Empirical Study on MMPOSE 2D Hand Keypoint Detection (Topdown Heatmap & ResNet)

Weijun Huang
Department of MIE
University of Toronto
Toronto, Ontario, Canada
weijun.huang@mail.utoronto.ca

Zhiyuan Lyu
Department of MIE
University of Toronto
Toronto, Ontario, Canada
zhiyuan.lyu@mail.utoronto.ca

Zixuan Wan
Department of MIE
University of Toronto
Toronto, Ontario, Canada
z.wan@mail.utoronto.ca

***Abstract*— This paper conducts an empirical study on the "mmpose 2D Hand Keypoint Detection" which adopts methods of HeatMap Regression [4] and Residual Network [3]. The model takes input from a hand image and outputs keypoint coordinates (x,y) of the specific 21 points in a hand. After redeploying and researching the detection model, it is found that some potential defects exist. Therefore, two hypotheses are held: H1: The test set may not cover all 25 gestures and it is an unbalanced set, resulting in inaccurate performance; H2: The training set may not cover all 25 gestures and is unbalanced, resulting in unbalanced performance between different gestures. To verify H1, 500 images from the FreiHand test set are selected based on 25 gesture classes for constructing a new test set baseline, then by adding 50 more images from own test set (selfie), the final test set for H1 is finalised, which consists of 500 FreiHand test set and 50 own image test set. And by comparing AUC, EPE, and PCK [8] between the final test set and the baseline, H1 is rejected because the performance matrices of two test sets differ only slightly. To verify H2, 5 images from each class are selected for constructing baseline of each class, then by adding 2 more images from each class of own test set, the final test set for H2 is finalised. And by comparing AUC and EPE between the final test set and the baseline, H2 is not reject, the training set is unbalanced, which results in poor performance in gestures 6 (fist gesture).**

***Index Terms*— MMPOSE, 2D Hand Keypoint Detection, FreiHAND dataset**

I. Introduction

A. Background

There are a variety of applications in gesture recognition, computer vision and artificial intelligence regarding 2D hand pose estimation and keypoint detection from a single RGB image. There are plenty of machine learning algorithms that produce 2D hand pose estimation in a variety of environments. In this paper, the 2D hand keypoint detection algorithm is re-implemented by using mmpose, which is an open-source toolbox for pose estimation based on PyTorch.

The 2D hand pose is generally visualized as a skeleton with 21 keypoints, visualized in Fig.1. However, for most of the images, not all 21 key points are visible and this misleads the results on a great scale. Thus, the input images are generally split into 25 classes of gestures [2], visualized in Fig.2. In these 25 gestures, some points are visible and some are not, this information is fed into the input data, and used to improve the performance of the model.

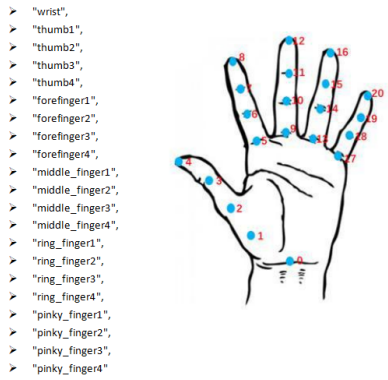

Fig. 1: Hand 21 Keypoints

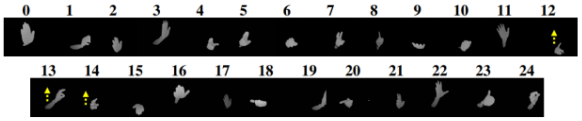

Fig. 2: 25 Hand Gestures [2]

In this project, the mmpose 2D Hand KeyPoint Detection Algorithm (Topdown Heatmap & ResNet) is redeployed, after doing research, it is found that the algorithm might has some defects. Therefore, 2 hypotheses are proposed, H1: The test set may not cover all 25 gestures and it is an unbalanced set, resulting in inaccurate performance; H2: The training set may not cover all 25 gestures and is unbalanced, resulting in unbalanced performance between different gestures. And for verifying these 2 hypotheses, 2 experiments are performed.

