H574160: Control Telerobotic Arm Remotely Via Pose Detection Of Human Arm

Ng Wei Jie Brandon A0184893L

Agenda

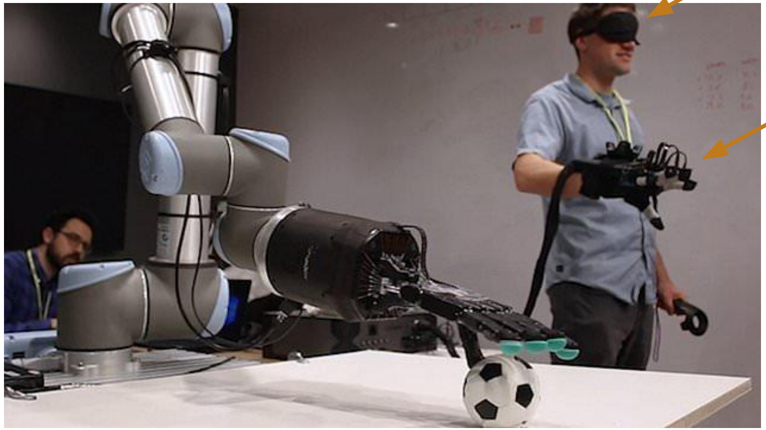
1. Introduction
2. Related Works
3. Datasets
4. Training Pipeline
5. Models
6. Results
7. Demo
8. Conclusion

Introduction

Motivation

Virtual Reality

Haptic Gloves



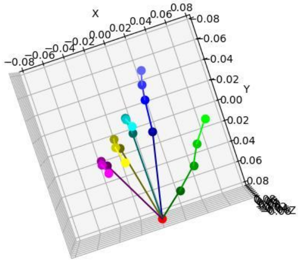
Accessible?

Contactless?

Seemingless?

Contributions

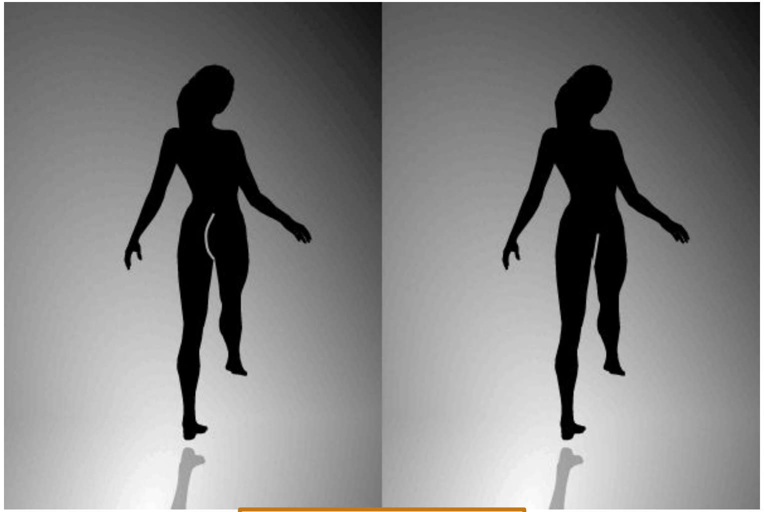
Demo pick and place



Design and train models

Collect upper body poses

Challenge



Temporal Info?

Multiview Info?

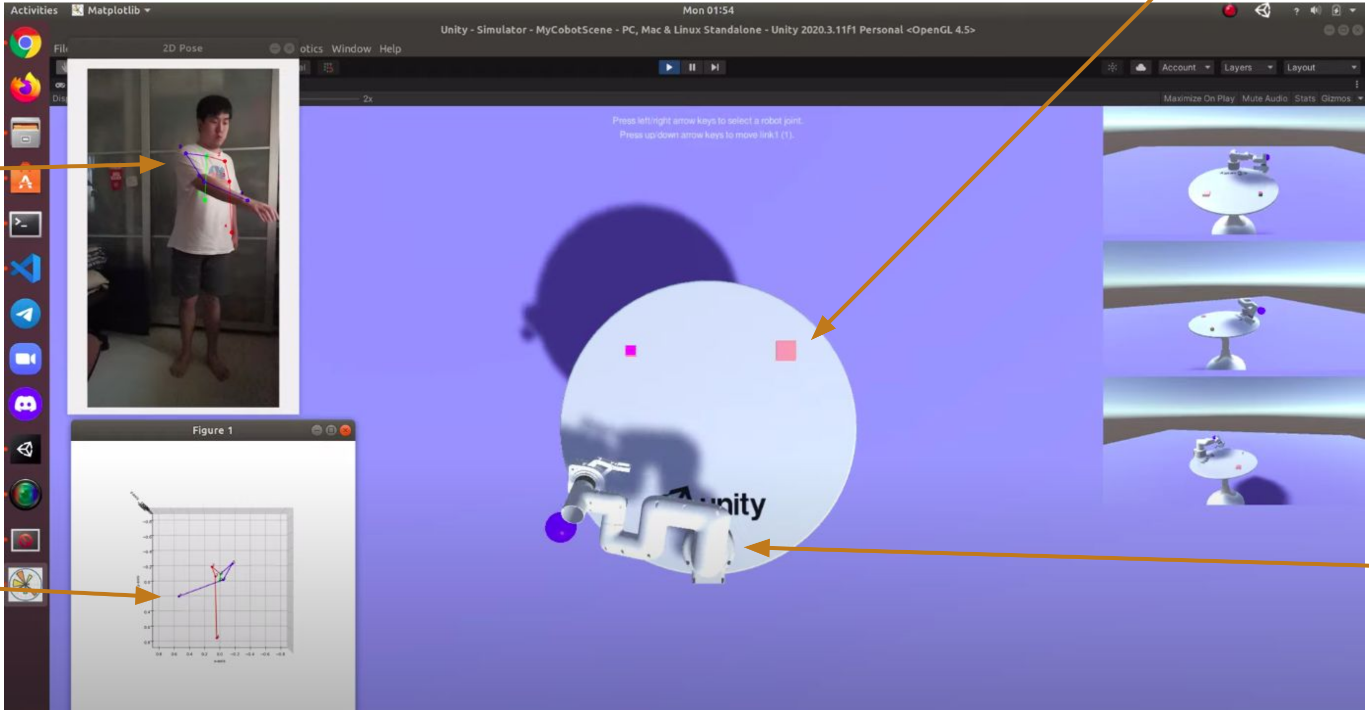
Image Info?

