Pattern Recognition 128 (2022) 108674

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

HandyPose: Multi-level framework for hand pose estimation

Divyansh Gupta, Bruno Artacho, Andreas Savakis*

Department of Computer Engineering, Rochester Institute of Technology, Rochester, New York 14623, USA

ARTICLE INFO

Article history:
Received 13 July 2021
Revised 7 February 2022
Accepted 27 March 2022
Available online 1 April 2022

Keywords:
Hand pose estimation
Feature representations
Computer vision

ABSTRACT

Hand pose estimation is a challenging task due to the large number of degrees of freedom and the frequent occlusions of joints. To address these challenges, we propose HandyPose, a single-pass, end-to-end trainable architecture for 2D hand pose estimation using a single RGB image as input. Adopting an encoder-decoder framework with multi-level features, along with a novel multi-level waterfall atrous spatial pooling module for multi-scale representations, our method achieves high accuracy in hand pose while maintaining manageable size complexity and modularity of the network. HandyPose takes a multi-scale approach to representing context by incorporating spatial information at various levels of the network to mitigate the loss of resolution due to pooling. Our advanced multi-level waterfall module leverages the efficiency of progressive cascade filtering while maintaining larger fields-of-view through the concatenation of multi-level features from different levels of the network in the waterfall module. The decoder incorporates both the waterfall and multi-scale features for the generation of accurate joint heatmaps in a single stage. Our results demonstrate state-of-the-art performance on popular datasets and show that HandyPose is a robust and efficient architecture for 2D hand pose estimation.

1. Introduction

Hand pose estimation is an important computer vision task that includes methods for 2D hand pose [1–3] and 3D hand pose estimation [4]. Hand pose is related to human body pose estimation [5–8], but it is more challenging due to the large number of degrees of freedom in the human hand and the high degree of occlusion of joints in a monocular view of the hand. To overcome the issue of localizing joints that are occluded, methods may employ probabilistic and graphical models [9] or use anchor poses to estimate the occluded joints [10]. Methods for 2D hand pose include the Spatial Information Aware Graph Neural Network [11] which uses a pool of graphical models and belief propagation to better extract relations between each joint and its neighbors. Various standard backbones are considered, but they don't capture multi-scale information.

Leveraging on recent advances of multi-scale feature representations with application to human pose estimation in [7] and [12], we propose HandyPose, a single-stage network for hand pose estimation that is end to end trainable and produces state-of-the-art results on the CMU Panoptic [13] and MPII+NZSL [14] datasets. To deal with the challenges of hand pose context and resolution, our architecture generates improved Waterfall Atrous Spatial Pooling (WASP) [15] representations by combining features from multiple levels of the backbone network via our Multi-Level WASP (MLW) module. Examples of hand pose estimation obtained with HandyPose are shown in Fig. 1. A main component of our HandyPose architecture is the integration of Multi-Level Features (MLF) along with the extended Field-of-View (FOV) extracted by the advanced Waterfall Atrous Spatial Pooling (WASPv2) module [12]. The HandyPose encoder-decoder architecture with our MLW module combines the cascaded approach for atrous convolutions with larger FOV with feature extraction from multiple levels of the backbone and has the potential for wider range of applications to other tasks such as object pose estimation [16] and segmentation.

HandyPose predicts the location of hand joints by utilizing contextual representations obtained with the multi-scale and multi-level scheme taken in our network. Our contextual representation approach allows better detection of shapes, resulting in a more accurate estimation of occluded joints, without requiring postprocessing relying on statistical or geometric methods. The main contributions of this paper are the following.

- We propose HandyPose, a multi-level and multi-scale, end-to-end trainable, single-stage framework for 2D hand pose estimation.
- We introduce the Multi-level Waterfall Atrous Spatial Pooling module that effectively encodes feature maps with large FOV and contextual information.

* Corresponding author.
E-mail address: Andreas.Savakis@rit.edu (A. Savakis).

https://doi.org/10.1016/j.patcog.2022.108674
0031-3203/© 2022 Elsevier Ltd. All rights reserved.

