Sensors and Materials, Vol. 38, No. 3 (2026) 1463–1479
MYU Tokyo

S & M 4388

Monocular 2D Baseball Swing Pose Estimation Using Multistacked Hourglass Networks

Yu-Liang Hsu[1,2*] and Yu-Ming Lo[1]

[1]Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-sen University, No. 70 Lien-hai Road, Kaohsiung 804201, Taiwan
[2]Department of Electrical Engineering, National Sun Yat-sen University, No. 70 Lien-hai Road, Kaohsiung 804201, Taiwan

(Received September 24, 2025; accepted February 13, 2026)

Keywords: monocular, 2D human pose estimation, baseball swing motions, multistacked hourglass network

Human pose estimation from monocular 2D images has been a fundamental yet challenging task in the computer vision community, with wide-ranging applications in human–computer interaction, animation, and behavior detection. With the rapid development of deep learning techniques, monocular human pose estimation has witnessed remarkable advancements in both 2D and 3D research areas. In this study, we aim to implement multistacked hourglass (MSH) networks to accurately estimate 2D baseball swing poses. Monocular 2D images captured by a monocular camera are inputted into the MSH networks to estimate the 2D keypoint coordinates of human poses. The proposed MSH networks are validated on the Max Planck Institute for Informatics (MPII) Human Pose dataset to prove their feasibility and effectiveness for 2D human pose estimation. In addition, the proposed MSH network trained by the MPII Human Pose dataset is utilized to estimate the 2D keypoint coordinates of the baseball swing poses from the monocular 2D images. The experimental results show that the MSH networks achieve average percentages of correct keypoints (PCK)@0.5 of 90.2 and 94.0% for the MPII Human Pose dataset and baseball swing motions, respectively.

1. Introduction

Human pose estimation has long been recognized as a fundamental yet challenging problem in computer vision over the past several decades, owing to its wide range of applications in human–computer interaction, sports analytics, healthcare, and surveillance.[1,2] It aims to automatically predict and track human posture by localizing anatomical keypoints in RGB images or videos, and by estimating the spatial orientation of the corresponding limbs. The anatomical keypoints refer to specific landmarks on the human body that represent important body parts, including the head and joints (e.g., elbows, wrists, knees, and ankles). These keypoints can then be used to reconstruct the human skeleton and posture, enabling many applications in computer vision domains. Early studies predominantly relied on traditional

*Corresponding author: e-mail: hsuyl@mail.nsysu.edu.tw
https://doi.org/10.18494/SAM5947

ISSN 0914-4935

image processing techniques and shallow machine learning models to localize human anatomical keypoints. These methods typically employed handcrafted features, such as edges, contours, or gradient-based descriptors, to infer body joint positions. While such approaches provided initial insights, they were constrained by limited accuracy and robustness, particularly under complex conditions involving background clutter, illumination changes, occlusions, or varying body shapes. The rapid development of deep learning has markedly transformed the field. In particular, convolutional neural networks (CNNs) have enabled models to automatically learn hierarchical feature representations from raw images and videos. This capability has significantly improved both the accuracy and efficiency of human pose estimation. Moreover, recent advancements in computing hardware, such as high-performance GPUs and CPUs, have reduced the computational burden of training and inference, allowing more complex and deeper network architectures to be deployed effectively. These technological advances have accelerated progress in human pose estimation, making it a vibrant area of study that continues to evolve rapidly.

Human pose estimation has found widespread applications across diverse domains, including film production, animation, action recognition, human–computer interaction, autonomous driving, medical rehabilitation, and sports analysis.[3,4] In film production and animation, motion capture systems combined with pose estimation techniques enable the transformation of actors' movements into digital skeletal models, thereby facilitating the creation of more realistic and fluid visual effects and animations. In action recognition, human pose estimation can be employed to analyze and classify human activities, such as fall detection or daily activity monitoring of elderly individuals, which further allows the identification of abnormal events to enhance personal safety.[5] In the context of human–computer interaction, human pose estimation provides a means of accurately capturing body movements through cameras, enabling more natural interaction with virtual environments and enhancing user immersion in virtual reality (VR), augmented reality (AR), and gaming applications. For autonomous driving, human pose estimation contributes to predicting pedestrian movement trajectories and assisting self-driving systems in making timely decisions such as evasive maneuvers or braking, thereby improving both vehicle and pedestrian safety.[6] In medical rehabilitation, human pose estimation facilitates the remote assessment of patients' joint angles and postures during rehabilitation exercises, enabling healthcare professionals to deliver personalized guidance and adapt treatment plans accordingly.[7] Moreover, gait cycle analysis based on human pose estimation has been leveraged to support the diagnosis of specific pathological conditions.[8] In the field of sports analysis, human pose estimation allows the quantitative evaluation of athletes' motion parameters and joint kinematics, providing valuable insights into performance during training or competition. This enables athletes and coaches to optimize training strategies in real time, ultimately enhancing performance while reducing the risk of injury.[9]

The earliest application of deep learning techniques to human pose estimation was proposed by Toshev and Szegedy, who introduced the DeepPose model based on CNNs.[10] This approach can directly yield the 2D keypoint locations by regression computation and achieve promising results. Since then, the powerful representational capacity of the convolutional operations has driven the development of numerous deep-learning-based models for human pose estimation. In

particular, 2D human pose estimation refers to tasks of predicting the coordinates of joint keypoints on 2D images using deep learning models. 2D human pose estimation can generally be categorized into two approaches based on the computation of loss. (1) The first approach directly regresses the predicted keypoint coordinates against the ground-truth annotations.[11] Since the direct regression of individual points constitutes a highly nonlinear problem, model training becomes more difficult and the resulting models tend to be less robust. Nevertheless, this method offers lower computational complexity because it predicts the keypoint coordinates directly. (2) The second approach generates the ground-truth heatmaps by applying Gaussian distributions to each annotated keypoint and then computes the loss between the predicted and ground-truth heatmaps.[12] When using the heatmap-based loss functions, the heatmap representation is more effective in capturing both the spatial relationships and inter-frame dependences in the image sequences, thereby providing more stable supervision that facilitates training convergence. However, because the resolution of heatmaps is typically lower than that of the original image, the final keypoint coordinates must be obtained through additional interpolation from the heatmaps. As a result, while the heatmap-based methods achieve higher robustness, they also introduce more computational complexity.