Turning left or right?

Previous

Trained model on vector of 2D poses

Unreasonable poses for difficult 2D poses



Choppy trajectory

Not physical robot

Related Works

Overview Of Related Works

1. **Chernytska**: Resnet based network (2019)
2. **Zhao**: SemGCN network (2019)
3. **Bazarevsky**: Lightweight network (2020)

ResNet Based Network

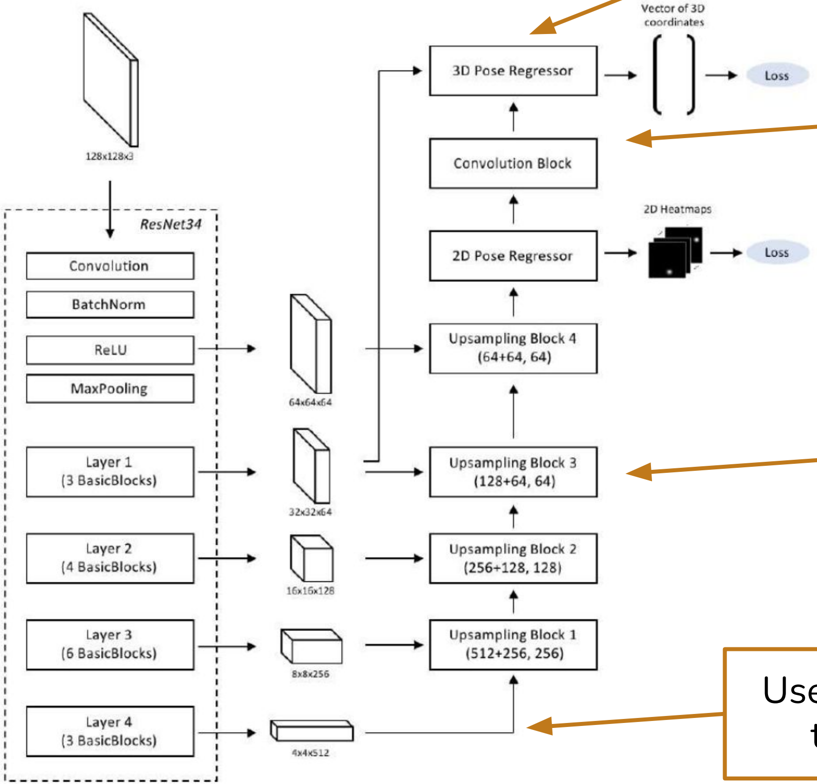
Use fully-connected layer as final layer