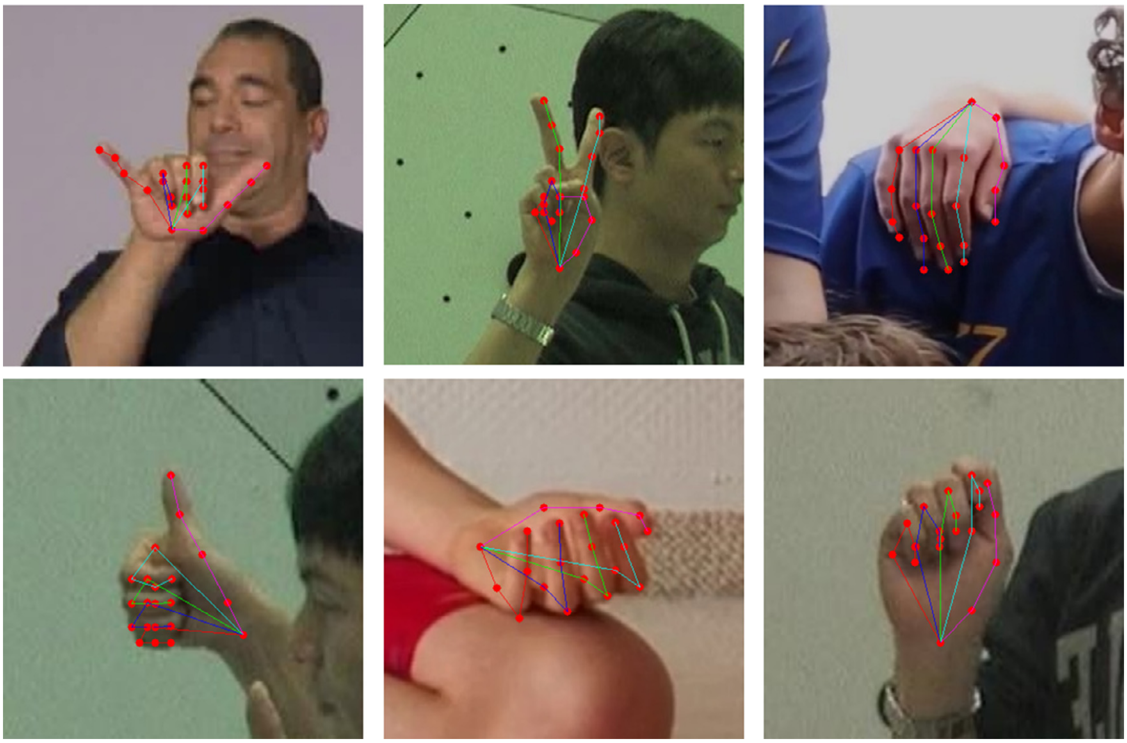

Fig. 1. Pose estimation examples with our HandyPose method.

- HandyPose is a modular encoder-decoder architecture that incorporates multi-level features in both the encoder and the decoder modules, making it easy to modify and expand.
- HandyPose achieves state-of-the-art result for 2D hand pose on two popular benchmarks.

2. Related work

2.1. Early works

Before the meteoric growth of deep learning methods, hand pose estimation was a laborious task that was rarely deployed in applications due to the setup expense and algorithm complexity. Most traditional computer vision algorithms were applied for estimating the hand pose. Early methods include k-nearest neighbors and decision trees [17]. For instance, [18] applied a multi-colored glove for reconstructing the pose of a hand from a single image, and relied on the nearest neighbors technique to map the hand detection using the multiple colors of the glove. A downside of these techniques is the high cost of implementation, complex setup, and requirement of multiple cameras. Aiming to reduce the setup complexity, [19] combines the use of a single camera setup with Bayesian technique to better connect and predict the hand pose estimation. Despite the reduced complexity, this method still relies on the color pattern of gloves to infer the pose of the hand. The Microsoft Kinect sensor [20] introduced a less complex setup with the use of a single camera and the addition of a depth sensor to extract the 3D hand pose estimation.

2.2. Deep learning methods

More recently, deep learning methods gained in popularity, as they achieved more accurate hand pose estimation [1,3,21,22]. Methods such as [23] combine depth estimation to extract the full 3D coordinates in addition to the 2D detections. Recurrent Neural Networks (RNNs) were also used to extract the spatial information of the joints and palm [24].

The similarity of the hand pose estimation task to human pose estimation, allows the adoption of methods developed for the overall human body. The Convolutional Pose Machine (CPM) approach [6] is popular for human pose estimation due to its easy implementation and the modularity of joint detections via the refinement of feature maps through multiple stages of the network. CPM was later expanded to integrate the concept of Part Affinity Fields (PAF) resulting in the widely used OpenPose method [25]. Leveraging on the innovations of CPM, the approach in [26] developed a multi-view bootstrapping method that implements a CPM-based architecture for 2D hand pose estimation, relying on a detector for generating a large dataset of hand keypoints. Similarly, the Hourglass (HG) network [5] stacks up to 8 iterations of its network to refine feature maps.

Pose estimation methods may be categorized as top-down [12,27] or bottom-up [25,28]. Top-down approaches rely on an object detection stage to locate instances of the person or pose by using detectors such as YOLO [29] or Faster R-CNN [30]. The detection stage is followed by the detection of keypoints to estimate pose for every instance. The High-Resolution Network (HRNet) [31] combines multi resolutions throughout the network, while also maintaining high-resolution feature maps through all layers of the network. Despite achieving high accuracy for individual poses, top-down methods are dependent on the performance of object detectors.