Moreover, 2D human pose estimation can also be categorized into single-person and multiperson pose estimation. In the former, the location of the target individual is assumed to be known, and the task is limited to inferring the person's body pose. In contrast, the latter requires simultaneously determining the number of individuals in the scene, localizing each of them, and estimating all corresponding body joints. Newell *et al.* proposed the stacked hourglass model, which is a 2D single-person pose estimation framework.[13] This model first identifies the center and scale of the target individual in the image to obtain the bounding box. The cropped region is then processed by the feature pyramid-like architecture, where the image undergoes multiscale resizing, feature extraction, and feature fusion. Through iterative refinement, the model produces the heatmaps representing the coordinates of the individual's body keypoints, ultimately achieving highly accurate estimation for 2D single-person poses. On the other hand, Cao *et al.* proposed the OpenPose model, which is a 2D multiperson pose estimation model that employs the visual geometry group (VGG)-19 model as the backbone network for the initial feature extraction, and the framework subsequently branches into the keypoint detection and part affinity fields (PAFs).[14] The keypoint detection branch iteratively refines the predicted heatmaps across multiple stages, with each stage improving upon the results of the previous one. Meanwhile, the PAFs are defined as the vector fields that encode the associations between the pairs of body parts. By jointly predicting the keypoints and PAFs, the OpenPose model can estimate the poses of the multiple individuals in the image and correctly assemble the detected keypoints into the coherent person-specific skeletons.

On the other hand, 2D multiperson pose estimation methods can generally be classified into two categories: top-down and bottom-up approaches.[15] In the top-down methods, a person detector is first applied to an image to generate bounding boxes for each individual, after which keypoints are localized within each cropped region, such as the HRNet and TokenPose models.[16,17] In contrast, the bottom-up methods detect all keypoints present in an image and subsequently group them into individual poses, such as PifPaf and DEKR models.[18,19] In

addition, in some studies, entire video sequences are input into recurrent neural networks (RNNs) or long short-term memory (LSTM) models, which exploit the temporal information across consecutive frames in conjunction with features extracted by CNNs to estimate 2D keypoint coordinates.[20,21]

In our literature review, we found that most 2D human pose estimation methods adopt heatmap-based loss functions owing to their robustness and high accuracy. However, there is a notable lack of literature applying such techniques to the pose estimation of baseball swing motions. In this work, multistacked hourglass (MSH) networks have been utilized for 2D human pose estimation. The MSH networks first perform feature extraction using a convolutional layer with a 7 × 7 kernel and a stride of 2, followed by multiple residual blocks and a max-pooling layer. Subsequently, the MSH networks employ hourglass networks, which consist of downsampling via max-pooling layers, intermediate processing via residual blocks, and upsampling using nearest-neighbor interpolation modules, to perform 2D keypoint estimation. The proposed MSH networks are validated on the Max Planck Institute for Informatics (MPII) Human Pose dataset to prove their feasibility and effectiveness for 2D human pose estimation. In addition, the proposed MSH network trained by the MPII Human Pose dataset is utilized to estimate the 2D keypoint coordinates of the baseball swing poses from the monocular 2D image sequences.

The rest of this paper is organized as follows: in Sect. 2, the architectures of the MSH networks are described in detail; the experimental setup and results are presented in Sect. 3; finally, the conclusions are given in Sect. 4.

2. MSH Networks

In this paper, we realize the 2D human pose estimation models using the MSH networks. For 2D human pose estimation, monocular RGB images captured by a monocular camera are inputted into the MSH networks for obtaining the heatmaps of the joint keypoints, which can be used to estimate the 2D keypoint coordinates of the human poses through additional interpolation from the heatmaps. With the heatmap-based method, the MSH networks can achieve better robustness and higher accuracy for 2D human pose estimation.

2.1 Stacked hourglass networks

The stacked hourglass network is a widely adopted framework for 2D human pose estimation and predicts joint keypoints from input images.[13] The hourglass network is designed to extract information at every scale. Cues such as the overall body orientation, limb arrangement, and adjacent joint relationships are most effectively recognized at distinct image scales. The hourglass architecture is a compact design that captures multiscale features and fuses them to produce pixel-wise predictions. The stacked hourglass network comprises a stem block and a stacked hourglass subnetwork. The stem block begins with a 7 × 7 convolution layer with stride 2, followed by a residual block, a max-pooling layer, and two additional residual blocks, thereby

reducing a 256 × 256 input image to a 64 × 64 feature map. The stacked hourglass subnetwork comprises multiple hourglass networks stacked in a series. Each hourglass network includes a max-pooling layer, residual blocks, and a nearest-neighbor interpolation module, as shown in Fig. 1. After each max-pooling operation, the resolution and computational burden of the feature maps is reduced. Then, the image features are extracted from the downsampled feature map by the residual blocks. The nearest-neighbor interpolation module is applied to upsample the feature map and restore higher resolution. The features extracted by the hourglass network integrate multiscale and contextual information, preserve information from all layers, and match the original image size. The network outputs a set of heatmaps that represent the predicted probability of joint presence at each and every pixel. Finally, the argmax in each heatmap is taken as the 2D coordinate of the corresponding joint keypoint.

For the MSH networks, the residual block inside the dashed box in Fig. 1 can be replaced by a new hourglass network to form a multilayer hourglass network; for example, a four-layer hourglass network is shown in Fig. 2. The number of layers denotes the counts of downsampling and upsampling operations. Increasing the layer depth generally improves 2D keypoint estimation accuracy. In addition, multiple hourglass networks can be further cascaded to form the MSH network, and additional stacks typically yield higher accuracy for 2D human pose estimation; for example, a two-stacked hourglass network is shown in Fig. 3. For the MSH architecture, each hourglass network is supervised with its own loss against the ground-truth 2D keypoint locations via the intermediate supervision. An overview of the MSH network used in

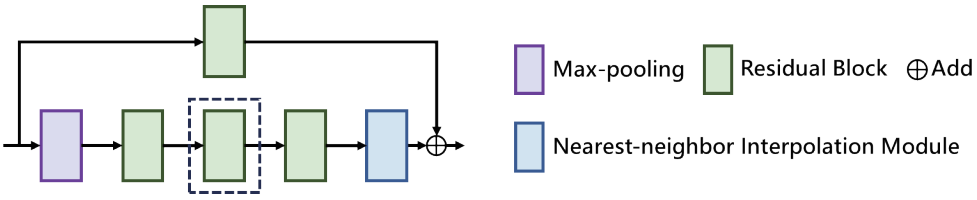

Fig. 1. (Color online) Hourglass network architecture.