For the first experiment, the model is firstly tested with 500 images from FreiHand test set [9], and the test results are treated as baseline. Then the 500 images from FreiHand test set + 50 images collected from other sources (selfies) are tested again for comparison with the baseline. These 500 labeled images from FreiHand test set are selected with 25

gestures evenly distributed to control the input [4]. And the 50 selfied images are manually labeled through the Labelme tool.

For the second experiment, 5 images from FreiHand test set are firstly tested for each class of 25 gestures [4], which return 25 baseline results. Then 5 images from FreiHand test set + 2 selfied images are also tested for each class, then the 25 results are compared to corresponding baseline result. The performance matrices are AUC, EPE (Average End Point Error) and PCK (Distance between predicted and true joint < 0.2 * torso diameter) [8].

B. Objectives

- Re-implement the "mmpose 2D Hand Keypoint Detection" algorithm and download the pre-trained model from mmpose GitHub.
- Collect 50 own images that are evenly distributed into 25 gestures, that is 2 images per gesture. Then label the images manually as true values for performance analysis.
- Verify the hypothesis H1 by comparing AUC, EPE and PCK score of the dataset of 500 FreiHand images for whether the 50 own image dataset is added or not.
- Verify the hypothesis H2. For each gesture, compare the AUC and EPE score for 5 FeriHand images vs 5 FeriHand images + 2 own testset images.

C. Problem Statements

Although the algorithm has been perfected, it still has some potential problems that are needed to define and solve. To better categorize these problems, The potential problems are analysed from two aspects: problems caused by external factors and problems caused by internal factors. Summarizing and learning from these problems beforehand can extend the project deeper, which can finalise a high-quality project.

- Caused by external factors:
  - It is found that different input image sizes can affect the result. When the picture quality is too high or too low, the key points captured by the model to the hand will be deteriorated, thereby reducing the performance of the model.
  - Moreover, images transferred from high resolution to low resolution will lose some features, so the predictions of hand key points might be blurred or overlapped. To avoid this problem as much as possible, the selection of hand pictures should be chosen with lower pixels and resized uniformly. In addition, the background with significant variance or varying illuminations will fail to precisely predict the hand key points. To solve these, participants need to choose the image with same background and light.
- Caused by internal factors:
  - The test set may not cover all 25 gestures and may be unbalanced, so the performance results might be inaccurate.
  - The training set may not cover all 25 gestures and may be unbalanced, which performs well in some classes and not so well in others.

The problems caused by external factors can be addressed by using image processing techniques, however, the problems caused by internal factors need to be verified through experiments. Therefore, two hypotheses are held based:

- Hypothesis 1: The test set may not cover all 25 gestures and is unbalanced, resulting in inaccurate performance.
- Hypothesis 2: The training set may not cover all 25 gestures and is unbalanced, resulting in unbalanced performance between different gestures.

These hypotheses will be judged in combination with the experimental results to determine whether they are rejected or not rejected.

D. Summary of Contributions

- Re-deploying and researching the mmpose detection model.
- Summaries two hypotheses:
  - H1: The test set may not cover all 25 gestures and is unbalanced, resulting in inaccurate performance.
  - H2: The training set may not cover all 25 gestures and is unbalanced, resulting in unbalanced performance between different gestures.
- For H1, build a baseline test set by selecting 500 images from FreiHand test set; Build a final test by combining the baseline test set and 50 more images from own test set (selfie).
- For H2, build 25 baseline test sets by selecting 5 images from each class of FreiHand test set; Build 25 final tests by adding 2 more images from each class of own test set to each baseline test set.
- By computing and comparing the AUC, EPE (PCK) between baseline and final test set, H1 is rejected because the performance matrices of two test sets differ only slightly. And H2 is not rejected, the training set is unbalanced, which results in poor performance in gestures 6 (fist gesture).