Use heatmaps

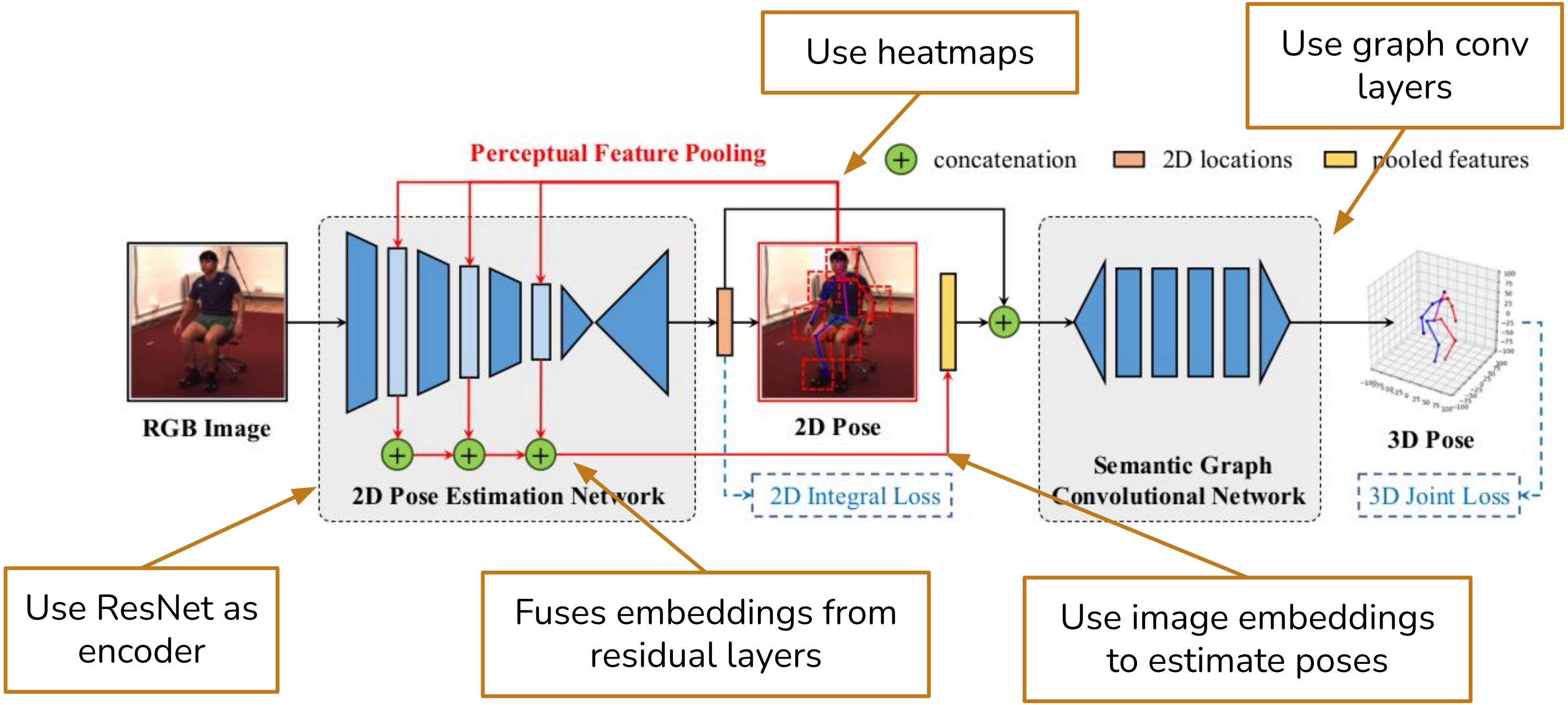
Use ResNet as encoder

Fuses embeddings from residual layers

Use image embeddings to estimate poses

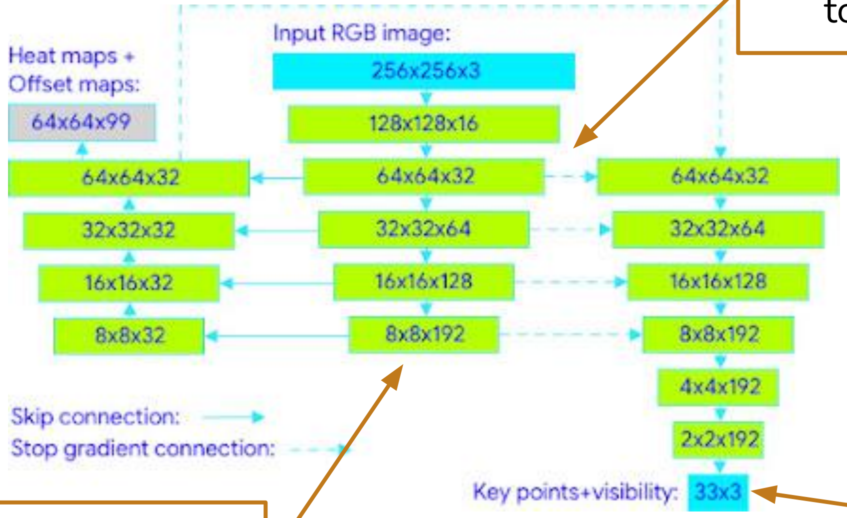


SemGCN


Use heatmaps
Use graph conv layers
Perceptual Feature Pooling
concatenation
2D locations
pooled features
RGB Image
2D Pose Estimation Network
2D Pose
2D Integral Loss
Semantic Graph Convolutional Network
3D Pose
3D Joint Loss
Use ResNet as encoder
Fuses embeddings from residual layers
Use image embeddings to estimate poses