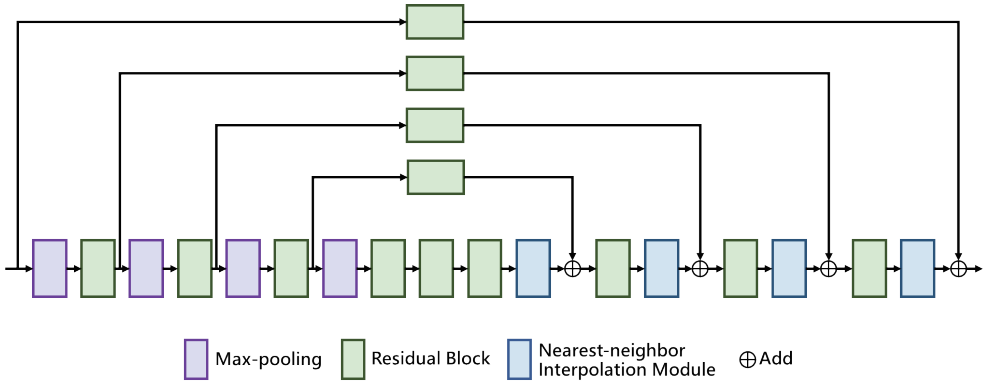

Fig. 2. (Color online) A four-layer hourglass network architecture.

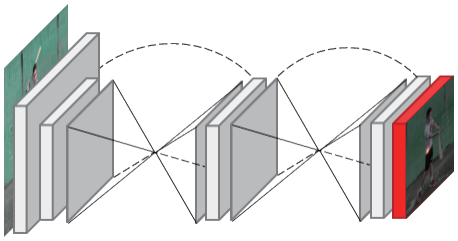

Fig. 3. (Color online) A two-stacked hourglass network architecture.

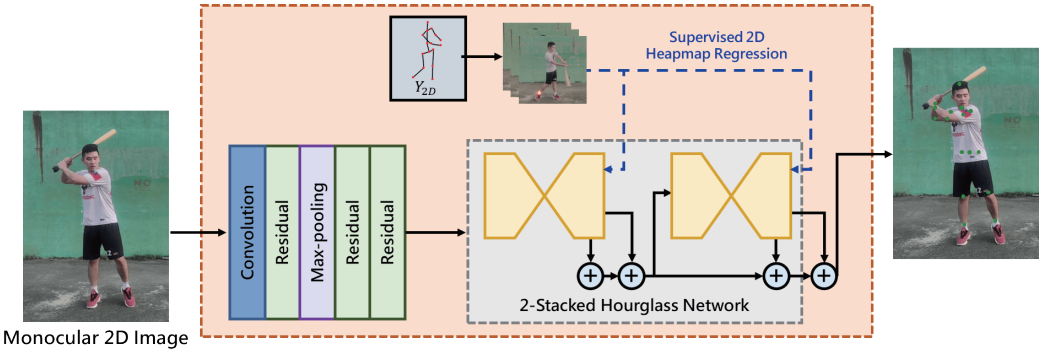

Fig. 4. (Color online) Overview of the MSH network used in this study.

this study is shown in Fig. 4. We now introduce the residual block, nearest-neighbor interpolation module, intermediate supervision, and loss function in detail as follows.

2.2 Residual block

In the stacked hourglass model, the residual block adopts a bottleneck-like structure, which facilitates effective information propagation in deeper networks by incorporating the original input, thereby reducing the likelihood of gradient vanishing or explosion. As illustrated in Fig. 5, the residual block is composed of batch normalization layers, ReLU activation functions, and convolutional layers. The upper branch employs a 1 × 1 convolution layer to align the dimensionality of the input with that of the output. In the lower branch, a 1 × 1 convolution layer is first applied to reduce dimensionality, followed by a 3 × 3 convolution layer to extract the salient features, and finally another 1 × 1 convolution layer to restore the output to the appropriate dimensionality. The fusion of the original input with the output forms the residual connection, which effectively mitigates the problems of gradient vanishing and gradient explosion. The operation of the residual block is defined as

$$\hat{\boldsymbol{X}}^{l} = F(\boldsymbol{X}^{l} + T(\boldsymbol{X}^{l-1})), \tag{1}$$

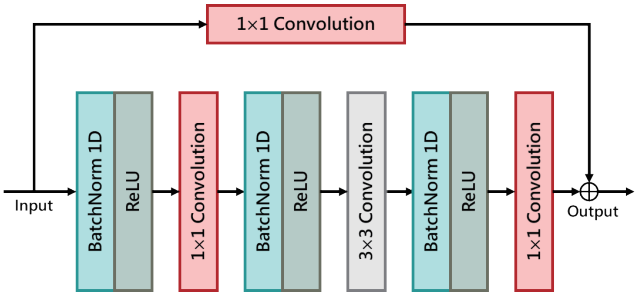

Fig. 5. (Color online) A residual block architecture.

where $\hat{\boldsymbol{X}}^{l}$ denotes the feature map obtained after the residual block operation, $\boldsymbol{X}^{l}$ represents the feature map produced by the convolution operation, $\boldsymbol{X}^{l-1}$ corresponds to the feature map prior to entering the residual block, and T denotes the transformation function, where a 1 × 1 convolution layer is applied to $\boldsymbol{X}^{l-1}$ when its channel dimension differs from that of $\boldsymbol{X}^{l}$, otherwise, it remains identical to the input feature.

2.3 Nearest-neighbor interpolation module

Nearest-neighbor interpolation is a simple method that increases resolution by replicating each pixel value to adjacent locations, thereby preserving the original image content. In the stacked hourglass architecture, it is employed for upsampling to restore the resolution of the feature maps.

2.4 Intermediate supervision

In the MSH network, the output of each hourglass network first passes through a residual block and is then divided into two branches. The upper branch continues feature extraction through an additional residual block, while the lower branch applies a 1×1 convolution layer to reduce the feature map from 256 channels to 16 channels. At this stage, each channel corresponds to the heatmap of one of the 16 joint keypoints. These heatmaps are subsequently compared with the ground-truth 2D keypoint heatmaps to compute the loss and perform backpropagation, thereby mitigating the problem of gradient vanishing. The feature map is then passed through another residual block to restore the channel dimension to 256, after which it is fused with the features obtained from the two upper residual blocks and the original input feature map. The architecture of the intermediate supervision of the MSH network is shown in Fig. 6. For each input image, every hourglass network generates the 16 heatmaps corresponding to the 2D joint keypoints. Each heatmap represents the probability distribution of the specific joint keypoint, where brighter regions indicate a higher likelihood of the keypoint location, whereas darker regions indicate a lower likelihood.