Bottom-up approaches initially detect all keypoints in the image, followed by keypoint clustering for pose estimation of separate instances. Top-Down methods can be expanded to Bottom-Up approaches by incorporating offset regression into their decoder, for instance the HigherHRNet [28] leverages the promising results of the HRNet method to achieve high accuracy bottom-up pose estimation. Both top-down [26] and bottom-up [2], approaches can be utilized for hand pose estimation.

Santavas et al. [2] proposed the AttentionNet for hand pose by combining a self-attention module [32] with a feed-forward CNN for directly estimating the hand keypoints without intermediate supervision. AttentionNet adopts a regression based approach in-

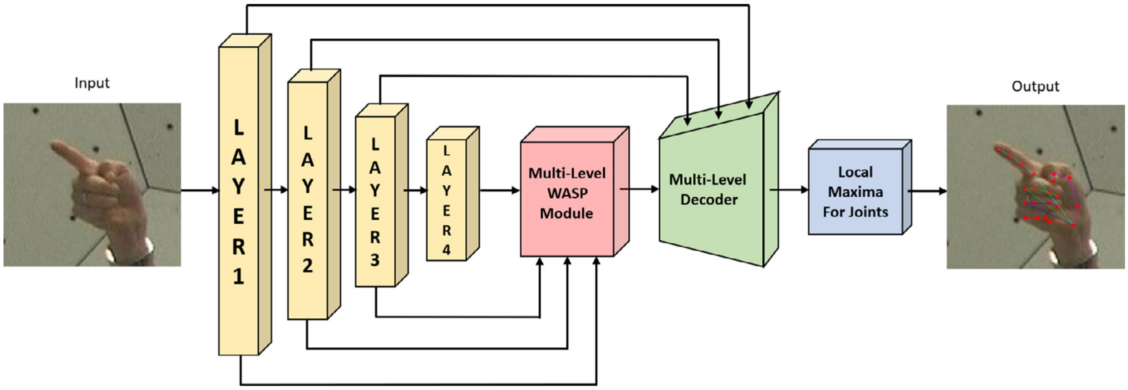

Fig. 2. The proposed HandyPose architecture for 2D hand pose estimation. The input RGB image is fed into the ResNet-101 backbone, obtaining 400 feature maps after the concatenation of Multi-Level WASP outputs and MLF feature channnel. The Multi-Level Decoder module generates heatmaps (one per joint) and location of keypoints are extracted from the heatmaps by applying a local maxima function.

stead of pixel-wise classification, which results in high processing speed but is limited in generalization performance. Aiming to achieve a better structural learning, a non-parametric structure regularization approach [33] utilizes the synthetic hand mask for learning the structure of keypoints. Mask-Pose [1] uses the silhouette information and builds a two-stage cascade network that includes a mask prediction stage and a pose prediction stage. The SRHandNet [3] approach regresses hand regions of interest (ROI) simultaneously with hand keypoints to improve the performance of hand pose estimation.

2.3. Graphical methods

Many methods apply techniques to assess the geometric constraints of hand joints [34]. The Adaptive Graphical Model Network (AGMN) [21] aims to learn adaptive parameters for the graphical model for each input image and refine its pose estimation. In another approach, SIA-GCN [11] uses a modified graph neural network for hand pose estimation, representing features at each node by a 2D spatial confidence map instead of 1D vectors, with the goal of preserving the spatial information provided in the 2D feature maps.

Using graphical models along with a CNN, R-MGMN [22] associated the spatial relationships with input hand shape in order to reduce the spatial irregularity of hand keypoints. More recently, the Hand-Object Pose Estimation Network (HOPE-Net) [9] applies graph convolutions in a modified U-Net configuration [35] to improve the hand pose estimation in conjunction with an object that is picked by the hand.

2.4. Multi-Scale feature representations

A challenge faced by networks using CNNs for pose estimation is the significant reduction of the resolution caused by the repeated use of pooling layers. For semantic segmentation, Deconvolution Networks [36] employed deconvolution layers to address the problem, by upsampling the resolution of each layer in the decoder stage of the network.

Other methods rely on the use of atrous convolutions to avoid downsampling and increase the size of the receptive fields in the network. Atrous convolutions are applied systematically by using a multi-scale context aggregation module [37] in order to better preserve the contextual information of the input image.

Deeplab [38] further explored the advantages of atrous convolution by proposing the Atrous Spatial Pyramid Pooling (ASPP) module for semantic segmentation. The ASPP approach increases the FOV at larger dilation rates in parallel branches without downsampling. The main challenge this network faces is the increased computational cost and memory requirements due to its increased resolution. DeepLabv3 [39] addressed this issue by applying a cascade of atrous convolutions in a sequential order with the help of progressive filtering to maintain the FOV at different layers of the network.