BlazePose

Use image embeddings to estimate poses

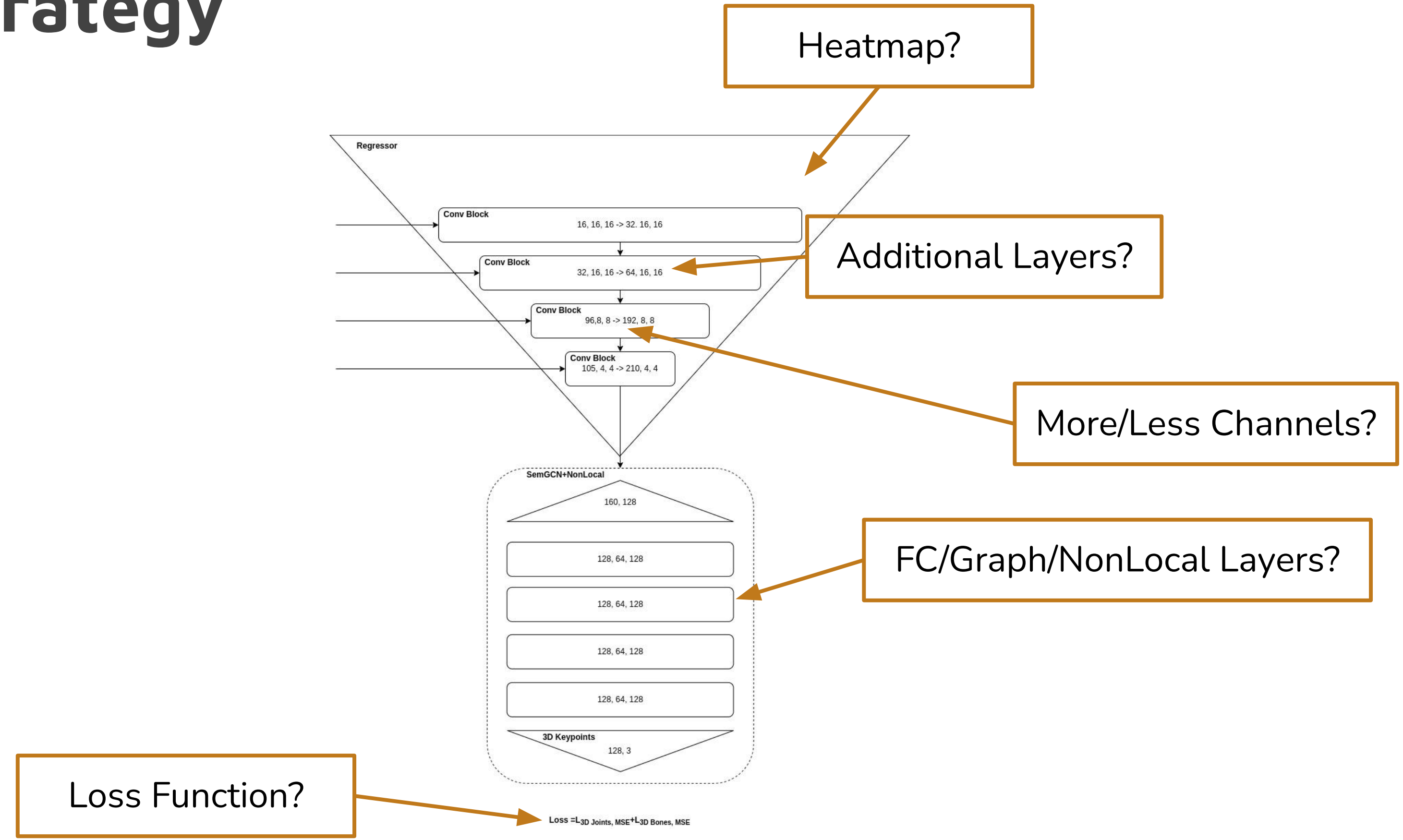


Discard heatmap to reduce computation

Use lightweight encoder

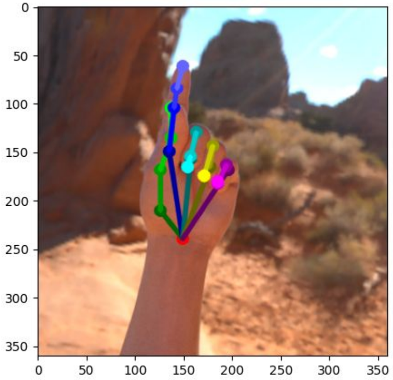
Use fully-connected layer as final layer

Strategy

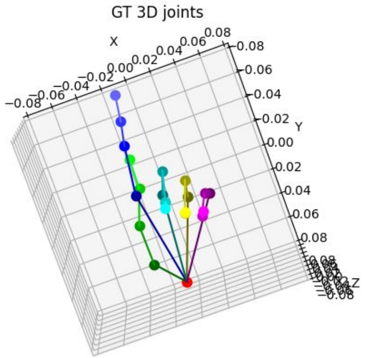

Heatmap?
Regressor
Conv Block
16, 16, 16 -> 32. 16, 16
Conv Block
32, 16, 16 -> 64, 16, 16
Additional Layers?
Conv Block
96,8, 8 -> 192, 8, 8
Conv Block
105, 4, 4 -> 210, 4, 4
More/Less Channels?
SemGCN+NonLocal
160, 128
128, 64, 128
128, 64, 128
FC/Graph/NonLocal Layers?
128, 64, 128
128, 64, 128
3D Keypoints
128, 3
Loss Function?
Loss =$L_{3D\ Joints,\ MSE}$+$L_{3D\ Bones,\ MSE}$

Datasets

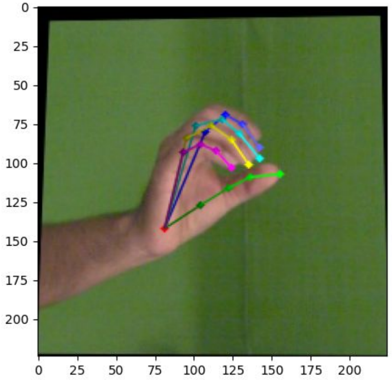
NTU Hand Dataset



Synthetically generated data

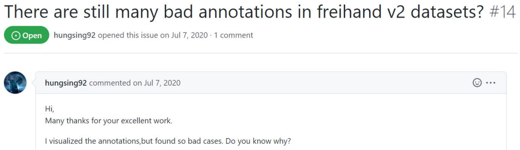


Freihand Dataset



Manually annotated data

Unresolved Github issue on the annotation accuracy


There are still many bad annotations in freihand v2 datasets? #14

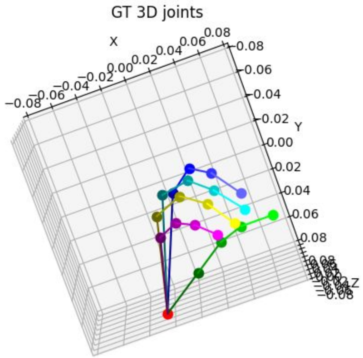
Open hungsing92 opened this issue on Jul 7, 2020 · 1 comment

hungsing92 commented on Jul 7, 2020

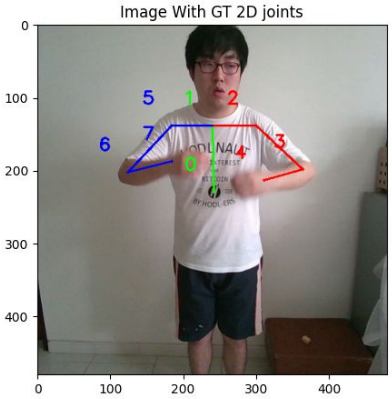
Hi,

Many thanks for your excellent work.

I visualized the annotations,but found so bad cases. Do you know why?