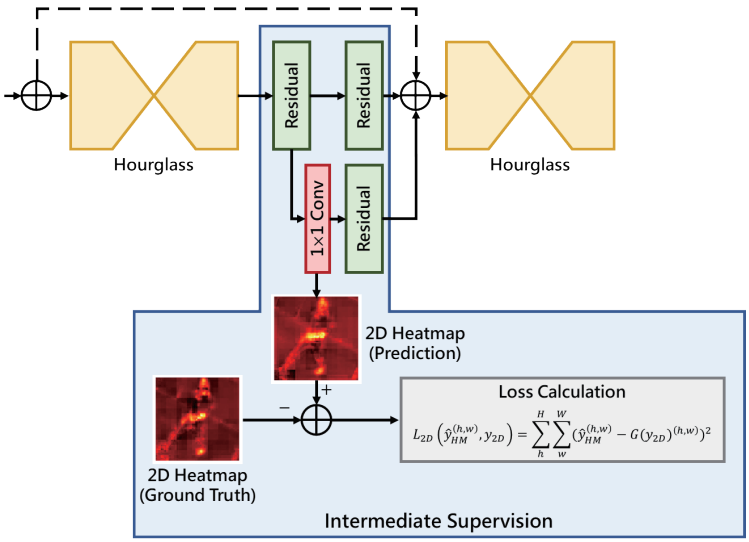

Fig. 6. (Color online) Architecture of the intermediate supervision of the MSH network.

2.5 Loss function

In the MSH network, the 2D heatmaps of the joint keypoints estimated by each hourglass network are compared pixel-wise with the corresponding 2D ground-truth heatmaps, which are generated using Gaussian kernels. This process is called intermediate supervision, which can prevent gradient vanishing and optimizes the predictive performance of the network. Through this supervision strategy, the MSH network is trained to estimate the 2D keypoint heatmaps, where the pixel with the maximum likelihood in each heatmap is taken as the predicted 2D coordinates of the corresponding joint keypoints. The loss function is defined as

$$L_{2D}(\hat{y}_{HM}^{(h,w)}, y_{2D}) = \sum_{h}^{H} \sum_{w}^{W} (\hat{y}_{HM}^{(h,w)} - G(y_{2D})^{(h,w)})^2, \tag{2}$$

where $\hat{y}_{HM}^{(h,w)}$ denotes the pixel of the predicted 2D keypoint heatmap at pixel location (h, w), $G(y_{2D})^{(h,w)}$ represents the pixel of the ground-truth 2D keypoint heatmap generated by Gaussian kernels at pixel location (h, w), and $L_{2D}(\hat{y}_{HM}^{(h,w)}, y_{2D})$ is the loss function computed between the two heatmaps.

3. Experimental Results and Discussion

In this study, we implemented the MSH networks for estimating the 2D human pose from the monocular 2D images. The 2D human pose estimation performance of the proposed MSH networks was first validated using the MPII Human Pose dataset. Subsequently, the proposed MSH network trained by the MPII Human Pose dataset was utilized to estimate the 2D keypoint coordinates of the baseball swing poses from the monocular 2D image sequences.

3.1 Experimental setup

The hardware used for the training and validation of the proposed MSH networks consist of a workstation equipped with an Intel Xeon W-2245 CPU, an NVIDIA GeForce RTX 4090 GPU, and 256 GB of memory. On the software side, the MSH networks were implemented using Python 3.12.3, CUDA version 12.5, and PyTorch version 2.3.1. In addition, the monocular image sequences of baseball swing motions were collected by a camera (MV-CS016-10UM, resolution of 1440 × 1080 pixels, frame rate of 249 Hz).

3.2 MPII human pose dataset

The MPII Human Pose dataset is one of the most widely used benchmarks for 2D human pose estimation.[22] It consists of approximately 25000 images extracted from a variety of YouTube videos, with manually annotated 2D keypoints. In total, the dataset includes more than 40000 annotated human instances, each labeled with 16 joint keypoints: right ankle, right knee, right hip, left hip, left knee, left ankle, pelvis, thorax, upper neck, head top, right wrist, right elbow, right shoulder, left shoulder, left elbow, and left wrist. Among these annotated instances, roughly 28000 instances are designated for training, while the remaining 11000 instances are reserved for testing. The images cover over 410 different human activities in daily life, each associated with an activity label, and exhibit variations such as scale differences, partial cropping, and occlusions. This diversity enhances the robustness of networks trained on the dataset. In this paper, the MPII Human Pose dataset was employed to train and validate the performance of the MSH networks. In addition, the percentage of correct keypoints (PCK) metric was employed to evaluate the performance of 2D human pose estimation for the MPII Human Pose dataset. In this paper, we used the PCK at a threshold of 0.5 (PCK@0.5) as the evaluation criterion for assessing the performance of 2D human pose estimation.

3.3 2D human pose estimation for MPII human pose dataset

In this work, the MPII Human Pose dataset was utilized to evaluate and validate the performance of the MSH networks for 2D human pose estimation. The related experiments include (1) performance comparison between different types of stacked hourglass networks, (2) performance comparison between different training iterations, (3) performance comparison between different learning rates, and (4) performance comparison of the MSH networks with existing models.

3.3.1 Performance comparison between different types of stacked hourglass networks

For the performance comparison between different types of stacked hourglass networks, in this study, we employed the 4-layer hourglass networks with the 2-, 4-, and 8-stack hourglass networks to investigate the impact of network depth on 2D human pose estimation performance. Table 1 shows that the PCK@0.5 values for the 2-, 4-, and 8-stack with 4-layer MSH networks are 88.3, 89.8, and 90.2%, respectively. These experimental results clearly indicate that

Table 1
Performance of different types of stacked hourglass networks for 2D human pose estimation.

MSH networks	PCK@0.5 (%)	Number of parameters (M)	Training time (day)
2-Stack with 4-Layer	88.3	8.4	1
4-Stack with 4-Layer	89.8	16.5	2
8-Stack with 4-Layer	90.2	32.9	4

increasing the number of stacked hourglass networks leads to improved 2D human pose estimation accuracy. However, this improvement comes at the cost of a proportional increase in the number of parameters, which in turn results in significantly longer training times. Although the 8-stack with 4-layer MSH network achieved higher estimation accuracy, the 2-stack with 4-layer MSH configuration was adopted in the following experiments to evaluate 2D human pose estimation performance, in order to balance accuracy with computational cost.