II. Related Works

[6] introduces a novel method for real-time 2D hand pose estimation from monocular color images, which is named as SRHandNet. It uses an encoder-decoder architecture as the backbone of our network to perform the 2D hand pose estimation. In the training stage, intermediate supervision is adopted. In the inference stage, performs the cycle detection according to the size of a hand. The key idea is to simultaneously regress the hand region of interests (ROIs) and hand key points for a given color image. For advantage, it can run

at 40fps for hand bounding box detection and up to 30fps for hand key points estimation under an ordinary desktop environment, which competes with all recent methods.

[1] proposes HandFoldingNet, an accurate and efficient hand pose estimator that regresses the hand joint locations from the normalized 3D hand point cloud input. The proposed model utilizes a folding-based decoder that folds a given 2D hand skeleton into the corresponding joint coordinates. For higher estimation accuracy, folding is guided by multi-scale features, which include both global and joint-wise local features. For advantage, the experimental results on three challenging benchmarks showed that this network outperforms previous state-of-the-art methods while requiring minimal computational resources. Ablation experiments demonstrated the contribution of its key components for better accuracy and efficiency.

[5] introduces a vision-based human pose estimation is a noninvasive technology for Human-Computer Interaction (HCI). The proposed model can be deployed on an embedded system, in contrast to most pose estimation methods which incorporate complex and computationally inefficient architectures, it proposed single-stage end-to-end CNN model exhibits competitive results with just 1.9M parameters and a model size of 11 Mbytes, which is achieved by directly predicting the joints' coordinates.

III. Empirical Materials & Methodology

A. Description of Algorithm

The basis of this hand pose estimation algorithm is Top-down Heatmap + ResNet. The network structure has no pyramid structure and is constructed based on standard ResNet, which is shown below in Fig.3. There are 3 deconvolution modules behind the output feature layer of the last residual model [7]. For each deconvolution module, batchnorm + ReLU follow the output of deconvolution layer, and each deconvolution has parameters of 256 channels, 44 convolution kernels, stride of 2 and pad of 1 [7]. After these 3 modules, a 1 x 1 convolutional layer is appended at last to produce predicted HeatMaps for keypoints regression [7]. Moreover, the loss between prediction and target HeatMap is MSE (Mean Squared Error) [7].

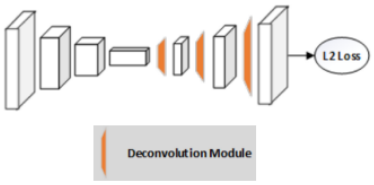

Fig. 3: Network Structure of Topdown Heatmap + ResNet

The input of the network is a hand-bounding-box, which predicts the coordinates of 21 keypoints. The Top-down HeatMap component uses a regular Gaussian Kernel HeatMap, which directly regresses the coordinate by feature matching on the spatial dimension, that is, it 'slides' the convolution kernel on the feature map plane. Then positioning is obtained by means of the response maximum point index (Argmax). Relatively speaking, the calculation of each key point is independent. (It can be understood as taking a convolution kernel to calculate one by one to see if each part conforms to the characteristics of the key point. The larger the calculated response, the more likely it is a key point) [4].

B. Description of Data

FreiHAND is a dataset for evaluation and training of deep neural networks for estimation of hand pose and shape from single color images, which is proposed by [9], The current version of it has 130,240 data (80% training set, 10% test set and 10% validation set). All images are annotated, the annotation JSON file is a large dictionary that contains 5 keys: ['info', 'licenses', 'images', 'annotations', 'categories']. The value of 'images' and 'annotations' keys are both lists containing 130,240 dictionaries, each dictionary in 'images' list has 4 keys: ['file_name', 'height', 'width', 'id'], and corresponding value are image name, bounding box height, bounding box width and number in image name. And each dictionary in the 'annotations' list also has several keys, where the key 'keypoints' has 21 values, which are the 2D coordinates of 21 keypoints and the visible or invisible labels for each coordinate.

