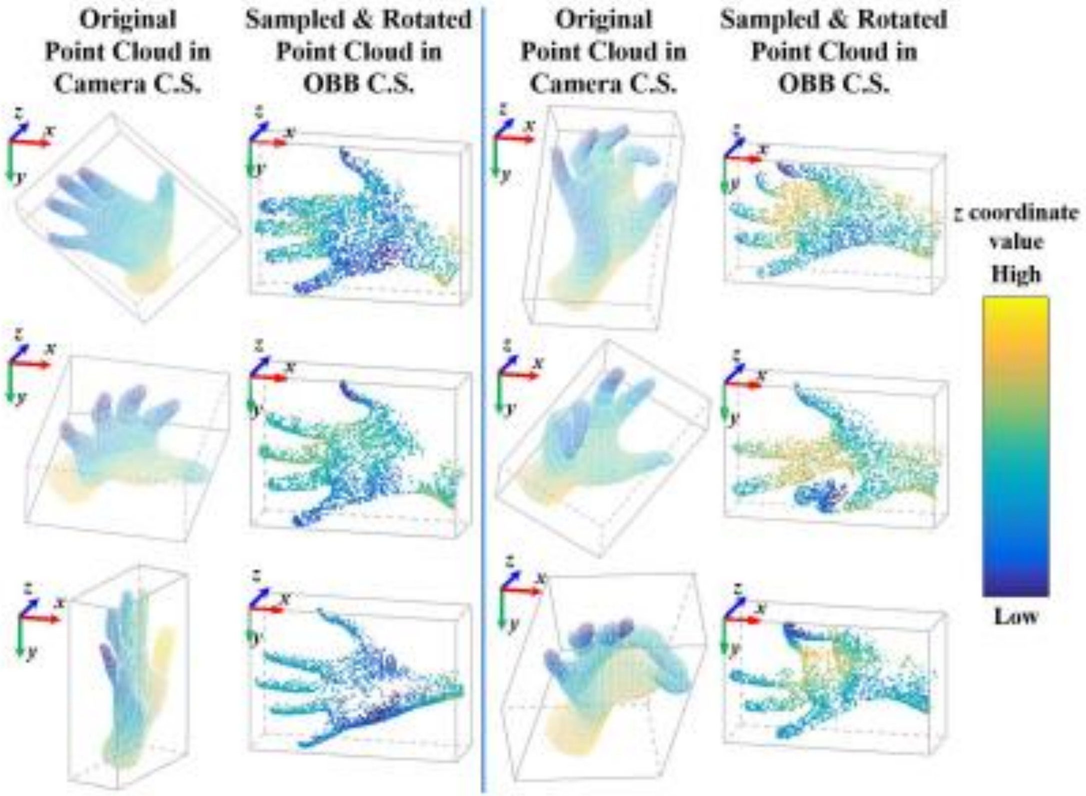
(a) (b)

3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation f-rom Single Depth Images