3.3.2 Performance comparison between different training iterations

For the performance comparison between different training iterations, the 2-stack with 4-layer MSH network was employed to examine the effect of different numbers of training epochs on 2D human pose estimation. The results are presented in Table 2, and the corresponding learning curve under different training epochs is shown in Fig. 7. As shown in Table 2, the PCK@0.5 values for 50, 100, 150, 200, 250, 300, and 350 training epochs are 84.9, 87.6, 88.1, 88.3, 88.3, 88.4, and 88.5%, respectively. The experimental results clearly demonstrate that increasing the number of training epochs leads to improved 2D human pose estimation accuracy. However, after 200 epochs, the improvement in PCK@0.5 begins to plateau. Therefore, in this study, we adopted 200 epochs as the optimal number of training iterations for the 2-stack with 4-layer MSH network.

3.3.3 Performance comparison between different learning rates

For the performance comparison between different learning rates, the 2-stack with 4-layer MSH network trained with 200 epochs was employed to investigate the impact of different learning rates on 2D human pose estimation. The results are summarized in Table 3. As shown, under the configuration of 200 training epochs, the PCK@0.5 values corresponding to learning rates of 0.01, 0.001, 0.0001, and 0.00001 are 87.4, 88.3, 87.3, and 86.4%, respectively. These results clearly indicate that the optimal 2D human pose estimation performance was achieved with a learning rate of 0.001.

3.3.4 Performance comparison of MSH networks with existing models

We compared the performances of the proposed 2-stack with 4-layer, 4-stack with 4-layer, and 8-stack with 4-layer MSH networks with the existing 2D human pose estimation models reported in the literature. The evaluation was conducted on the MPII Human Pose dataset with respect to 13 joint keypoints, including head, shoulders (right and left), elbows (right and left), wrists (right and left), hips (right and left), knees (right and left), and ankles (right and left), as

Table 2
Performance of 2D human pose estimation at different training epochs for the 2-stack with 4-layer MSH network.

Training epoch	PCK@0.5 (%)
50	84.9
100	87.6
150	88.1
200	88.3
250	88.3
300	88.4
350	88.5

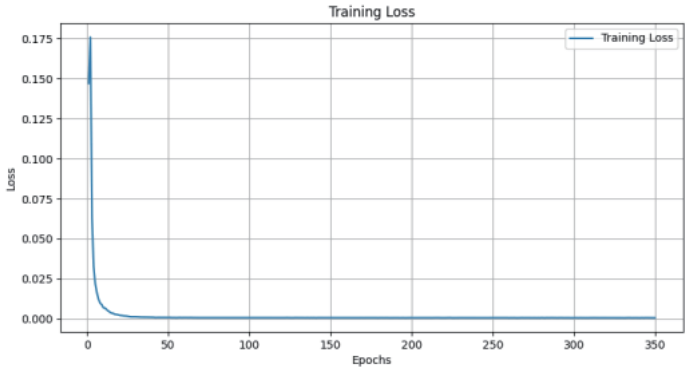

Fig. 7. (Color online) Learning curve of the 2-stack with 4-layer MSH network for 2D human pose estimation.

Table 3
Performance of 2D human pose estimation with different learning rates for the 2-stack with 4-layer MSH network.

Learning rate	PCK@0.5 (%)
0.01	87.4
0.001	88.3
0.0001	87.3
0.00001	86.4

Table 4
Performance of the proposed MSH networks and the existing 2D human pose estimation models on the MPII Human Pose dataset.

Model	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Average
2-stack with 4-layer MSH	95.5	94.7	88.1	82.7	87.1	81.3	77.9	88.3
4-stack with 4-layer MSH	95.8	95.2	89.5	84.2	88.4	83.5	79.0	89.4
8-stack with 4-layer MSH	96.5	95.7	90.3	85.0	92.6	84.6	80.9	90.2
Liang *et al*. 2018[23]	97.5	94.3	87.0	81.2	86.5	78.5	78.5	86.4
Wei *et al*. 2016[24]	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Insafutdinov *et al*. 2016[25]	96.8	95.2	89.3	84.4	88.4	83.4	78.5	88.5
Li *et al*. 2022[26]	97.2	96.0	90.4	85.6	89.5	85.8	81.8	90.0

summarized in Table 4. The average PCK@0.5 values achieved by the 2-stack with 4-layer, 4-stack with 4-layer, and 8-stack with 4-layer MSH networks are 88.3, 89.4, and 90.2%, respectively. For comparison, Liang *et al.* proposed the structure-aware regression approach, which represents the poses using the skeletal structures and defines the composite loss function

based on keypoint connectivity, achieving an average PCK@0.5 of 86.4%.[23] Wei *et al.* proposed the convolutional pose machine, which integrated the convolutional layers with the sequential prediction framework of the pose machines to progressively learn image and spatial features for coarse-to-fine 2D human pose estimation.[24] By employing the structured prediction, the method implicitly modeled long-range dependences among variables, achieving an average PCK@0.5 of 88.5%.[24] Insafutdinov *et al.* proposed the novel bottom-up body part detector augmented with the image-conditioned pairwise terms to estimate the associations between the keypoints.[25] Furthermore, the incremental optimization strategy was employed to improve both accuracy and efficiency. This approach achieved an average PCK@0.5 of 88.5%.[25] Li *et al.* reformulated human keypoint localization as a classification task, where each pixel was treated as a distinct class.[26] The CNN was employed as the feature extractor, and the extracted features were subsequently fed into the vertical and horizontal coordinate classifiers to obtain the 2D keypoint positions. This method achieved an average PCK@0.5 of 90.0%.[26] From these results, it is evident that the proposed MSH networks can achieve competitive 2D human pose estimation performance for the MPII Human Pose dataset.