C. Experimental Design

1. Prerequisite Setup
   - 1.1 CUDA download
     - 1.1.1 CUDA 11.4
   - 1.2 Conda environment setup
     - 1.2.1 conda create -n openmmlab python=3.7 -y
     - 1.2.2 conda activate openmmlab
     - 1.2.3 pip install pytorch==1.12.1 torchvision==0.13.1 torchaudio==0.12.1
     - 1.2.4 pip install openmim
     - 1.2.5 mim install mmcv-full
   - 1.3 Git clone mmpose file
     - 1.3.1 git clone https://github.com/open-mmlab/mmpose.git
     - 1.3.2 cd mmpose
     - 1.3.3 pip install -e .
2. File structure setup in mmpose.
   - 2.1 Create file 'model' under the 'mmpose' root path, then put the downloaded model into it.
     - 2.1.1 Downloaded model: res50_freihand_224x224-ff0799bc_20200914.pth.

2.2 Download FreiHAND data set, extract and rearrange files as mmpose official site shown.
2.3 Create file 'selecting_1', 'selecting_2', 'selecting_3' under 'mmpose/data/freihand'.
2.4 Create file 'project_test_data/own_test' under 'mmpose/data/freihand'.
2.5 Create file 'own_resize_test' under 'mmpose/data/freihand/training'.
2.6 Create file 'class_own_test' under 'mmpose/data/freihand/training'.
2.6.1 Create 'c0' to 'c24' files under 'class_own_test'.

3. Configure new test set for verifying Hypothesis 1: The test set may not cover all 25 gestures and unbalanced, resulting in inaccurate performance.
3.1 Based on 25 gesture classes of Fig.2, 500 images are selected from Freihand test set (gesture classes are evenly distributed), and corresponding annotations are copied from original freihand_test.json (renamed to freihand_test_ori.json) to a new freihand_test.json under 'mmpose/data/freihand/annotations'.
3.1.1 Since images (130,240) under 'mmpose/data/freihand/training/rgb' are training (80%) + test (10%) + validation set (10%), mmpose separate sets by using JSON. And since 3-member group, test images are separated and copied to another 3 files for selecting.
3.1.2 The keys "file_name" are read from original freihand test JSON file (now named 'freihand_test_ori.json') under 'mmpose/data/freihand/annotations' to locate all the test set images, then images are copied to 3 new files 'selecting_1', 'selecting_2' and 'selecting_3' under 'mmpose/data/freihand'.
3.1.3 Then based on 25 gesture classes, member 1/2/3 select 167 imagses from 'selecting_1/2/3', 6-7 images for each gesture classes.
3.1.4 Each member returns a TXT file containing the selected image name, and the image names in the TXT file are displayed line by line.
3.1.5 The 3 TXT files, which named 'selected_imgs_name_1', 'selected_imgs_name_2', 'selected_imgs_name_3', are saved under 'mmpose/data/freihand/annotations'.
3.1.6 According to the images name in 3 TXT files, extract corresponding annotations from original freihand_test.json (renamed to freihand_test_ori.json) to a new freihand_test.json under 'mmpose/data/freihand/annotations'.
3.2 Construct a dictionary based on original freihand test JSON file.
3.2.1 Read the 500 selected images and extract the corresponding values of key 'images' and 'annotations' from freihand_test_ori.json.
3.2.2 Write 500 selected images json to new freihand_test.json (with annotations of selected 500 images) under 'mmpose/data/freihand/annotations'.
3.3 Test on 500 Freihand test set (baseline for Hypothesis 1).
3.3.1 Performance Metric: AUC, EPE, PCK.
3.4 Collect 50 more images from other sources (provided by Zhiyuan Lyu) for constructing the own images test set, images are stored into 'mmpose/data/freihand/project_test_data/own_test', 2 images for each class.
3.5 Resize the 50 own image test set to 244×244, then store the resized images into 'mmpose/data/freihand/training/own_resize_test'.
3.6 Use Labelme to manually annotate 21 keypoints for each resized image. Each images produce one annoations file, so the 50 annoation json files are stored under 'mmpose/data/freihand/training/own_resize_test'.
3.7 Reconstruct the JSON dictionary structure of own resized image test set according to the outputted Labelme annotation JSON file.
3.8 Append the reconstructed own image JSON data to the new freihand_test.json under 'mmpose/data/freihand/annotations'.