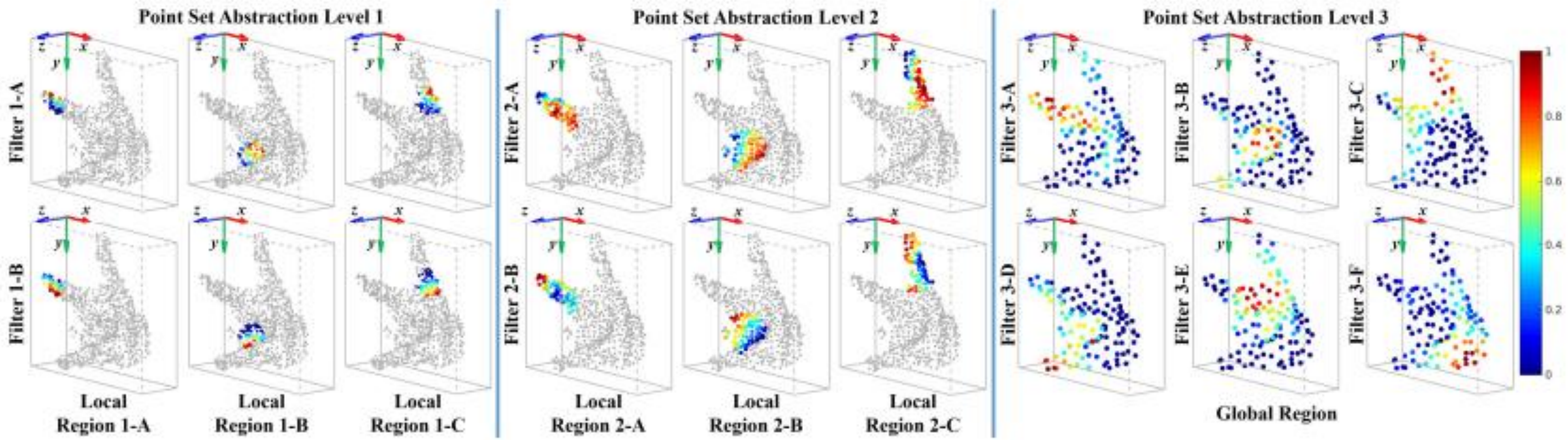


Hand PointNet: 3D Hand Pose Estimation using Point Sets

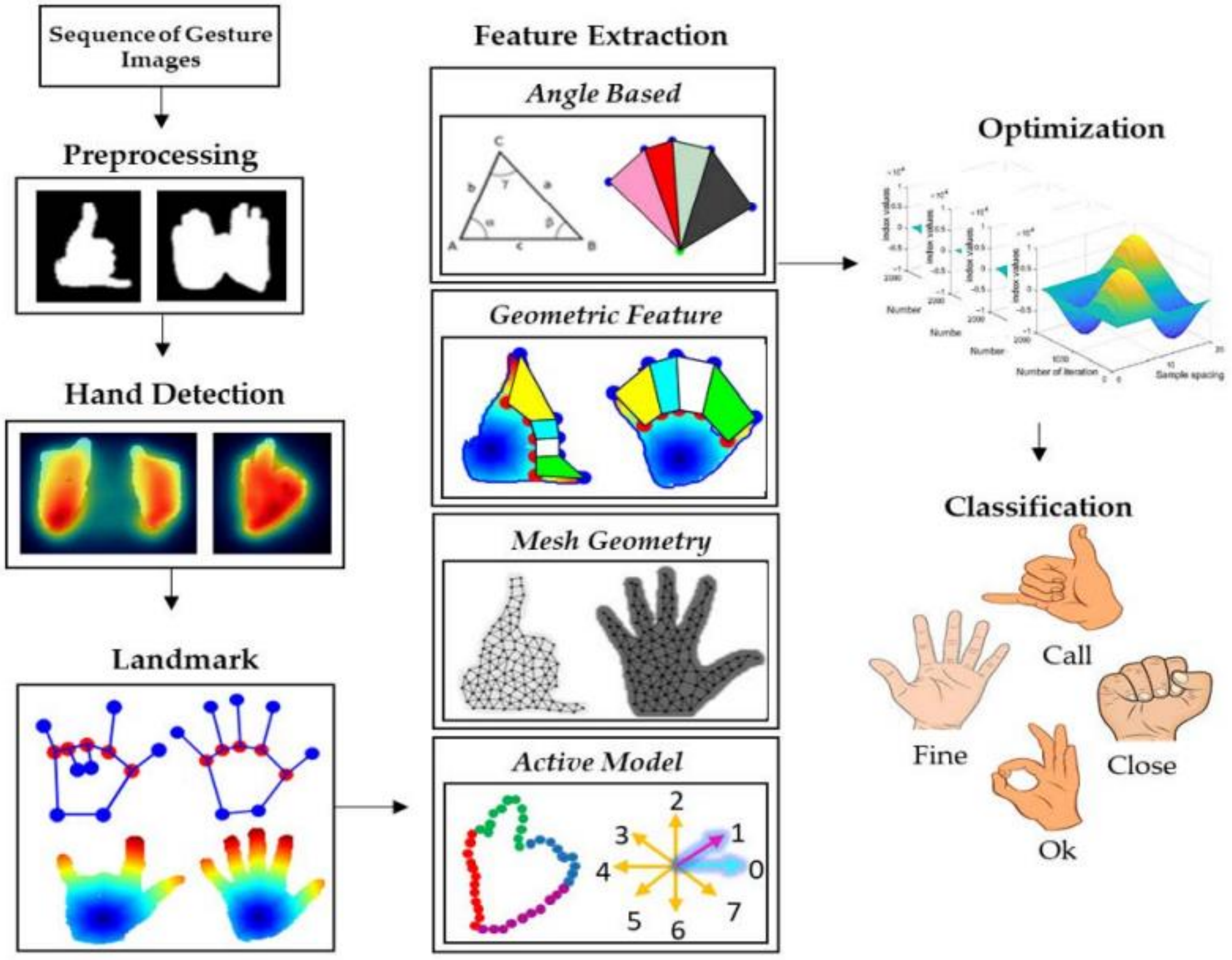




Hand PointNet: 3D Hand Pose Estimation using Point Sets



Hand Gesture Recognition Based on Auto-Landmark Localization and Reweighted Genetic Algorithm for Healthcare Muscle Activities



Paper	Year	Method	Metric	Pros	Cons
Hands Deep in Deep Learning for Hand Pose Estimation	2016	Using LRF to detect the hand. Using the CNN model to predict the position of the knuckles.	Euclidean distance between the predicted result of the knuckle and the actual measurement	The execution time is fast and the accuracy is high.	Require labeled dataset and 3D locations for training. When the pixel in the image is not available due to the brightness effect, the result will be inaccurate.

Paper	Year	Method	Metric	Pros	Cons
3D Convolut-ional Neural Networks for Efficient and Robust Hand Pose Estimation f-rom Single Depth Images	2017	Create cubes 3D hand representations fro m deep imageUsing 3layer network architecture convol utional 3D connected. Enhance 3D data by rotating and stretching point clouds hands in 3D space.	Error Euclide Distance	Get quick results in real time. Applicable on multiple data variations with different hand sizes thanks to 3D data enhancement on the training dataset.	High computational complexity depends on the resolution.

Paper	Year	Method	Metric	Pros	Cons
Hand PointNet: 3D Hand Pose Estimation using Point Sets	2018	Extract distinguishing features using PointNet OBB-based point cloud normalization Construction of hand posture regression network Re-screen the fingertips.	Error Euclide Distance	Get quick results in real time. Applicable on multiple data variations with different hand sizes thanks to 3D data enhancement on the training dataset.	High computational complexity depends on the resolution.

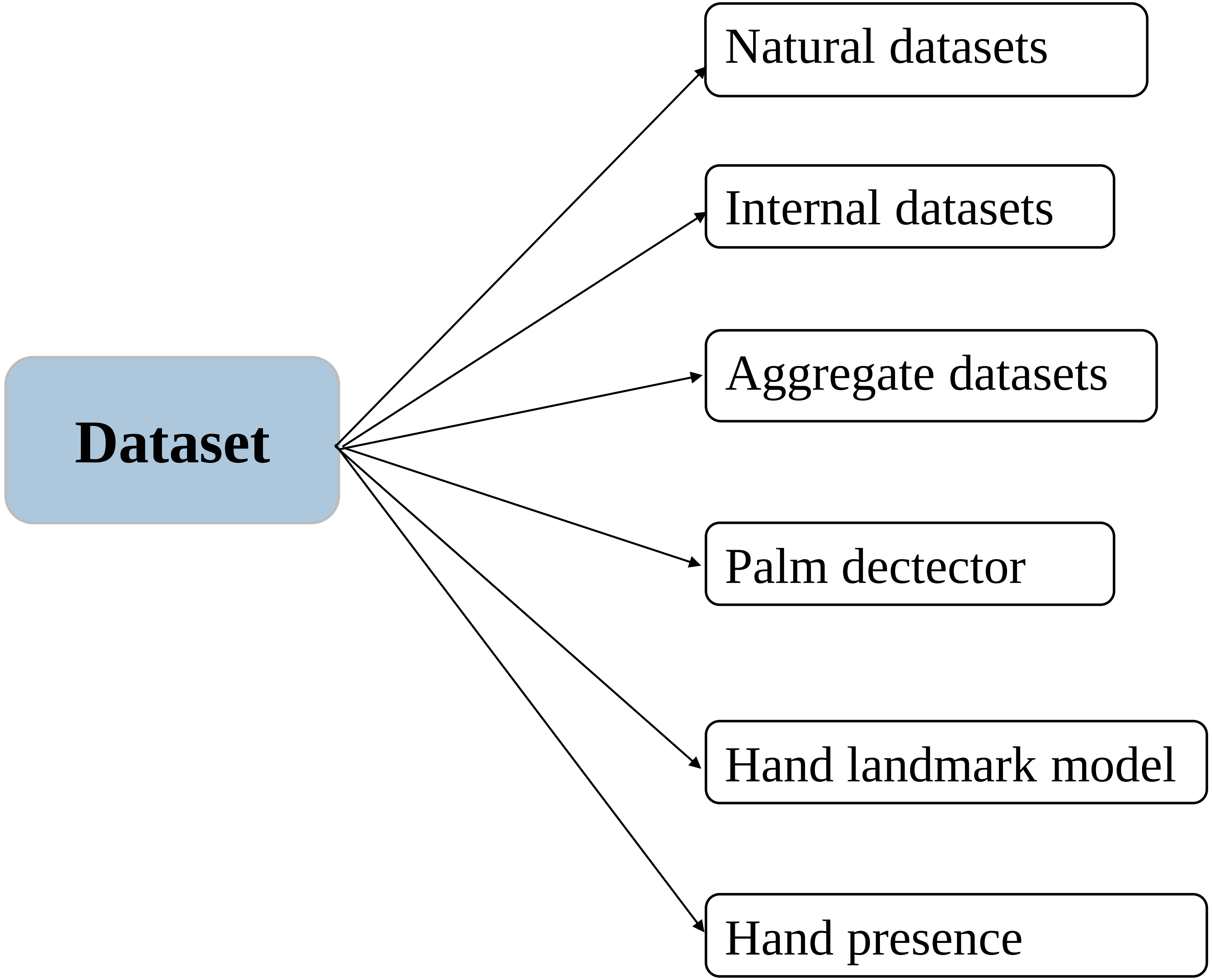
Paper	Year	Method	Metric	Pros	Cons
The joint locations of the hand pose using deep networks	2019	Depth-based hand posture estimation through CNN neural network and Resnet50 model.	Measure on available data sets, containing hand frames, finger gestures to assess the accuracy of the method. ICVL Dataset MSRA Dataset NYU dataset	Avoid lost 3D spatial information in 2D heatmap Encode the point cloud in 3D representing the volume of the hand and use the 3D CNN for direct regression of the 3D hand posture	High computational complexity depends on resolution.