The WASP module [15] was initially proposed for semantic segmentation and was used in UniPose [7] for human pose estimation. The WASP module operates in a waterfall-like flow, progressively extracting the larger FOV from a series of atrous convolutions at different dilation rates. The waterfall architecture effectively generates multi-scale features from the backbone without immediately parallelizing the input stream, as it maintains the advantages of the ASPP module with lower computational and memory requirements.

The waterfall approach was recently enhanced by the introduction of the WASPv2 module for multi-person pose estimation in OmniPose [12]. The WASPv2 architecture extracts feature maps at multiple scales, while preserving the original resolution by avoiding downsampling. The cascade approach used in WASPv2 maintains the high resolution of the feature maps by arranging atrous convolutions in a cascaded structure with increasing dilation rates of $(1, 6, 12, 18)$. This arrangement increases the FOV in the feature representations without affecting the input resolution. Furthermore, WASPv2 integrates the decoder with the feature extraction process, effectively reducing the overall computational cost. This effective multi-scale architecture of WASPv2 achieves high accuracy for human pose estimation. In this paper, we adopt the waterfall architecture for hand pose estimation and extend it with multi-level features.

3. HandyPose architecture

We propose *HandyPose*, a multi-level framework for hand pose estimation, that achieves high performance by the use of a novel multi-level WASP module. Taking into consideration the issue of frequent occlusion in the joints of the hand, we designed HandyPose to combine feature maps from different levels of the backbone with the multi-scale approach of the WASPv2 module to obtain a more powerful representation.

The HandyPose architecture is shown in Fig. 2. Our feature representation framework uses features from all successive blocks (levels) of the ResNet-101 backbone and incorporates them at various places in the network. Our enhanced MLW module increases the number of feature maps, forming a more robust representation, and maintains the high resolution of the maps. These representations, along with a multi-level decoder, generate more accurate predictions for both occluded and visible joints.

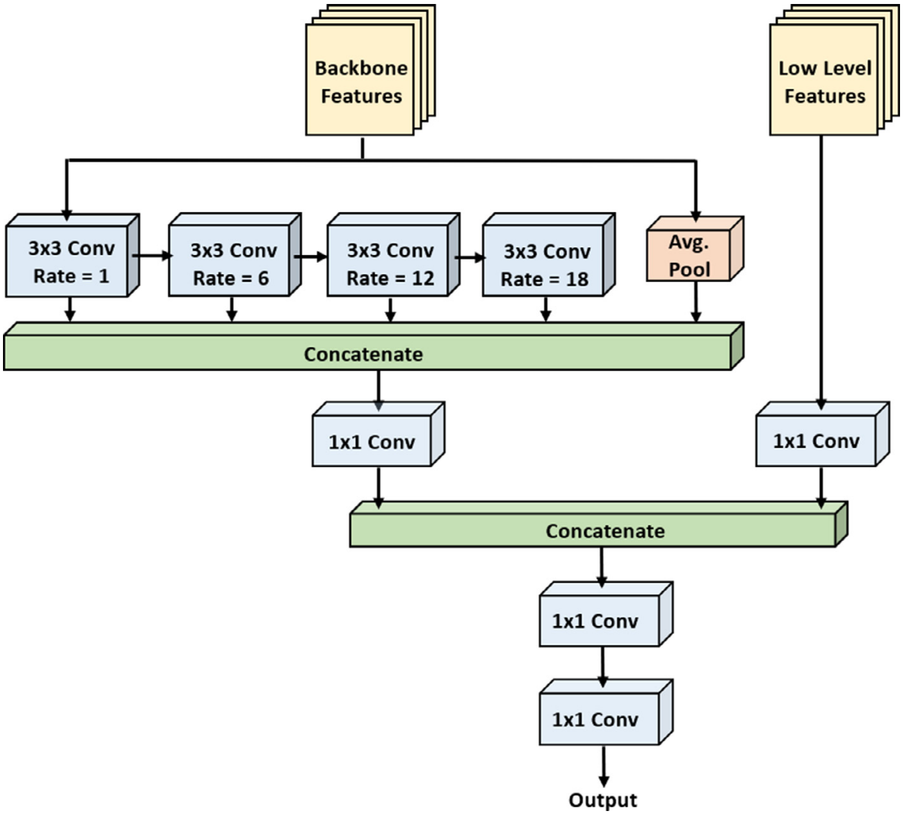

Fig. 3. The WASPv2 waterfall module with integrated decoder. The inputs are 2048 channels of backbone features and 256 channels from the lowest and highest level of the backbone. The number of output channels is equal to the number of joints.

In order to deal with the loss of contextual and spatial information by successive pooling, our feature utilization approach fuses multi-level features extracted from all blocks of the ResNet backbone. The feature maps from Layers 1, 2, and 3 of the ResNet backbone are fed-forward in the network with the use of a 1 × 1 convolution and bilinear interpolation on multi-level features to generate feature maps of matching dimensions. This helps reduce the size of the network but preserves the image information.