4. Test for Hypothesis 1 (500 FreiHand + 50 own_test)
4.1 Performance Metric: AUC, EPE, PCK.

5. Test for Hypothesis 2 (performance of each class).
5.1 Based on 'selected_imgs_name_2', select 5 images for each class.
5.2 Store the image names for each class in 25 individual TXT files under 'mmpose/data/freihand/annotations'
5.3 Copy the 50 own test image annotation JSON files to 'mmpose/data/freihand/training/class_own_test/c*' (* refer to class number)
5.4 Testing (performance of each class) Hypothesis 2, For each class:
5.4.1 Delete the previous freihand_test.json file before testing for each class.
5.4.2 Write the 5 selected Freihand images annoation into freihand_test.json.
5.4.3 Test on the 5 selected Freihand images (baseline for Hypothesis 2, Performance Metric: AUC, EPE.
5.4.4 Add the 2 own_test_image annoations into freihand_test.json.
5.4.5 Test on the 5 selected Freihand images + 2

own_test images (Performance Metric: AUC, EPE).

6. Results DataFrame construction (hypothesis 1 and 2)
   6.1 DataFrame of baseline and test result for H1.
   6.2 DataFrame of baseline and test result for H2.

IV. Results & Discussions

A. Empirical Results of Hypothesis 1

Empirical Results of Hypothesis 1 are shown below in Fig.4 and Fig.5. The baseline (blue legend) represents the evaluated results of 500 images from the FreiHAND test set, and the test results (red legend) indicate the combination of the 500 images from FreiHAND test set and the 50 images from own test set. As the figures show, there are not appear significant differences in AUC and PCK for the two datasets, but EPE value has more than 0.3 volatility.

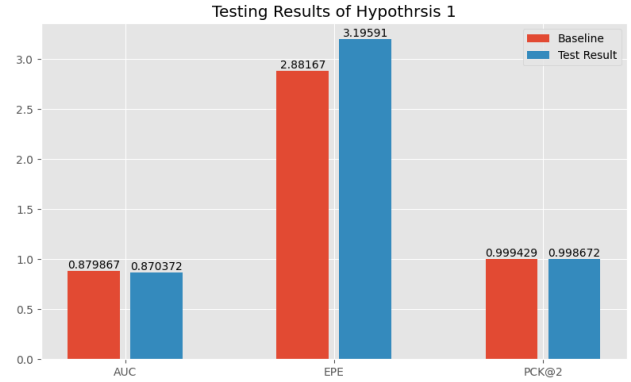

Fig. 4: Testing Results of H1

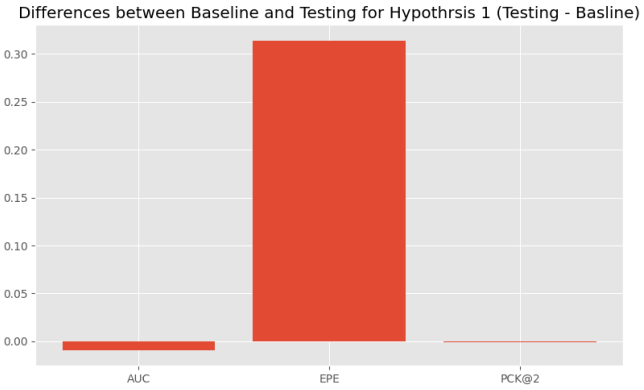

Fig. 5: Differences between Baseline and Testing for H1

Therefore, based on these analyses, Hypothesis 1 is rejected since the model performance on baseline and final test set has only a slight difference. The test set covers all 25 gestures and it is a balanced set.