Paper	Year	Method	Metric	Pros	Cons
Fast Monocular Hand Pose Estimation on Embedded Systems	2021	MobileNet-SSD model hand detection Prediction of hand joint position using Hand Landmark Localization model	Sum squared error (SSE), endpoint error (EPE) and probability of correct score (PCK) within the normalized distance threshold.	Avoid lost 3D spatial information in 2D heatmap Encode the point cloud in 3D representing the volume of the hand and use the 3D CNN for direct regression of the 3D hand posture	Requires good and stable equipment.

MEDIAPIPE APPROACH

CONTENT:

Dataset



Dataset
Natural datasets
Internal datasets
Aggregate datasets
Palm dectector
Hand landmark model
Hand presence

Dataset

Natural datasets

- Contains 6K images that vary in geography, lighting conditions, and hand appearance
- Does not contain complexities at the joints of the hand

Dataset

Internal datasets

Contains 10 thousand images covering various angles of all possible hand gestures

The data set was collected from only 30 people with limited changes in hand gestures

Dataset

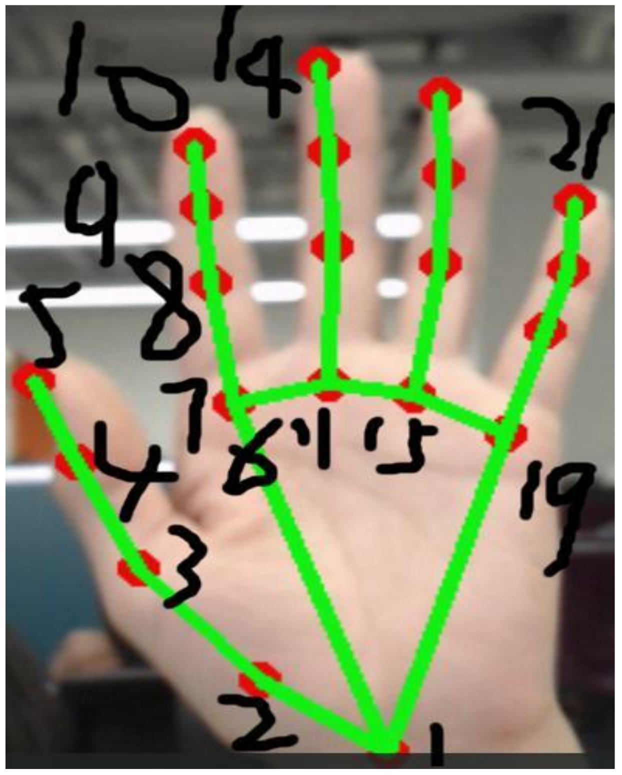
Aggregate datasets

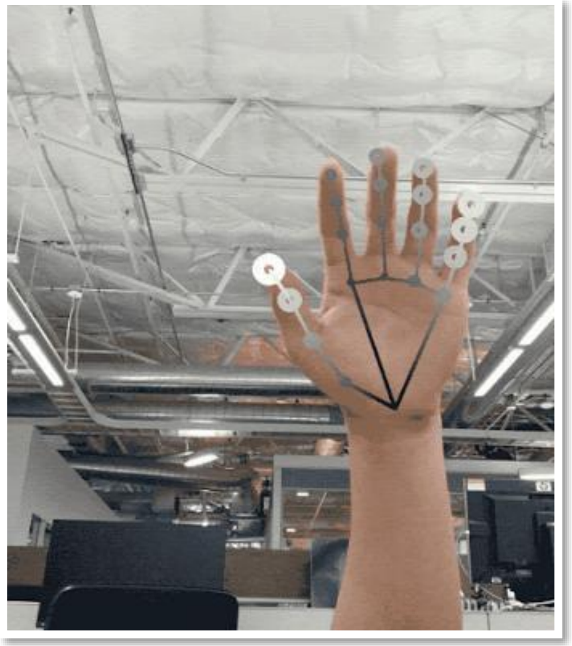
- Generate a high-quality composite hand model over various hand gestures and map it to the corresponding 3D coordinates
- Using 3D hand model equipped with 24 bones and including 36 blend shapes, control the thickness of fingers and palm
- The model also offers 5 textures with different skin tones
- Create video sequences that switch between hand poses and sample 100K images from the video