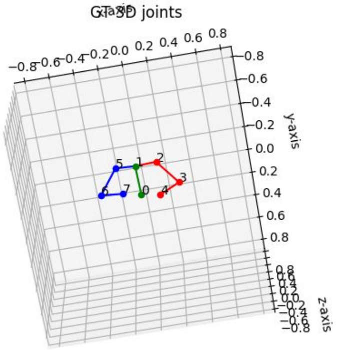


Custom Upper Body Dataset



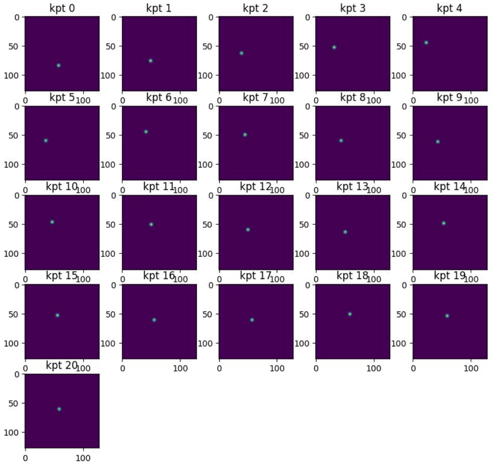
Record 3D poses with depth camera

Identify joint using 2D pose estimator to read the depth pixel

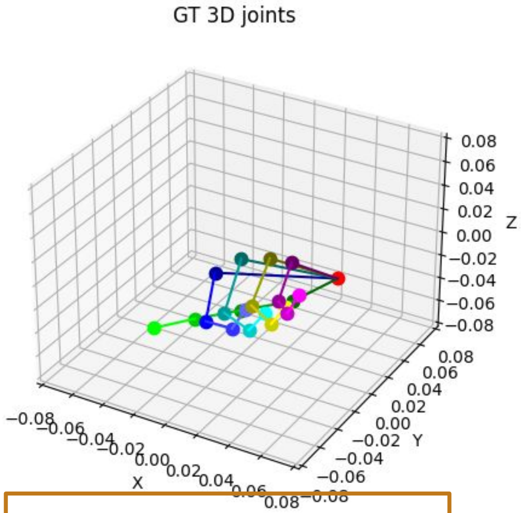


Training Pipeline

Data Annotation



Bright spot in heatmap is the joint location



3D poses is a vector of N by 3

Data Augmentation

Augmentation Step	Description	Hand Pose	Upper Body Pose
Brightness	Adjusted image brightness by a factor in the range of [-0.25, 0.25]	Yes	Yes
Contrast	Adjusted image contrast by a factor in the range of [-0.25, 0.25]	Yes	Yes
Sharpness	Adjusted image sharpness by a factor in the range of [-0.25, 0.25]	Yes	Yes
Mirror	Mirror original image	Yes	No
Flip	Flip original image	Yes	No
Rotate	Rotate by an angle in the range of [0, 360) degrees	Yes	Yes
Translate	Translate by a pixel in the range of [-100, 100] pixels	Yes	Yes
Fill holes	Fill black pixels after transformation with background images	Yes	Yes

Formula to rotate 3D poses

$$\mathbf{P}_{\mathbf{rotated}} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{P}$$

Formula to translate 3D poses

$$x_{new} = x_{old} - \frac{p_x}{f_x} \times z$$

$$y_{new} = y_{old} - \frac{p_y}{f_y} \times z$$

Training Details

Training Detail	Description
Epochs	Trained for about 200 epochs for Pose 2D and 50 epochs for Pose 3D till the validation loss saturated
Batch Size	Trained using minibatch size of 32 to average the loss
Optimizer	Trained with Adam optimizer to adapt the learning rate for different parameters
Learning Rate	Trained with 0.001 for the first half epochs and 0.0001 for the second half to fine tune performance
Embedding	Trained 3D Regressor using the encoded features in Pose 2D Rsti-mator as image embedding
Workers	Set workers to 8 to reduce time spent in I/O computation
Shuffle	Set shuffle flag to true when training the model to vary the training data
Loss Function	Supervisor training for Pose 2D Estimator using IoU loss and Pose 3D Regressor using MSE

Added bone vector loss during fine-tuning

Loss Functions

IOU Loss Function
(Supervise 2D Poses)

$$L_{IoU} = 1 - IoU$$

$$IoU = \frac{I}{U}$$

$$I = \sum_i (y_{pred,i} * y_{true,i})$$

$$U = \sum_i (y_{pred,i} * y_{pred,i}) + \sum_i (y_{true,i} * y_{true,i}) - \sum_i (y_{pred,i} * y_{true,i})$$

Joint Position Loss Function
(Supervise 3D Joint Position)

$$L_{mse} = \sum_i (P_{pred,i} - P_{true,i})^2$$

Bone Vector Loss Function
(Supervise 3D Bone Vector)

$$L_{mse} = \sum_i (B_{pred,i} - B_{true,i})^2$$

Metrics

Mean Per Joint Position Error (MPJPE)