B. Empirical Results of Hypothesis 2

Empirical Results of Hypothesis 2 are shown below in Fig.6, Fig.7, Fig.8 and Fig.9, where the former 2 figures are AUC results and the latter 2 figures are EPE results. These figures show the comparison plots of AUC and EPE values for each class in the two datasets.

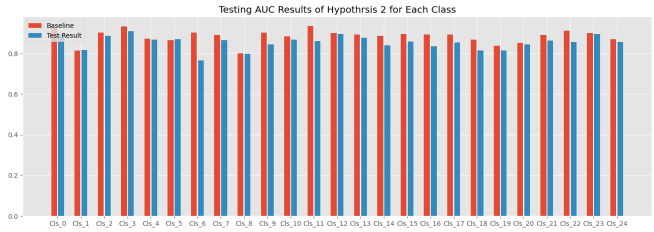

Fig. 6: Testing AUC Results of H2 for Each Class

From Fig.7, it can be clearly seen that the AUC Differences of Class_6, Class_9, Class_11, Class_16, Class_17 and Class_22 are significant if the threshold value is 0.05.

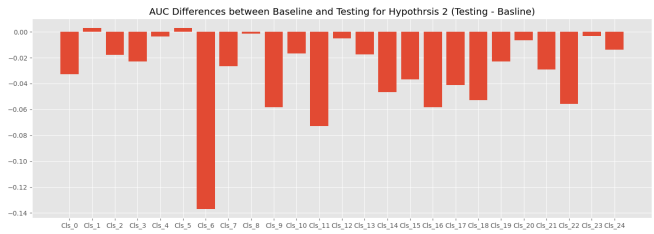

Fig. 7: AUC Differences Between Baseline and Testing for H2 (Testing - Basline)

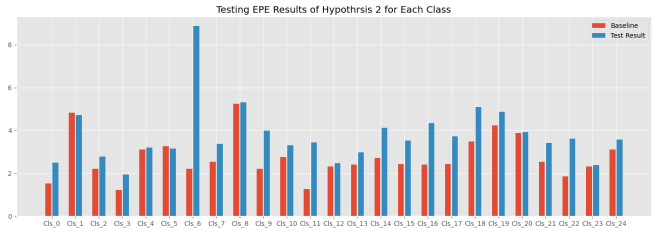

Fig. 8: Testing EPE Results of H2 for Each Class

From Fig.9, most Classes' difference is between 0 to 2 but the EPE difference of Class_6 is the highest (approximately 6.7).

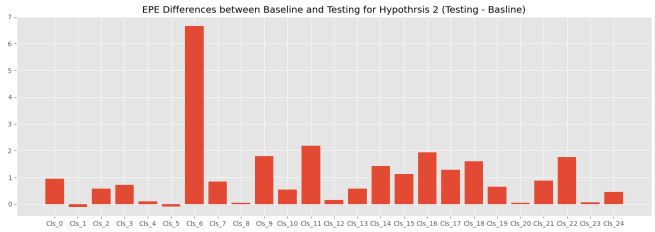

Fig. 9: EPE Differences Between Baseline and Testing for H2 (Testing - Basline)

Therefore, based on these analyses, Hypothesis 2 is not rejected since the AUC and EPE results of Class_6 are significantly poor. The training set is unbalanced, it lacks Class_6 gestures in training data relatively.