Dataset

Palm Dectector

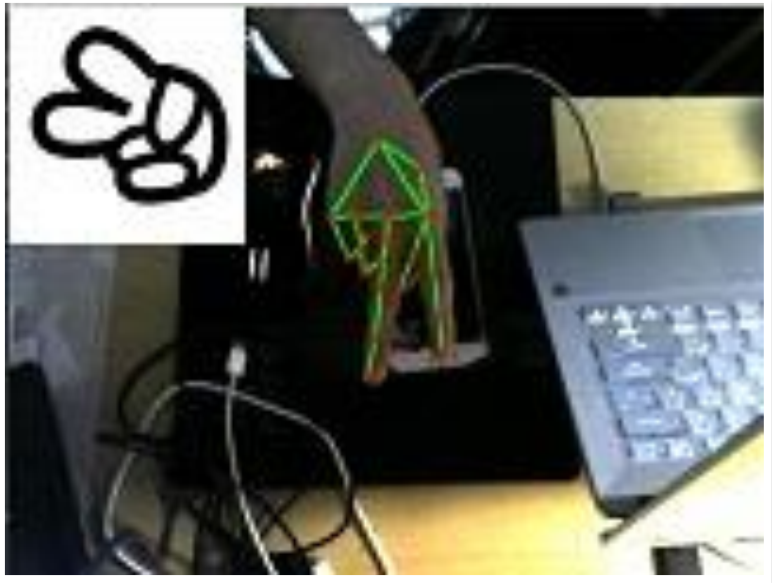
Use only actual data sets





Localize the hand and provide the highest variety of forms

Dataset



Hand Landmark Model

Annotate real-world images with 21 landmarks

Use projected groundtruth 3D joints for composite images

Dataset

Hand Presence

- Choose a subset of real-world images as a positive example
- Samples on the area that do not include the hand area are annotated as negative examples

Đối với thuận tay, chúng tôi chú thích một tập hợp con của hình ảnh trong thế giới thực với sự thuận tay để cung cấp dữ liệu đó.

MediaPipe Hands

Result

The combination of synthetic and natural (real-world) datasets gives the best results for the hand landmark model

In addition to quality improvement, training with a large composite dataset results in less visual jitter across frames.

Result

Dataset	MSE normalized by palm size
Only real-world	16.1%
Only synthetic	25.7%
Combined	13.4%

evaluate only on pictures in the real world

Model	Params (M)	MSE	Time(ms) Pixel 3
Light	1	11.83	6.6
Full	1.98	10.05	16.1
Heavy	4.02	9.817	36.9

performance trade-off between quality and speed

Architecture

Our hand tracking solution utilizes an ML pipeline consisting of two models working together:

• A palm detector that operates on a full input image and locates palms via an oriented hand bounding box.

• A hand landmark model that operates on the cropped hand bounding box provided by the palm detector and returns high-fidelity 2.5D landmarks.

Palm Detection Model

Single Shot Detector (SSD)

SSD is a type of object detection algorithm that can be used for palm detection. SSD only needs an input image and ground truth boxes for each object during training.

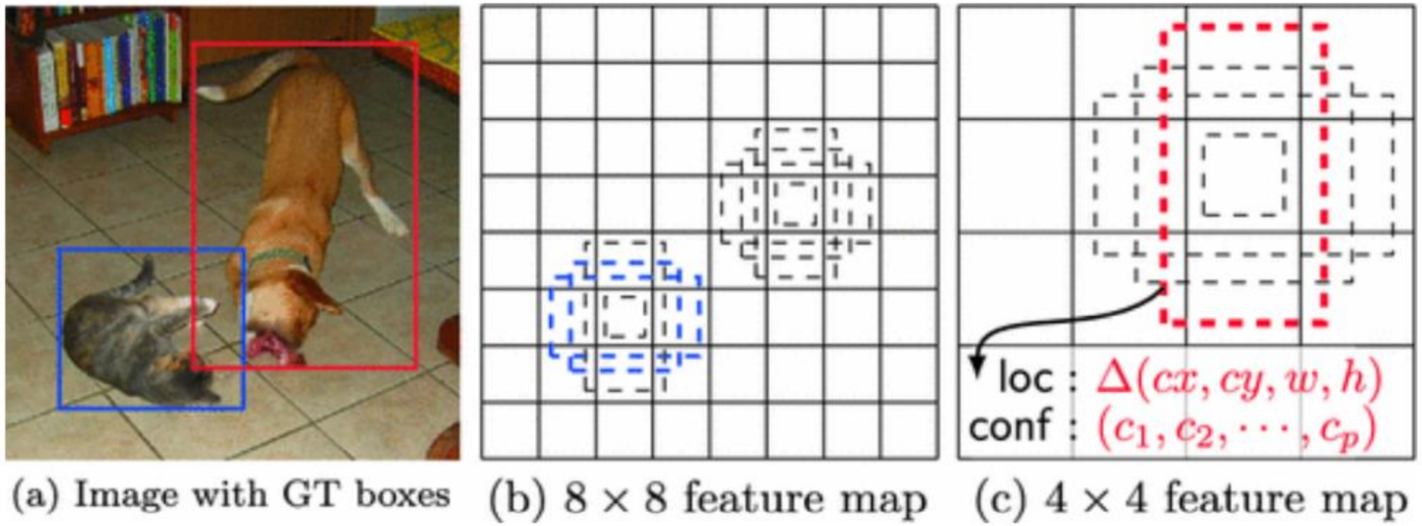


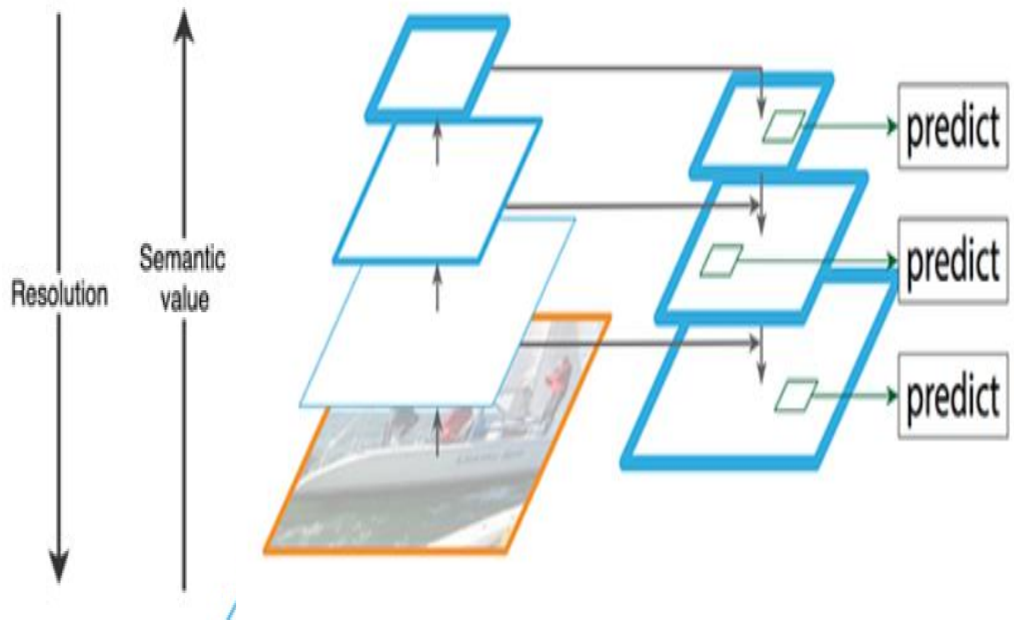
Figure 2.7: Single Shot Detector (SSD) framework, loc is location of the bounding box, conf is the confidence of all object categories, Figure from [16].