$$E = \sum_{i=1} |\mathbf{p}_{\mathbf{pred}}(i) - \mathbf{p}_{\mathbf{gt}}(i)|$$

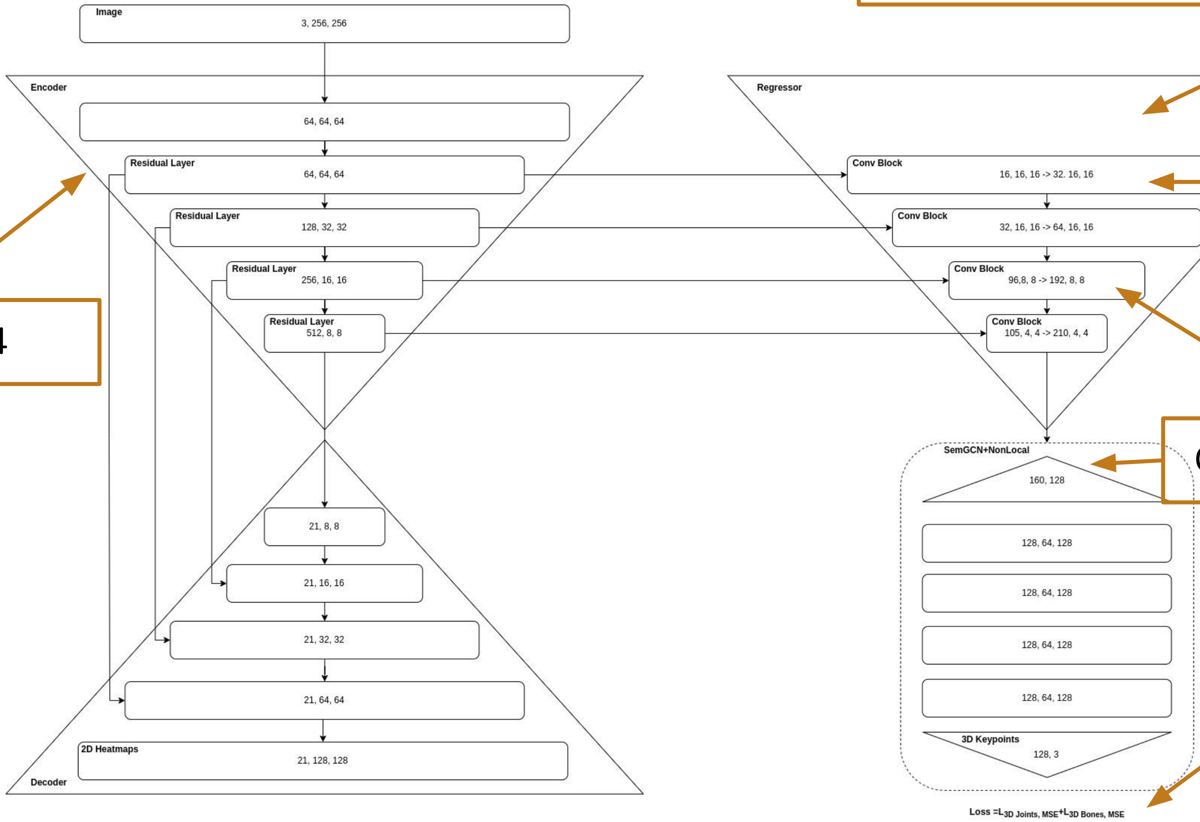
Percentage Correct Keypoints (PCK)

$$Accuracy_{<5/15mm} = \frac{N_{keypoints<5/15mm}}{N_{keypoints}} \times 100\%$$

Models

Best Model

Params: 1.57M
MPJPE: 6.79mm
PCK5mm: 51.1%
PCK15mm: 93.1%



No Heatmap

2 Conv Layers

ResNet34

Default Channels

Graph + NonLocal Layers

MSE Joint + Bone

Model 1.X

Residual blocks adds complexity

Models	Heat map	Regressor Module	Final Layer	MSE Loss Function
v1.0	Yes	1 Conv + Residual Block	Fully-Connected	3D Joints
v1.1	No	1 Conv + Residual Block	Fully-Connected	3D Joints
v1.2	No	1 Conv + Residual Block	Fully-Connected	3D Joints
v1.3	No	1 Conv + Residual Block	Fully-Connected	3D Joints

Heatmap contributes little to performance

Model	Params	FPS	MPJPE	PCK@5mm	PCK@15mm
v1.0	13.44M	12.6	8.66	30.77	87.85
v1.1	13.30M	14.77	8.24	34.68	89.04
v1.2	16.85M	13.85	8.64	32.05	87.46
v1.3	8.58M	17.33	9.24	29.99	84.63

Channels drastically increase the number of params

Model 2.X

1 Conv layer used for comparison

Models	Heat map	Regressor Module	Final Layer	MSE Loss Function
v2.0	Yes	1 Conv	Fully-Connected	3D Joints
v2.1	No	1 Conv	Fully-Connected	3D Joints
v2.2	No	1 Conv	Fully-Connected	3D Joints
v2.3	No	1 Conv	Fully-Connected	3D Joints

Heatmap contributes little to performance

Model	Params	FPS	MPJPE	PCK@5mm	PCK@15mm
v2.0	2.08M	20.16	10.61	21.73	80.08
v2.1	2.07M	21.02	10.46	22.95	80.14
v2.2	8.20M	18.14	11.84	17.24	74.90
v2.3	0.63M	22.47	12.47	17.02	70.57

Channels drastically increase the number of params

Model 3.X

SemGCN + NonLocal replaced FC layer

Models	Heat map	Regressor Module	Final Layer	MSE Loss Function
v3.0	Yes	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.1	No	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.2	No	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.3	No	1 Conv	4 SemGCN + NonLocal	3D Joints

Heatmap contributes little to performance

Model	Params	FPS	MPJPE	PCK@5mm	PCK@15mm
v3.0	2.19M	18.02	8.24	35.63	88.76
v3.1	2.18M	19.17	8.28	33.84	89.41
v3.2	8.46M	16.23	7.64	39.70	90.70
v3.3	0.72M	19.40	9.10	33.65	84.90

Channels drastically increase the number of params

Model 4.X

SemGCN replace FC layer only

Models	Heat map	Regressor Module	Final Layer	MSE Loss Function
v4.0	Yes	1 Conv	4 SemGCN	3D Joints
v4.1	No	1 Conv	4 SemGCN	3D Joints
v4.2	No	1 Conv	4 SemGCN	3D Joints
v4.3	No	1 Conv	4 SemGCN	3D Joints

Heatmap contributes little to performance

Model	Params	FPS	MPJPE	PCK@5mm	PCK@15mm
v4.0	2.18M	18.99	8.31	34.71	88.89
v4.1	2.17M	20.00	8.63	33.26	87.76
v4.2	8.45M	17.38	8.09	37.64	88.89
v4.3	0.71M	21.01	10.25	24.60	81.43

Channels drastically increase the number of params

Model 3.X

Models	Heat map	Regressor Module	Final Layer	MSE Loss Function
v3.0	Yes	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.1	No	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.2	No	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.3	No	1 Conv	4 SemGCN + NonLocal	3D Joints

Model	Params	FPS	MPJPE	PCK@5mm	PCK@15mm
v3.0	2.19M	18.02	8.24	35.63	88.76
v3.1	2.18M	19.17	8.28	33.84	89.41
v3.2	8.46M	16.23	7.64	39.70	90.70
v3.3	0.72M	19.40	9.10	33.65	84.90

Can we improve this?