C. Inference Results of 50 Own Test Images

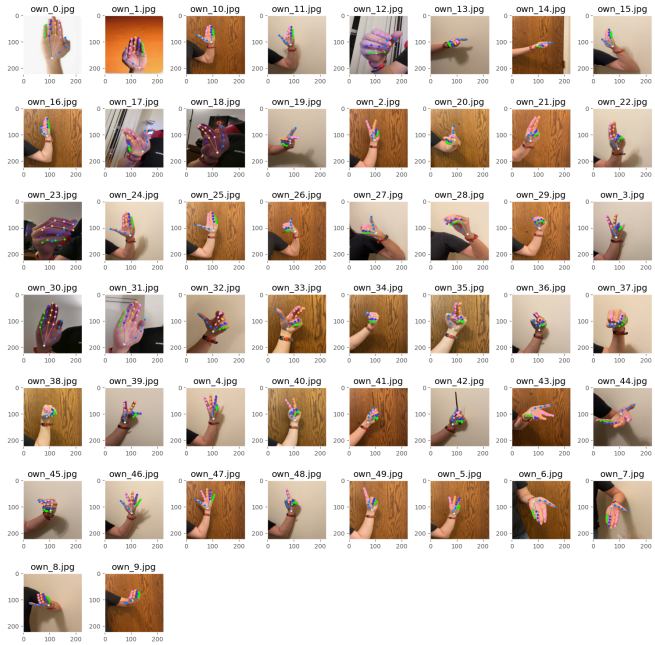

Fig. 10: Inference Results of 50 Own Test Images

Additionally, Fig.10 above shows the inference results of 50 own test images. Overall speaking, the processing results of the model have marked the keypoints of the hand relatively accurately. However, in detail, the outputs for images own_37 and own_38 (Class_6) are not particularly ideal.

V. Conclusion

In this empirical study project, the "mmpose 2D Hand Keypoint Detection" algorithm is re-implemented and the pre-trained model is collected. Then, participants test for the accuracy of the model with the 500 FeriHand test set as well as the collected 50 own image test set. Two hypotheses were brought out as the potential pitfalls of this algorithm, that is H1: The test set may not cover all 25 gestures and it is an unbalanced set, resulting in inaccurate performance; H2: The training set may not cover all 25 gestures and unbalanced, resulting in unbalanced performance between different gestures.

During the verification procedure, participants verify empirically that H1 is rejected, by adding the 50 own images test set that is evenly distributed in 25 gestures to the 500 FreiHand test set, the performance matrices do not differ significantly. This provides the result that the 500 FreiHand test set covers all 25 gestures and it is balanced, thus the performance is still accurate after adding the own test set, and H1 is rejected.

Then H2 is verified empirically, which is not rejected. By comparing the performance metrics of 5 FreiHand images vs 5 FreiHand images + 2 own_images for each gesture, it is found that the AUC and EPE difference on gesture 6 (Fist gesture) is significantly different. This proves the results that the training set of the pre-trained model is unbalanced and lacks gesture 6 and H2 is not rejected.

In conclusion, through empirical study, the mmpose 2D hand pose estimation algorithm is systematically verified based on facts, objective phenomena, and true and accurate data, which can be greatly improved in the future. While the testing set covers all 25 gestures and is balanced, the training set of the pre-trained model might not be balanced in 25 gestures, causing significantly poor performance in gesture 6. This finding can be verified further in the future by training the model on an improved training set and increasing the size of own_image set.

References

[1] W. Cheng, J. H. Park, and J. H. Ko, "Handfoldingnet: A 3d hand pose estimation network using multiscale-feature guided folding of a 2d hand skeleton," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 260–11 269.

[2] P. Gupta, K. Kautz, *et al.*, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," in *CVPR*, vol. 1, no. 2, 2016, p. 3.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] Z. Luo, Z. Wang, Y. Huang, L. Wang, T. Tan, and E. Zhou, "Rethinking the heatmap regression for bottom-up human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 264–13 273.

[5] N. Santavas, I. Kansizoglou, L. Bampis, E. Karakasis, and A. Gasteratos, "Attention! a lightweight 2d hand pose estimation approach," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 488–11 496, 2020.

[6] Y. Wang, B. Zhang, and C. Peng, "Srhandnet: Real-time 2d hand pose estimation with simultaneous region localization," *IEEE transactions on image processing*, vol. 29, pp. 2977–2986, 2019.

[7] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.

[8] L. Yang, S. Chen, and A. Yao, "Semihand: Semi-supervised hand pose estimation with consistency," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 364–11 373.

[9] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 813–822.