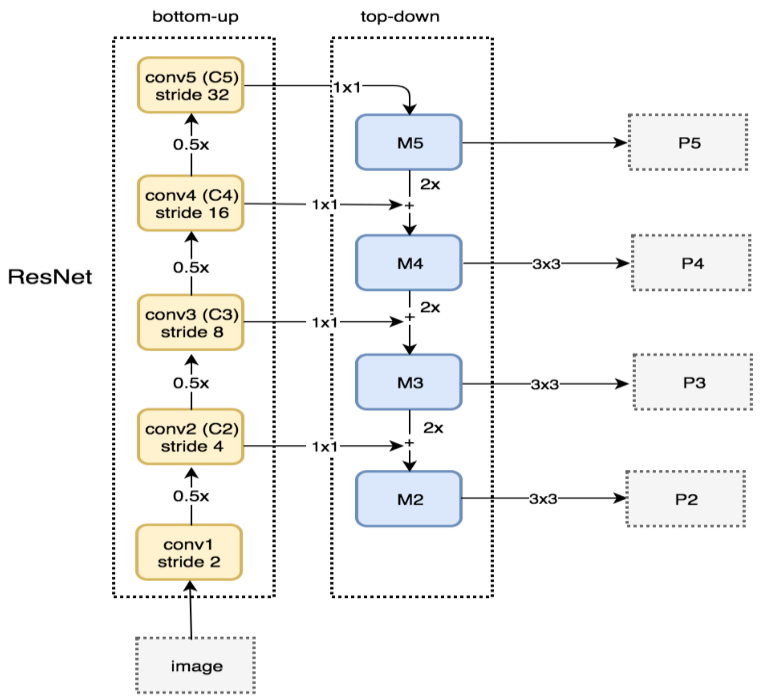
Palm Detection Model

Feature Pyramid Network (FPN)

FPN is a feature extractor designed for such pyramid concept with accuracy and speed in mind.

FPN composes of a **bottom-up** and a **top-down** pathway. The bottom-up pathway is the usual convolutional network for feature extraction. The top-down pathway to construct higher resolution layers from a semantic rich layer.





Palm Detection Model

Minimising focal loss

Focal loss addresses the problem of one stage detector where there is an imbalance between the foreground mage and the background image.

- CE is cross entropy loss:

$$CE(p_t) = -\alpha \log(p_t)$$

- FL is focal loss:

$$FL(p_t) = -(1 - p_t)^y \log(p_t)$$

- FL when using alpha variant focal loss:

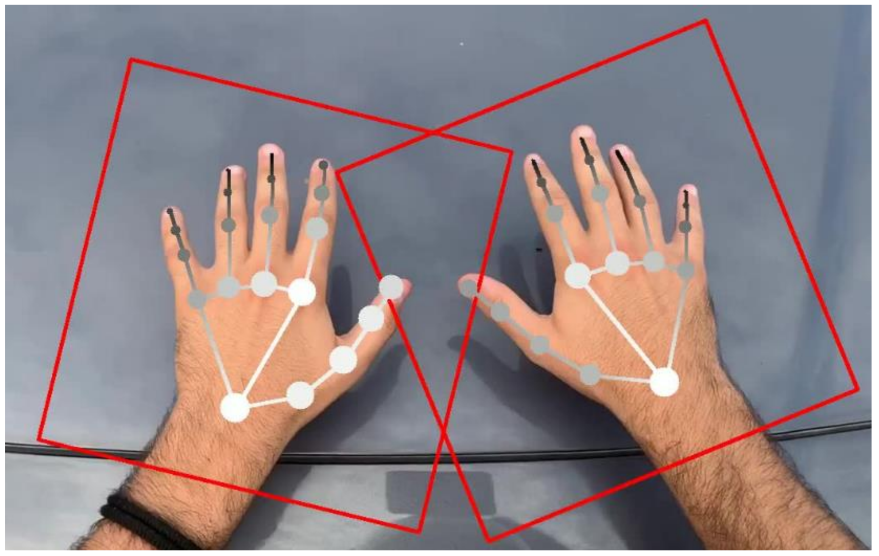
$$FL(p_t) = -\alpha_t(1 - p_t)^y \log(p_t)$$

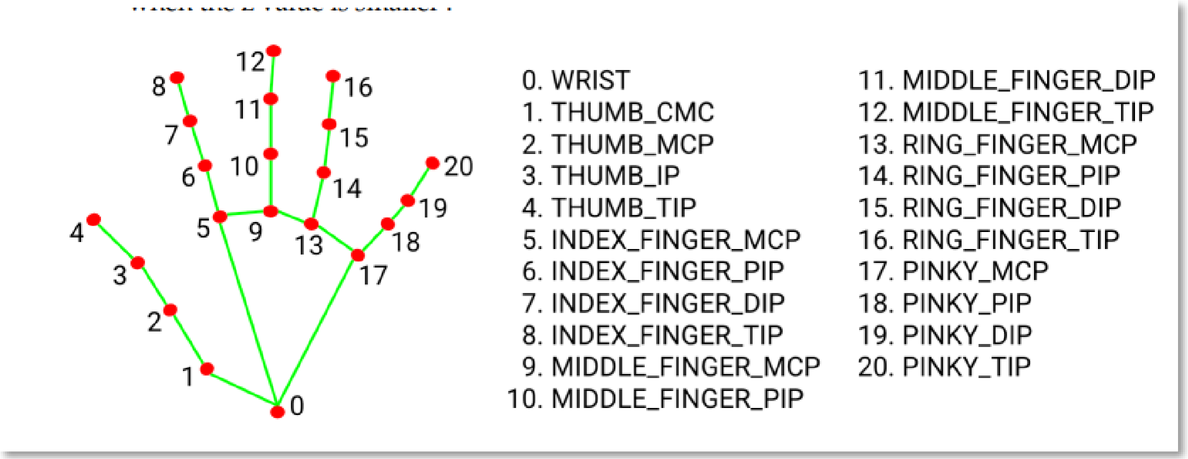
Hand land mark model

Input: The image contains the bounding box of the hand

Output:

1. The image contains 21 anchor points drawn on the hand
2. A hand flag indicating the probability of hand presence in the input image
3. A binary classification of handedness, e.g. left or right hand

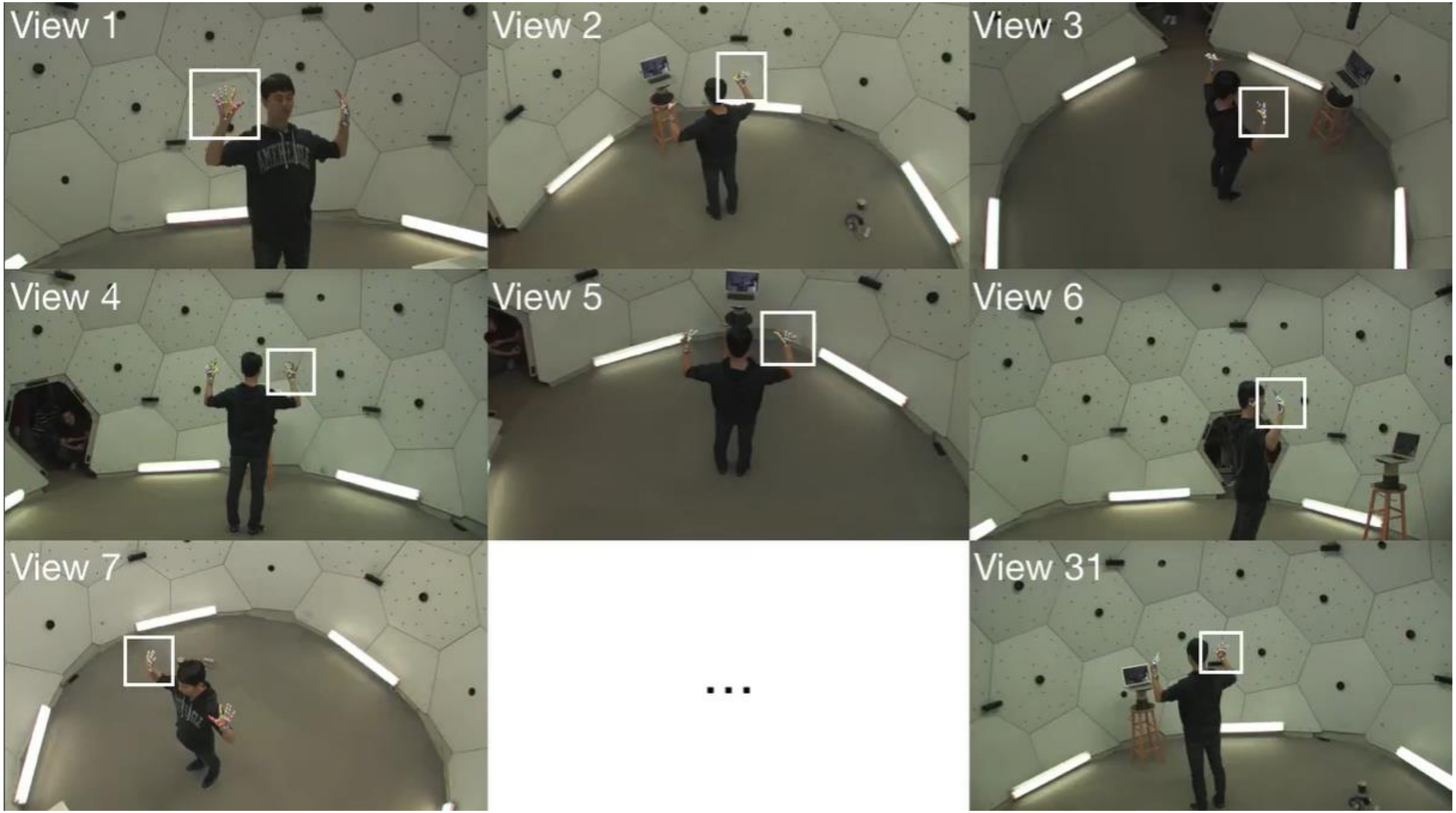




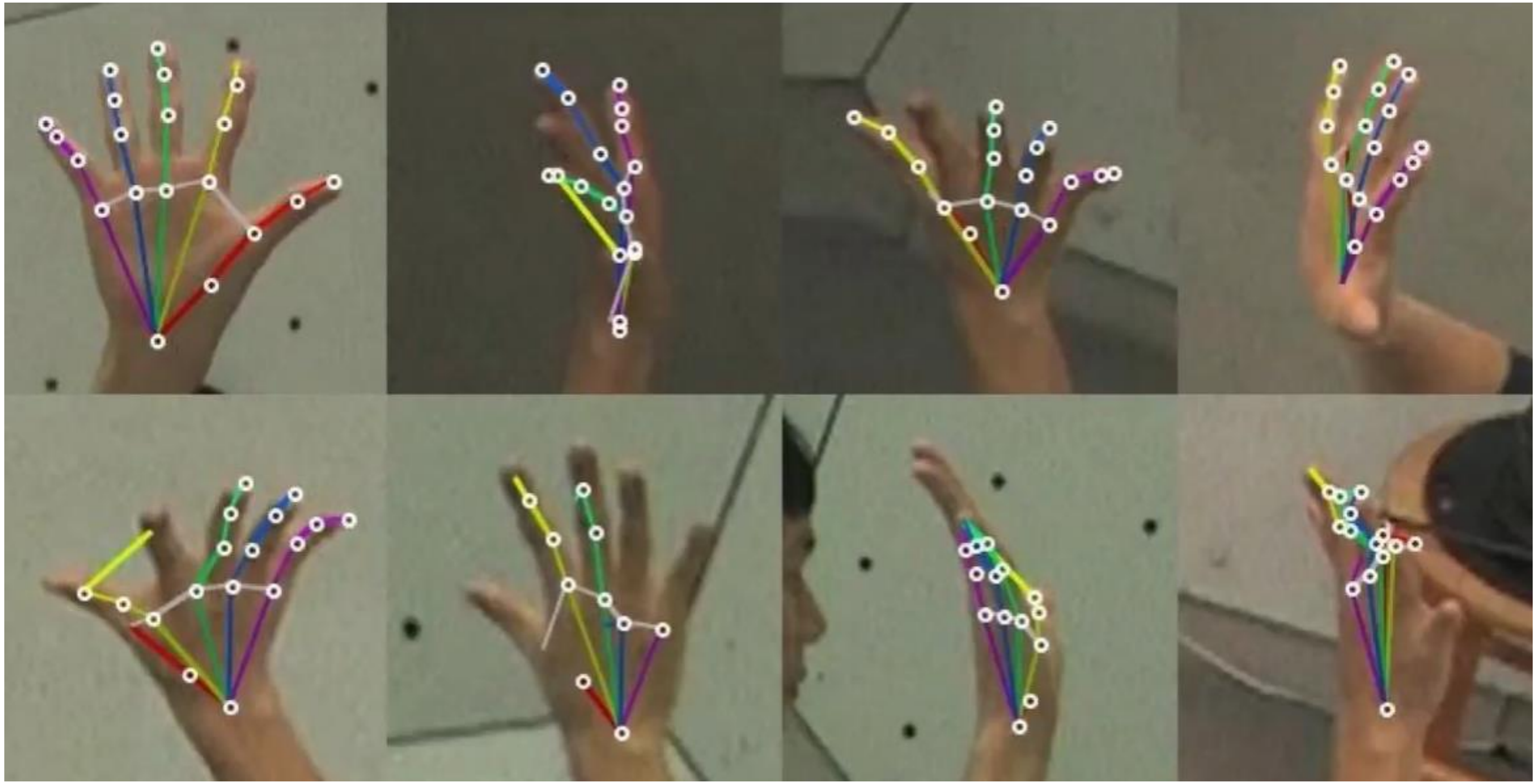
Architecture of hand landmark model for MediaPipe

1. **Multiview Bootstrapping**
2. **Recover tracking failure**

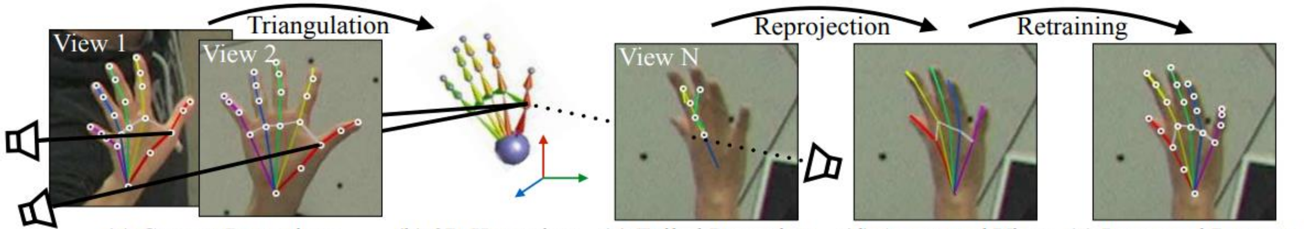
Idea behind multiview Bootstrapping



Idea behind multiview Bootstrapping



Multiview Bootstrapping algorithm



(a) Correct Detections (b) 3D Keypoints (c) Failed Detection (d) Annotated View (e) Improved Detector

Algorithm 1 Multiview Bootstrapping

Inputs:

- Unlabeled images: $\{\mathbf{I}_v^f \text{ for } v \in \text{views}, f \in \text{frames}\}$
- Keypoint detector: $d_0(\mathbf{I}) \mapsto \{(\mathbf{x}_p, c_p) \text{ for } p \in \text{points}\}$
- Labeled training data: $\mathcal{T}_0$

for iteration i in 0 to K:

1. Triangulate keypoints from weak detections

   for every frame f:

   (a) Run detector $d_i(\mathbf{I}_v^f)$ on all views v (Eq. (5))