The MLF component of HandyPose extracts the residual feature maps at different resolutions from the first three blocks of the ResNet backbone and combines them with the high-level features of the backbone after passing through a waterfall of atrous convolutions to fuse the spatial information at different scales. We experimented with different fusing combinations and the best results were obtained by combining the multi-level feature maps in increasing order of Layer 1, 2 and 3 and concatenating them at different levels of the MLW module containing the high-level features.

Before concatenation, we perform a 1 × 1 convolution preceding the bilinear interpolation on the multi-level feature maps to match the resolution and shape. The unit convolutions reduce the depth channels to the desired amount of feature channels of 48 at each level. Experiments with different numbers of high and low level features of the network were performed, resulting in higher performance when using 48 channels of feature maps from each of the first three blocks of the ResNet backbone and 256 feature maps from the last block output of the ResNet backbone.

3.1. Multi-level WASP module

We present the advanced MLW module in Fig. 4, incorporating the multi-scale extraction of the WASPv2 module shown in Fig. 3 with the multi-level backbone features.

HandyPose extracts four residual blocks in the ResNet backbone generating feature maps with 256, 512, 1024, and 2048 channels at all levels of the ResNet. We organized atrous convolutions in a waterfall like architecture receiving inputs from different levels of the ResNet backbone and concatenating them with the output of progressive filtering of the successive layer of atrous convolutions in a waterfall fashion, as shown in Fig. 4.

The operations in the MLW module can be described by the following equations:

$$F_i = \begin{cases} K_{d_1} \circledast f_4 & \text{if } i\text{=1} \\ K_{d_i} \circledast (f_{i-1} \circledast K_1 + F_{i-1}) & \text{otherwise} \end{cases} \tag{1}$$

$$F_{Waterfall} = K_1 \circledast \left(\sum_{i=1}^{4} (F_i) + AP(f_4) \right) \tag{2}$$

where $\circledast$ represents convolution, K_1 and K_{d_i} represent convolutions of kernel size 1×1 and 3×3 with dilations of $d_i = (1, 6, 12, 18)$, f_i represents the output of block i from the ResNet backbone, the symbols + and $\sum$ denote the concatenation operation, and AP denotes the Average Pool operation, as shown in Fig. 4.

Our module achieves a multi-level and multi-scale representation by fusing the cascade of atrous convolutions and MLF features and concatenating the outputs with the average pooling of the high level features from the last block of ResNet-101. The resulting feature maps are reduced in channel depth by applying a 1 × 1 convolution on them. These feature maps along with the MLF are used as inputs for further processing and generating the final heatmaps in the decoder, which receives features from all levels of the backbone.

The WASPv2 and MLW modules are illustrated in Figs. 3 and 4 respectively. The WASPv2 module uses only the highest and lowest level features as input followed by a cascade of atrous convolutions. In contrast, The proposed MLW module adopts a multi-level

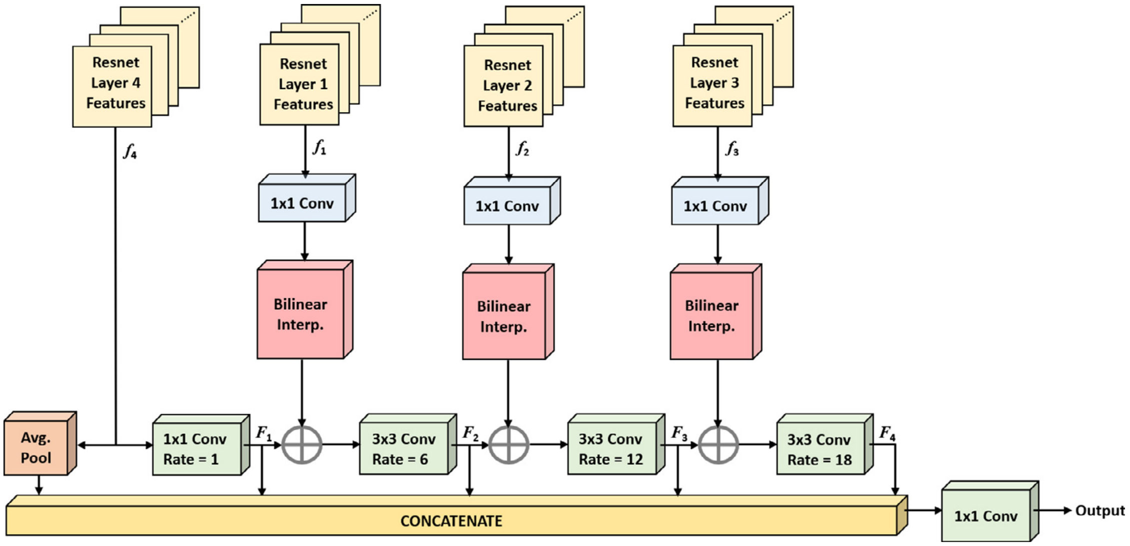

Fig. 4. The proposed Multi-Level WASP module, a multi-level and multi-scale architecture with larger FOV for preserving the contextual information with the introduction of multi-level features along the cascade of atrous convolutions. The ⊕ refers to concatenation. The input is 2048 feature channels from the lowest level of the backbone and the output generates 256 feature channels that are fed into the decoder.