3.4 Monocular 2D baseball swing pose estimation

In this study, the camera (MV-CS016-10UM, resolution of 1440 × 1080 pixels, frame rate of 249 Hz) was utilized to collect monocular 2D image sequences of baseball swing motions from 10 subjects in lateral (side) and posterior (rear) views. For each view, 10 repetitions per subject were recorded. Ethics approval was obtained from the institutional review board (IRB) of Chung Shan Medical University Hospital (IRB No. CS1-24083). The 2-stack with 4-layer MSH network trained by the MPII Human Pose dataset with 200 epochs and learning rate of 0.001 was utilized to estimate the 2D human pose for baseball swing motions. We employed the 2D keypoint coordinates of the 16 joints defined on the MPII Human Pose dataset for the 2D human pose estimation; such as right ankle, right knee, right hip, left hip, left knee, left ankle, pelvis, thorax, upper neck, head top, right wrist, right elbow, right shoulder, left shoulder, left elbow, and left wrist. The ground-truth 2D keypoints of the 16 joints on all the images were manually annotated to evaluate the performance of 2D baseball swing pose estimation. Table 5 shows the average PCK@0.5 for 2D keypoint estimation by the 2-stack with 4-layer MSH network across the full swing sequences of 10 baseball batters. The average PCK@0.5 values for head, shoulder, elbow, wrist, hip, knee, ankle, pelvis, upper neck, and thorax are 99.2, 98.6, 91.5, 93.9, 94.5, 98.9, 98.8, 95.5, 98.7, and 99.7%, respectively. The overall average PCK@0.5 across all keypoints is 96.6%. Table 6 shows the average PCK@0.5 for 2D keypoint estimation by the 2-stack with 4-layer MSH network for each of the 10 batters' swing sequences. The average PCK@0.5 values across all keypoints for subjects S01–S10 are 97.5, 98.4, 96.5, 97.8, 93.6, 95.0, 98.1, 97.8, 93.0, and 97.2%, respectively. As shown in Table 6, the model achieves its best performance on subject S02. The swing sequences of subject S02 are shown in Fig. 8, where red denotes manually annotated 2D keypoints (ground truth) and green indicates the 2D keypoints estimated by the 2-stack with 4-layer MSH network. The heatmaps of each joint on the swing sequence (f) in Fig. 8 of subject S02 are shown in Fig. 9. These results indicate that the 2-stack with 4-layer MSH

Table 5
Average PCK@0.5 for each 2D joint keypoint using the 2-stack with 4-layer MSH network.

Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Pelvis	Neck	Thorax	Average
99.2	98.6	91.5	93.9	94.5	98.9	98.8	95.5	98.7	99.7	96.6

Table 6
Average PCK@0.5 estimated by the 2-stack with 4-layer MSH network for each of the 10 batters' swing sequences.

Subject	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Pelvis	Neck	Thorax	Average
S01	100	98.0	91.8	95.0	98.2	99.1	98.5	99.4	99.7	99.7	97.5
S02	99.6	100	93.9	93.7	100	100	100	100	99.0	100	98.4
S03	98.7	97.5	93.1	91.1	95.1	99.3	100	94.6	97.7	100	96.5
S04	97.4	98.9	95.1	95.9	97.2	99.6	99.2	98.1	98.9	98.5	97.8
S05	98.4	96.2	82.2	91.2	93.0	95.4	95.2	92.8	99.2	100	93.6
S06	99.1	99.3	85.9	93.0	87.8	98.8	97.7	87.8	98.1	100	95.0
S07	100	100	95.8	947.6	96.2	99.3	100	98.2	100	100	98.1
S08	99.5	100	96.0	96.8	97.6	100	98.9	98.9	100	100	97.8
S09	99.0	97.8	89.4	93.5	79.5	97.8	98.6	83.1	94.2	98.5	93.0
S10	100	98.8	89.1	95.1	95.7	100	99.4	98.3	100	100	97.2

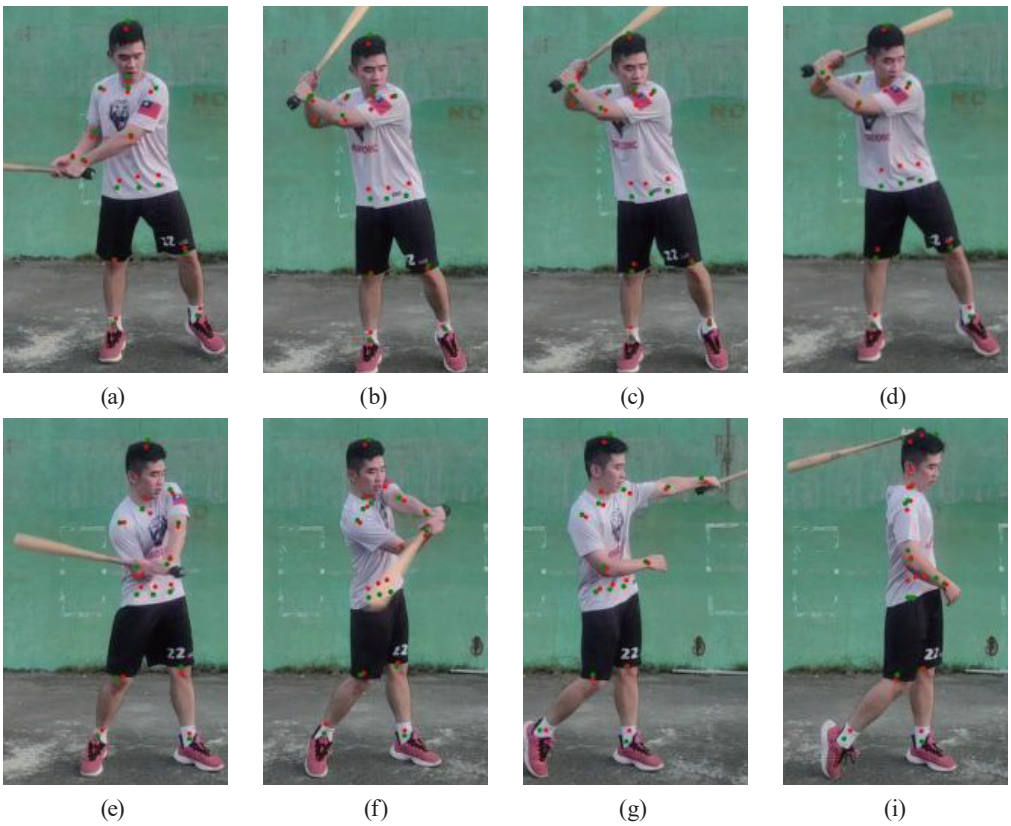

Fig. 8. (Color online) Experimental results of the 2D baseball swing pose estimation. Red color: Manually annotated 2D keypoints (ground truth). Green color: 2D keypoints estimated by the 2-stack with 4-layer MSH network. (a) Initial rest, (b) wind up, (c)stride, (d) foot down, (e) down swing, (f) impact, (g) follow through, and (i) ending rest.