   (b) Robustly triangulate keypoints (Eq. (6))

2. Score and sort triangulated frames (Eq. (7))
3. Retrain with N-best reprojections (Eq. (8))

   $d_{i+1} \leftarrow \text{train}(\mathcal{T}_0 \cup \mathcal{T}_{i+1})$

Outputs: Improved detector $d_K(\cdot)$ and training set $\mathcal{T}_K$

Triangulating Keypoints from Weak Detections

Given V views of an object in a particular frame f, we run the current detector d_i (trained on set $\mathcal{T}_i$) on each image $\mathbf{I}_v^f$, yielding a set $\mathcal{D}$ of 2D location candidates:

$$\mathcal{D} \leftarrow \{d_i(\mathbf{I}_v^f) \text{ for } v \in [1 \ldots V]\}. \tag{5}$$

For each keypoint p, we have V detections $(\mathbf{x}_p^v, c_p^v)$, where $\mathbf{x}_p^v$ is the detected location of point p in view v and $c_p^v \in [0, 1]$ is a confidence measure (we omit the frame index for clarity). To robustly triangulate each point p into a

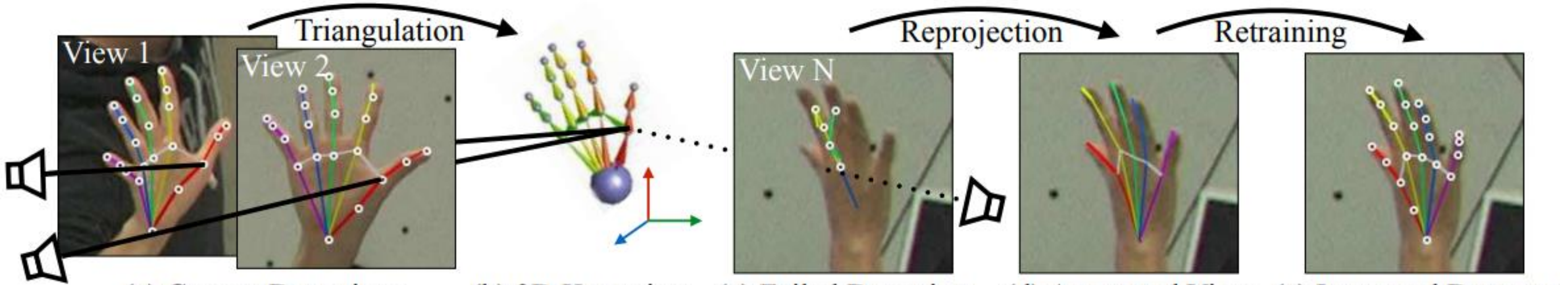
points (because the entire finger needs to be correct in the same view) but it further reduces the number of false positives, which is more important so that we do not train with incorrect labels.