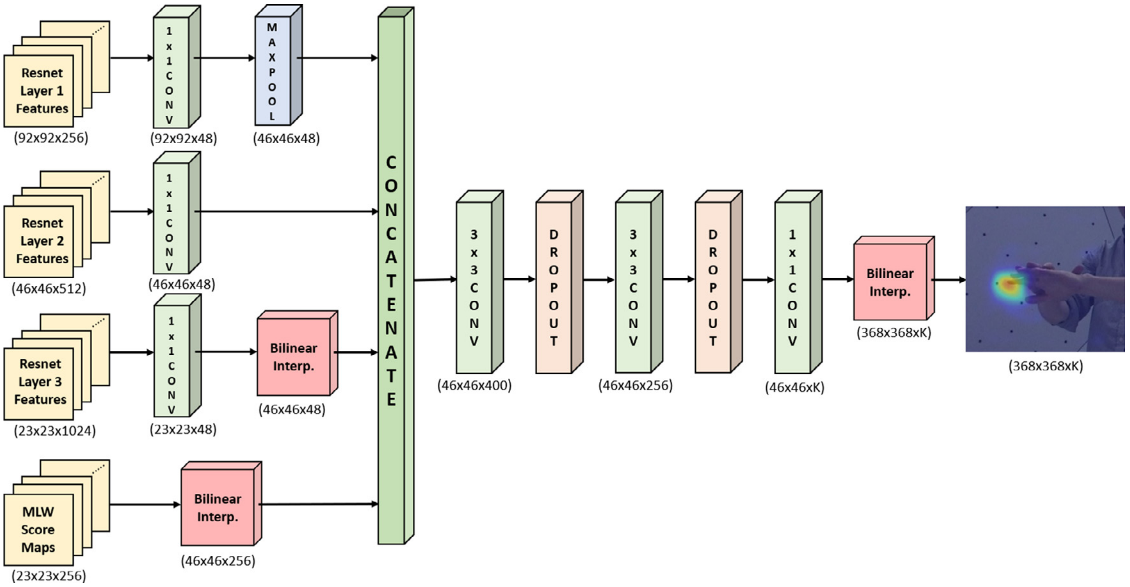

Fig. 5. The proposed Multi-Level Decoder (MLD) module receives $(256, 512, 1024)$ feature maps as input from the ResNet backbone along with the output of the MLW module. Further processing generates K output heatmaps corresponding to the hand joints. The output image illustrates a single joint output produced by HandyPose.

approach extracting feature maps from different levels of the network and concatenating them at different stages of the cascade of atrous convolutions. Furthermore, the WASPv2 module contains an inbuild decoder to output the final feature maps, whereas the MLW module feature maps are fed in a separate multi-level decoder module to further improve the performance.

3.2. Multi-level decoder

The feature maps generated by different stages of our HandyPose architecture are fused in the Multi-Level Decoder (MLD), generating the final K heatmaps corresponding to each hand point joint from the dataset. Fig. 5 shows the MLD module where multi-level features are processed to generate 48 feature maps each for the first three layers of the ResNet backbone and combined with 256 score maps generated by the MLW module to form a total of 400 feature maps. The resultant feature maps are then processed through convolutional layers and finally interpolated to generate the output heatmaps of the same size as the input images. The concatenation of multi-level features followed by the convolution and dropout layers in our MLD improves the prediction accuracy of the network by 0.8%. The multi-level features output, after concatenation with the MLW module, is described as follows:

$$F_{Concat} = MP(f_1 \circledast K_1) + \sum_{i=2}^{3} (f_i \circledast K_1) + F_{Waterfall} \tag{3}$$

$$F_{out} = ((F_{Concat} \circledast K_3) \circledast K_3) \circledast K_1 \tag{4}$$

where $\circledast$ represents convolution, K_1 and K_3 represent convolutions of kernel size 1×1 and 3×3, f_i represents the output of block i from the ResNet backbone, + and $\sum$ denotes the concatenation operation, MP denotes the Max Pool operation, $F_{Waterfall}$ is the output from the MLW module, and F_{out} is the output of HandyPose, as shown in Fig. 5.

Our decoder process the output to generate heatmaps corresponding to the keypoints, extracting the joint locations through local maximum operation on each heatmap, that is, the location in which there is the highest confidence that a joint is located. HandyPose does not require any post-processing operations as Non-Maximum Suppression (NMS) for joint localization purposes. We performed experiments with different depth channels for our multi-level feature maps and their effects on the performance.