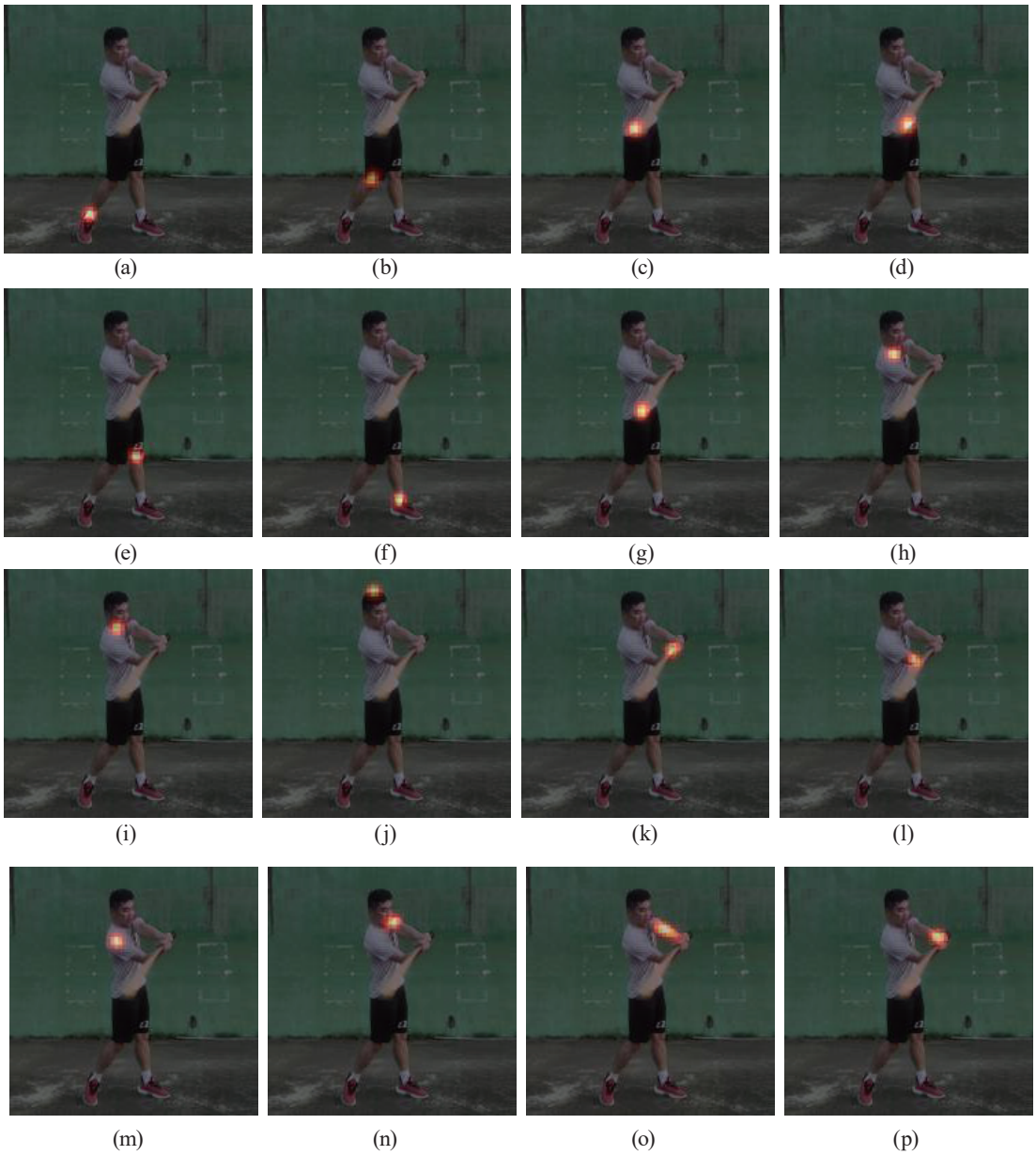

Fig. 9. (Color online) Heatmaps of each joint on the baseball swing sequence (f) in Fig. 8. of subject S02. (a) Right ankle. (b) Right knee. (c) Right hip. (d) Left hip. (e) Left knee. (f) Left ankle. (g) Pelvis. (h) Thorax. (i) Upper neck. (j) Head top. (k) Right wrist. (l) Right elbow. (m) Right shoulder. (n) Left shoulder. (o) Left elbow. (p) Left wrist.

network employed in this study achieves good performance for 2D keypoint estimation on the baseball swing sequences.

4. Conclusions

In this study, the MSH networks were implemented to accurately estimate 2D human pose. The monocular 2D images captured by the camera were input to the MSH networks to estimate

the 2D keypoint coordinates of human poses. The MPII Human Pose dataset was used to validate the performance of the proposed MSH networks for 2D human pose estimation. The PCK@0.5 values for the 2-stack with 4-layer, 4-stack with 4-layer, and 8-stack with 4-layer MSH networks were 88.3, 89.8, and 90.2%, respectively. In addition, the proposed 2-stack with 4-layer MSH network trained by the MPII Human Pose dataset was utilized to estimate the 2D keypoint coordinates of the baseball swing poses from the monocular 2D images. The average PCK@0.5 values for head, shoulder, elbow, wrist, hip, knee, ankle, pelvis, upper neck, and thorax were 99.2, 98.6, 91.5, 93.9, 94.5, 98.9, 98.8, 95.5, 98.7, and 99.7%, respectively. The overall average PCK@0.5 across all keypoints is 96.6%. Clearly, the experimental results successfully validated the effectiveness of the proposed MSH networks for 2D human pose estimation.

Acknowledgments

This work is supported by the National Science and Technology Council, Taiwan, under Grant Nos. MOST 111-2628-E-110-012-MY3 and NSTC 113-2223-E-110-002-MY3.