Algorithm 1 Multiview Bootstrapping

Inputs:
- Unlabeled images: $\{\mathbf{I}_v^f$ for $v \in$ views, $f \in$ frames$\}$
- Keypoint detector: $d_0(\mathbf{I}) \mapsto \{(\mathbf{x}_p, c_p)$ for $p \in$ points$\}$
- Labeled training data: $\mathcal{T}_0$

for iteration i in 0 to K:
1. Triangulate keypoints from weak detections
   for every frame f:
   (a) Run detector $d_i(\mathbf{I}_v^f)$ on all views v (Eq. (5))
   (b) Robustly triangulate keypoints (Eq. (6))
2. Score and sort triangulated frames (Eq. (7))
3. Retrain with N-best reprojections (Eq. (8))
   $d_{i+1} \leftarrow \text{train}(\mathcal{T}_0 \cup \mathcal{T}_{i+1})$

Outputs: Improved detector $d_K(\cdot)$ and training set $\mathcal{T}_K$



(a) Correct Detections (b) 3D Keypoints (c) Failed Detection (d) Annotated View (e) Improved Detector

Scoring and Sorting Triangulated Frames

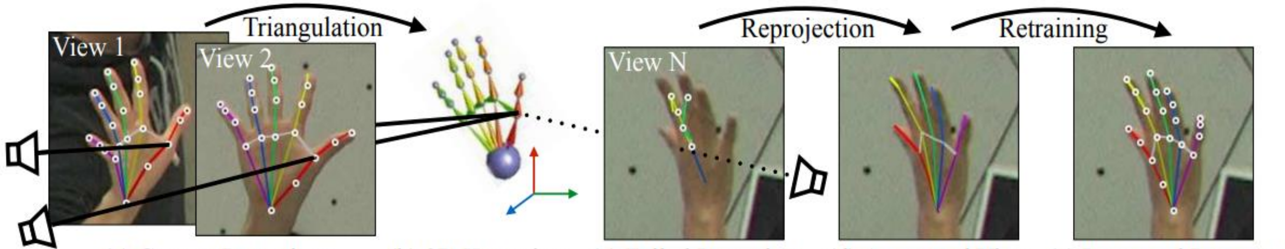
$$\text{score}(\{\mathbf{X}_p^f\}) = \sum_{p \in [1 \dots P]} \sum_{v \in \mathcal{I}_p^f} c_p^v. \tag{7}$$

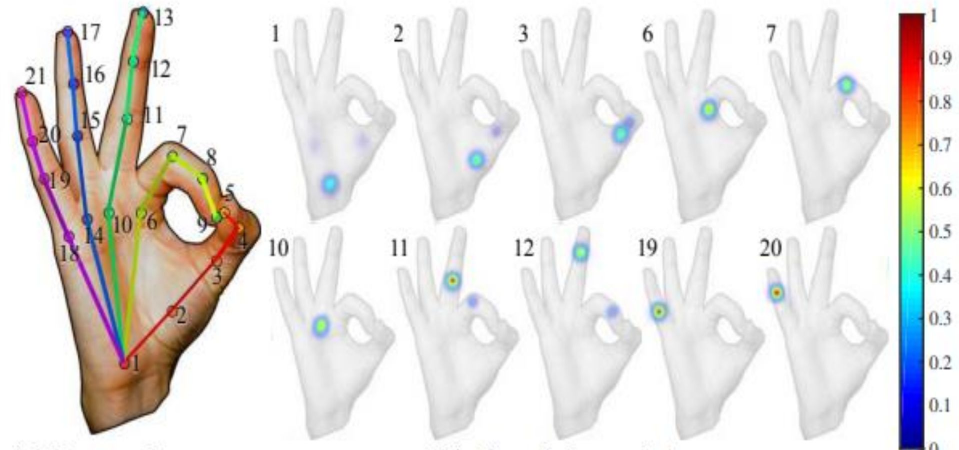
c_p^v ∈ [0, 1] is a confidence measure

Retraining with N-best Reprojections

We use the N-best frames according to this order to define a new set of training image-keypoint pairs for the next iteration i+1 detector

$$\mathcal{T}_{i+1} = \Big\{ \big(\mathbf{I}_v^{s_n}, \{ \mathcal{P}_v(\mathbf{X}_p^{s_n}) : v \in [1 \ldots V], \, p \in [1 \ldots P] \} \big) \text{ for } n \in [1 \ldots N] \Big\}, \quad (8)$$





(a) Detections (b) Confidence Maps

Figure 4: (a) Input image with 21 detected keypoints. (b) Selected confidence maps produced by our detector, visualized as a "jet" colormap overlaid on the input.

$\mathcal{P}_v(\mathbf{X}_p^{s_n})$: Projection of point p for frame index s_n into view v. Finally, we train a new detector using the expanded training set as $d_{i+1} \leftarrow \text{train}(\mathcal{T}_0 \cup \mathcal{T}_{i+1})$.

References

[1]https://arxiv.org/pdf/2006.10214v1.pdf

[2]Tomas Simon, Hanbyul Joo, Iain A. Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. CoRR, abs/1704.07809, 2017.

[3]https://www.youtube.com/watch?v=EgjwKM3KzGU&t=1690s

[4]https://github.com/nicknochnack/AdvancedHandPoseWithMediaPipe

[5]https://mediatum.ub.tum.de/doc/1658161/cyze4r5r5ptb06q0cb6rdi24h.pdf

[6]https://www.frontiersin.org/articles/10.3389/frai.2022.759255/full#B18

[7]https://openaccess.thecvf.com/content_ICCV_2019/papers/Chen_SO-HandNet_Self-Organizing_Network_for_3D_Hand_Pose_Estimation_With_Semi-Supervised_ICCV_2019_paper.pdf

[8]https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxnZWxpdWhhb250dXxneDoxMGU4YzMyMzNhMzM2NDUz

[9] Hand Keypint Detection in Single Images using Multiview Bootstrapping - YouTube

Thanks for watching and listening