Number of params is more than 4 times

Model 3.1.X

Models	Heat map	Regressor Module	Final Layer	MSE Loss Function
v3.1.0	No	1 Conv	4 SemGCN + NonLocal	3D Joints
v3.1.1	No	1 Conv	2 SemGCN + NonLocal	3D Joints
v3.1.2	No	1 Conv	6 SemGCN + NonLocal	3D Joints
v3.1.3	No	2 Conv	4 SemGCN + NonLocal	3D Joints
v3.1.4	No	1 Conv	4 SemGCN + NonLocal	2D Joints + 3D Joints
v3.1.5	No	1 Conv	4 SemGCN + NonLocal	3D Joints + 3D Bone

Model	Params	FPS	MPJPE	PCK@5mm	PCK@15mm
v3.1.0	2.18M	19.17	7.29	40.78	92.78
v3.1.1	2.11M	20.05	7.32	41.59	92.49
v3.1.2	2.24M	18.38	6.93	44.52	93.64
v3.1.3	1.57M	19.37	6.87	44.94	93.89
v3.1.4	2.18M	19.13	8.52	30.43	89.30
v3.1.5	2.18M	19.17	6.87	45.27	93.56

Train longer at lower learning rate

Add 1 more Conv layer instead of residual block

Add bone vector loss for supervision

Results

Overview Of Results

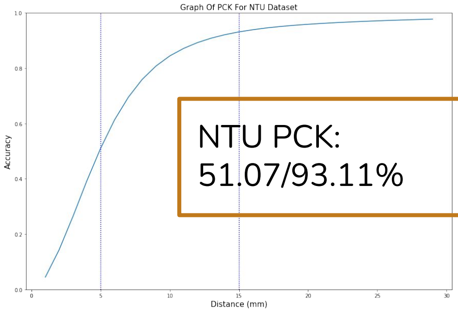
1. Hand Pose (Benchmark)
2. Upper Body Pose (Image Embedding vs 2D Poses)
3. Successful Pose Estimation
4. Failed Pose Estimation

Hand Models

Performs better on NTU dataset*

Model	Dataset	Full Params	Pose 3D Module Params	MPJPE
Ours	NTU	23.05M	1.57M	6.79*
L. Ge [3]	NTU	21.77M	9.19M	8.03
Ours	Freihand	23.05M	1.57M	9.08*
K. Lin [6]	Freihand	98.43M	-	6.00
H. Choi [2]	Freihand	74.96M	67.60M	7.40

Does not perform too far off but model params is much smaller



NTU PCK: 51.07/93.11%

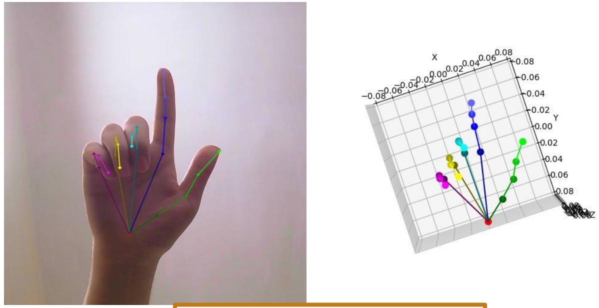
Upper Body Models

Model	v3.1.6			SemGCN		
	MPJPE	PCK@15mm	PCK@30mm	MPJPE	PCK@15mm	PCK@30mm
Average	25.93	39.05	70.58	51.06	21.70	47.67