4. Experiments

HandyPose experiments were based on metrics set by each dataset and processed at the same resolution to the dataset and other networks to allow comparison of performance.

4.1. Datasets

We perform 2D hand pose experiments on two hand pose datasets, the CMU Panoptic Hand Dataset [13] and the MPII + NZSL Dataset [14]. Following procedures adopted by [11,21], and [22], initial cropping of a square image patch for the annotated hands was performed in the original images, resulting in a square bounding box with dimensions 2.2× the size of the hand.

The CMU Panoptic Hand Dataset consists of 14,817 images taken in a panoptic studio. Each image has 21 joint annotations of a single hand and is one of the benchmark datasets for 2D hand pose estimation. Following procedures of other methods in the comparison section, we divided the dataset by splitting the samples into training, validation, and testing sets containing 70%, 15%, and 15% of the images, respectively. The main challenge of the dataset is the high occurrence of self-occlusion for hand keypoints.

The MPII + NZSL dataset contains images from the MPII human pose dataset combined with the New Zealand Sign Language dataset. It consists of 2758 images of people in everyday activities, and contains 21 labeled joint keypoints for the hand. Following the same procedure as the previous dataset, we crop the region with the target hand.

4.2. Evaluation metric

For the evaluation of HandyPose, we apply the metric of Probability of Correct Keypoint (PCK). This metric measures a correct prediction when the joint detection is within a certain distance threshold σ in relation to the hand bounding box, compared to the groundtruth label. The main threshold adopted for the dataset evaluations is $\sigma = 0.02$. Therefore, the PCK metric is defined as:

$$PCK(@.02) = P(\sigma)/K \tag{5}$$

$$mPCK = \frac{1}{N}\left(\sum_{\sigma=1}^{6} PCK@[\frac{\sigma}{100}]\right) \tag{6}$$

For a threshold σ of 0.02 and input image of size w × w, PCK is defined as the number of predicted keypoints (P) that are within the threshold range $\sigma \times 0.02$ of the ground truth keypoints location divided by the total number of keypoints (k). The normalized threshold σ used is with respect to the size of the hand bounding box. We evaluated our model for different threshold values ranging from {0.01 - 0.06} with a constant increment of 0.01. We also reported the mean PCK (mPCK) showing the average performance of our network compared to current state-of-the-art methods by substituting the value of N = 6.

4.3. Implementation details

Since the hand occupies a small area of the images in the dataset, and following procedures adopted by [11,21], and [22], we pre-process the images, cropping it to a square with 2.2× the hand size. The cropped images are then resized to a constant resolution of 368 × 368, scaled between [0,1], and and normalized following procedures comparable to other methods. Resized images reduce the computational footprint of the network, and decreased the training time of the network. We used a batch size of 32 images during training, and performed training for 80 epochs, applying an initial learning rate of $lr = 10^{-4}$, being reduced by a factor of 0.1 after 60 epochs.

5. Results and analysis

We present HandyPose results on two prominent datasets and provide comparisons with current state-of-the-art methods. We also performed experiments by changing our MLW module with ASPP, WASP, and WASPv2 modules, the results obtained are discussed in the Ablations section.

5.1. Ablations studies

We initially performed a series of experiments to analyse the accuracy, as well as computational cost and number of parameters, for each component added to our HandyPose framework. We performed a series of ablation studies to investigate the performance of atrous convolutions before developing our MLW module. Performing experiments with the ASPP, WASP and WASPv2 module on the HandyPose architecture, we observe the improvement in performance by using a cascade of atrous convolutions. WASPv2 with multi-level features is performing the best, compared to the other configurations, as show in Table 1.

We also compared the use of different feature extractors and decoders for our HandyPose framework. Table 1 demonstrates the results for the inclusion of the ASPP module [38], WASP module [7], and WASPv2 module [12] in combination with the improved feature extractor in our architecture. The combination of the modified WASPv2 module with our implementation of MLF and our MLD demonstrated to be the more efficient architecture, gaining 3.46% in accuracy (from 71.15% to 74.61%) when compared to DeepLab.

Table 2 demonstrates the performance comparison of three different backbones, ResNet-50, ResNet-101 and HRNet-W48 [31], combined with different components of HandyPose on the CMU Panoptic Hand dataset [13]. This dataset consists of images of human of resolution 1920 × 1080, but hands are cropped from a small part of the image due to their smaller size. The average size of hand crops in the dataset is 44 × 48. From ablations performed in Table 2 we can infer that the multi-level feature resolutions of the ResNet backbone help to further extract important information from different scales. In comparison, the high-resolution HRNet backbone is not as effective for processing smaller portions of

Table 1
Ablation studies for different configurations of HandyPose with ResNet-101 backbone for the CMU Panoptic Hand dataset [13]. MLW and MLD represents the Multi-Level WASP and Multi-Level Decoder modules using Multi-Level Features (MLF). ASPP, WASP, and WASPv2 indicates the use of various atrous modules in the network.