References

1 Y. Chen, Y. Tian, and M. He: Comput. Vision Image Understanding **192** (2020) 102897. https://doi.org/10.1016/j.cviu.2019.102897
2 W. Liu, Q. Bao, Y. Sun, and T. Mei: ACM Comput. Surv. **55** (2022) 80. https://doi.org/10.1145/3524497
3 C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, R. Liu, and M. Shah: ACM Comput. Surv. **56** (2024) 11. https://doi.org/10.1145/3603618
4 J. Wang, S. Tan, X. Zhen, S. Xu, F. Zheng, Z. He, and L. Shao: Comput. Vision Image Understanding **210** (2021) 103225. https://doi.org/10.1016/j.cviu.2021.103225
5 W. Chang, C. Hsu, and L. Chen: IEEE Access **9** (2021) 129965. https://doi.org/10.1109/ACCESS.2021.3113824
6 A. Zanfir, M. Zanfir, A. Gorban, J. Ji, Y. Zhou, D. Anguelov, and C. Sminchisescu: Proc. 6th Conf. Robot Learning (CoRL 2022) 1. https://doi.org/10.48550/arXiv.2212.07729
7 J. Stenum, M. K. Cherry-Allen, C. O. Pyles, R. D. Reetzke, M. F. Vignos, and R. T. Roemmich: Sensors **21** (2021) 7315. https://doi.org/10.3390/s21217315
8 D. Sethi, S. Bharti, and C. Prakash: Artif. Intell. Med. **129** (2022) 102314. https://doi.org/10.1016/j.artmed.2022.102314
9 L. Citraro, P. Marquez-Neila, S. Savare, V. Jayaram, C. Dubout, F. Renaut, A. Hasfura, H. B. Horesh, and P. Fau: Mach. Vision Appl. **31** (2020) 16. https://doi.org/10.1007/s00138-020-01064-7
10 A. Toshev and C. Szegedy: Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2014) 1653. https://doi.org/10.1109/CVPR.2014.214
11 F. Wei, X. Sun, H. Li, J. Wang, and S. Lin: Proc. European Conf. Computer Vision (ECCV 2020) 527. https://doi.org/10.48550/arXiv.2007.02846
12 J. Li and M. Wang: IEEE Trans. Circuits Syst. Video Technol. **32** (2022) 5521. https://doi.org/10.1109/TCSVT.2022.3153044
13 A. Newell, K. Yang, and J. Deng: Proc. European Conf. Computer Vision (ECCV 2016) 483. https://doi.org/10.48550/arXiv.1603.0693
14 Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh: IEEE Trans. Pattern Anal. Mach. Intell. **43** (2021) 172. https://doi.org/10.1109/TPAMI.2019.2929257
15 M. B. Gamra and M. A. Akhloufi: Image Vision Comput. **114** (2021) 104282. https://doi.org/10.1016/j.imavis.2021.104282
16 K. Sun, B. Xiao, D. Liu, and J. Wang: Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (IEEE, 2019) 5686. https://doi.org/10.1109/CVPR.2019.00584
17 Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.-T. Xia, and E. Zhou: Proc. 2021 IEEE/CVF Int. Conf. Computer Vision (IEEE, 2021) 11293. https://doi.org/10.1109/ICCV48922.2021.01112

18 S. Kreiss, L. Bertoni, and A. Alahi: Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (IEEE, 2019) 11969. https://doi.org/10.1109/CVPR.2019.01225
19 Z. Geng, K. Sun, B. Xiao, Z. Zhang, and J. Wang: Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition (IEEE, 2021) 14671. https://doi.org/10.1109/CVPR46437.2021.01444
20 Z. Fan, J. Liu, and Y. Wang: Proc. 2021 IEEE/CVF Int. Conf. Computer Vision (IEEE, 2021) 11699. https://doi.org/10.1109/ICCV48922.2021.01151
21 X. Nie, Y. Li, L. Luo, N. Zhang, and J. Feng: Proc. 2019 IEEE/CVF Int. Conf. Computer Vision (IEEE, 2019) 6941. https://doi.org/10.1109/ICCV.2019.00704
22 M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele: Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2014) 3686. https://doi.org/10.1109/CVPR.2014.471
23 S. Liang, X. Sun, and Y. Wei: Comput. Vision Image Understanding **176** (2018) 1. https://doi.org/10.1016/j.cviu.2018.10.006
24 S. Wei, V. Ramakishna, T. Kanade, and Y. Sheikh: Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016) 4724. https://doi.org/10.1109/CVPR.2016.511
25 E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele: Proc. European Conf. Computer Vision (ECCV 2016) 34. https://doi.org/10.48550/arXiv.1605.03170
26 Y. Li, S. Yang, P. Liu, S. Zhang, Y. Wang, Z. Wang, W. Yang, and S. Xia: Proc. European Conf. Computer Vision (ECCV 2022) 89. https://doi.org/10.48550/arXiv.2107.03332

About the Authors

Yu-Liang Hsu received his B.S. degree in automatic control engineering from Feng Chia University, Taichung, Taiwan, in 2004, and his M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2007 and 2011, respectively. He is currently an assistant professor in the Department of Mechanical and Electro-Mechanical Engineering and a joint assistant professor in the Department of Electrical Engineering, National Sun Yat-sen University (NSYSU), Taiwan. His research interests include wearable intelligent technology, microsensor design, artificial intelligence, human–robot interaction, intelligent rehabilitation assistive devices, and sport sciences. He received the Wu Ta-You Memorial Award from the Ministry of Science and Technology Council in 2022, Outstanding Young Scholar Award from the Ministry of Science and Technology Council in 2019 and 2022, Phi Tau Phi Scholastic Honor in 2004, Outstanding Young Faculty from the Kaohsiung Chapter of the Chinese Society of Mechanical Engineering (CSME) in 2024, Young Scholar Award in Mechatronics Engineering and Technology from the International Society of Mechatronic Engineering (ISME) in 2023, and Best Advisor Award during the 18th, 21st, and 23rd Macronix Golden Silicon Awards in 2018, 2021, and 2023, respectively. He was also elected as the Distinguished Junior Research Professor at NSYSU in 2022. He received the Best Paper Award of the 2012 Conference on ISG*ISARC, the Merit Paper Award of TAAI 2013, the First Prize Paper Award of IEEE ICASI 2017, the Best Conference Paper Award of IEEE ICASI 2018, 2023, 2024, and 2025, and the First Prize of the Best Conference Paper Award of Automation 2025. (hsuyl@mail.nsysu.edu.tw)

Yu-Ming Lo received his B.S. degree in mechanical engineering from National Central University, Taoyuan, Taiwan, in 2022, and his M.S. degree in mechanical and electro-mechanical engineering from National Sun Yat-sen University, Kaohsiung, Taiwan, in 2024. He is currently an installation engineer in KLA-Tencor Corporation. His research interests include artificial intelligence, image processing, and sports sciences. (loym0426@gmail.com)