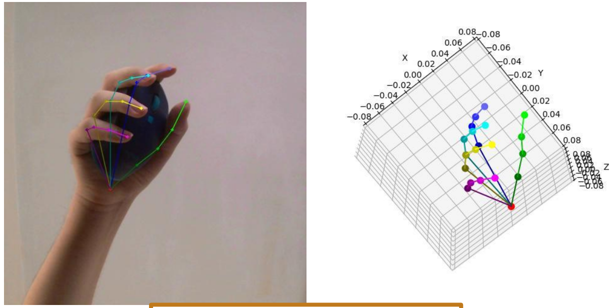
Uses image embeddings as input

Uses vector of 2D poses as input

Success

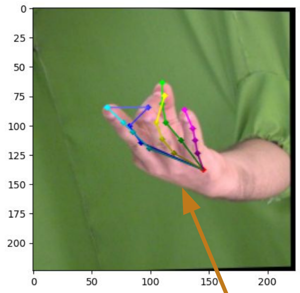


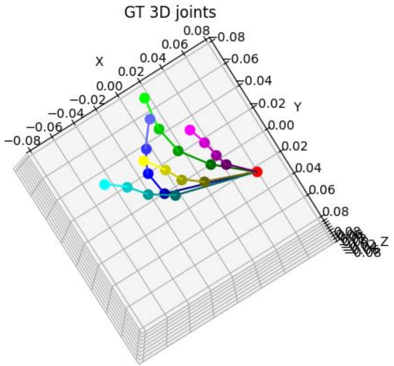
Generalise well
to my hand

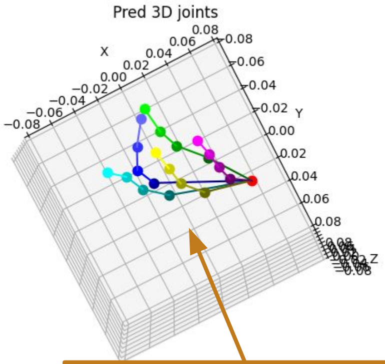


Generalise well
with occlusions

Failure





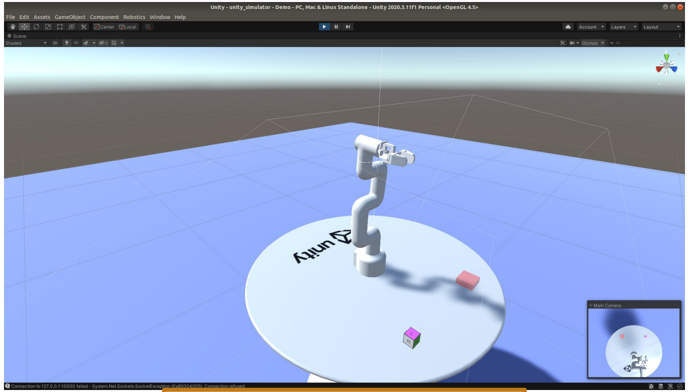


Estimate 2D poses incorrectly

Predict reasonable 3D poses

Demo

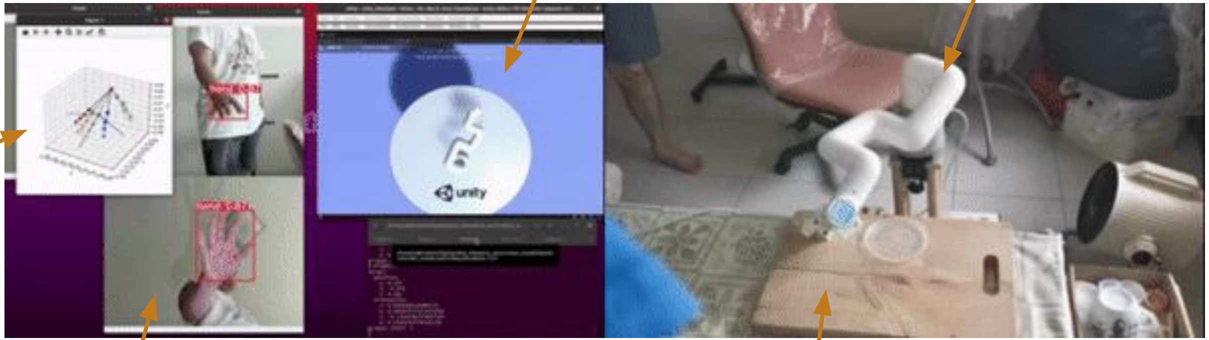
Robot Setup



Setup Unity as simulated robot

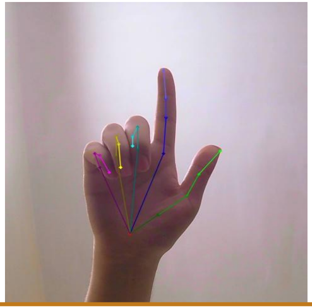
Setup MyCobot as physical robot

Teleoperation

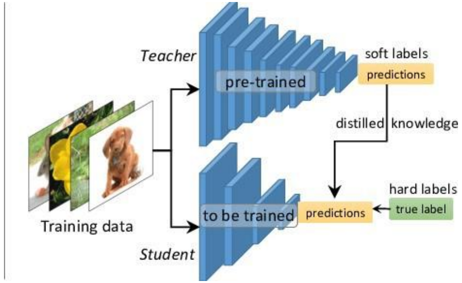


Conclusion

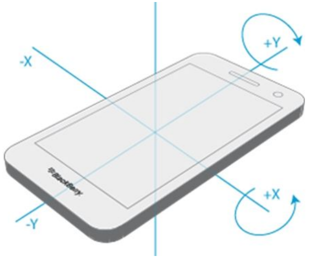
Future Exploration



Limited to one hand/body



Model Distillation



Mobile Phone IMU And Cameras For Controls

References

[1] Valentin Bazarevsky et al. BlazePose: On-device Real-time Body Pose tracking. 2020.

[2] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose. 2020.

[3] Liuhao Ge et al. 3D Hand Shape and Pose Estimation from a Single RGB Image. 2019.

[4] Kaiming He et al. Deep Residual Learning for Image Recognition. 2015.

[5] Shuang Li et al. A Mobile Robot Hand-Arm Teleoperation System by Vision and IMU. 2020.

[6] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh Graphormer. 2021.

[7] Julieta Martinez et al. A simple yet effective baseline for 3d human pose estimation. 2017.

[8] Chernytska Olha and Pranchuk Dmitry. 3D Hand Pose Estimation from Single RGB Camera. 2019.

[9] Daniil Osokin. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight Open-Pose. 2018.

[10] Elephant Robotics. MyCobot Ros. URL : https://github.com/elephantrobotics/mycobot_ros.

[11] Susumu Tachi. Forty Years of Telexistence — From Concept to TELESAR VI. 2019.

[12] Unity Technologies. Unity Robotics Hub. URL : https://github.com/Unity-Technologies/Unity-Robotics-Hub.

[13] Jinbao Wang et al. Deep 3D human pose estimation: A review. 2021.

[14] Xiaolong Wang et al. Non-local Neural Networks. 2018.

[15] Long Zhao et al. Semantic Graph Convolutional Networks for 3D Human Pose Regression. 2020.

[16] Christian Zimmermann et al. FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. 2019.