Method	Params (M)	GFLOPs	ASPP	WASP	WASPv2	MLW	MLD	PCK @0.2
ResNet [40]	44.6	28.3						69.20%
Deeplab [38]	59.3	34.9	✓					71.15%
Unipose [7]	47.5	29.2		✓				70.32%
WASPv2	47.0	28.8			✓			73.58%
WASPv2 + MLF	47.2	29.3				✓		73.97%
HandyPose	47.5	29.5				✓	✓	**74.61%**

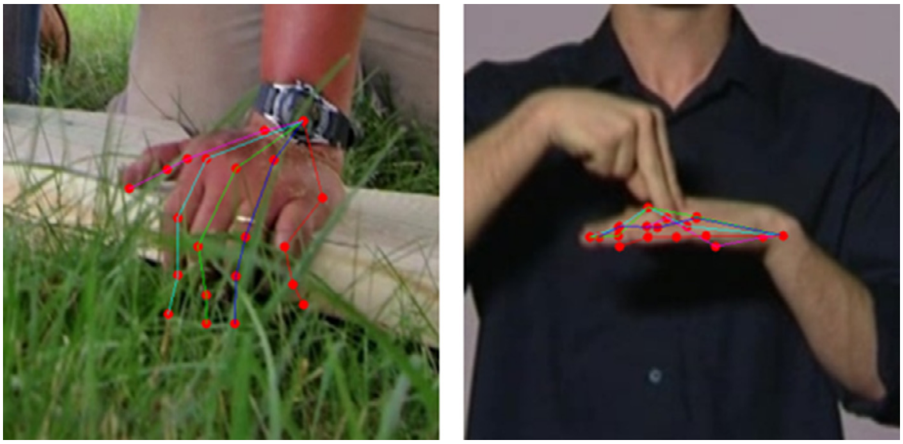

Fig. 6. Examples of fail cases for images in the datasets: (a) High occlusion by grass and bat causes the joint positions to be misplaced; and (b) Self occlusion of joints are a common issue in hand pose estimation.

the image containing the hand. The ResNet-101 model improves the accuracy by 5.41% (from 69.20% to 74.61%), while the HRNet configuration improves by only 1.72% (from 69.55% to 71.27%).

Table 3 presents a comparison for the implementation of our multi-level feature maps by applying different numbers of feature maps from all blocks of the ResNet backbone into the modified MLW module and the MLD of HandyPose. We tested our network with different numbers of feature maps for the first three blocks of ResNet $\{24, 48, 96, 128\}$, generating the multi-level approach. Similar to results previously observed by architectures applying low-level features to the decoder stage [7,12,38], the use of 48 feature maps for lower level features and 256 maps for high level features was found to be the more efficient combination.

Examples of failure cases are shown in Fig. 6. Most common issues of failure can be seen due to object/self occlusion. Self occlusion results in joints to be positioned very close to each other, confusing the network when detecting joints in very close proximity. Object occlusion results in obstruction of some part or most of the hand, as hands are readily used for holding objects. Fortunately, there are few failed cases and in most cases our model is able to successfully detect occluded joints.

Table 2
Performance comparison of three different backbones, ResNet-50, ResNet-101 and HRNet-W48 in the presence or absence of different components of the HandyPose architecture for the CMU Panoptic Hand dataset [13].

Backbone	Params (M)	GFLOPs	WASPv2	MLW	MLD	PCK @0.2
ResNet-50	25.6	19.0				65.44%
ResNet-50	27.9	19.5	✓			69.28%
ResNet-50	28.2	20.1		✓		70.13%
ResNet-50	28.5	20.2		✓	✓	70.92%
ResNet-101	44.6	28.3				69.20%
ResNet-101	47.0	28.8	✓			73.58%
ResNet-101	47.2	29.3		✓		73.97%
ResNet-101	47.5	29.5		✓	✓	**74.61%**
HRNet-W48	68.0	38.1				69.55%
HRNet-W48	68.2	38.9	✓			70.30%
HRNet-W48	68.2	39.3		✓		70.91%
HRNet-W48	68.3	39.6		✓	✓	71.27%

Table 3
HandyPose results for the CMU Panoptic Hand dataset [13] showing the effects of varying the number of feature maps in the multi-level-features.

Feature Maps	PCK@0.02
24	72.8%
48	**74.61%**
96	71.5%
128	70.3%

5.2. Experimental results on CMU Panoptic Hand dataset

We compared our multi-level approach to current state-of-the-art methods as shown in Table 4. The SiaPose method [11] considers several backbones in its configuration, including the heavy-weight HG backbone. In addition, SiaPose also adds up to 40% in its size by combining the backbone with the 10 heads for the refinement of predictions through graphical models. HandyPose achieved an overall best performance, with significant gains in comparison to the previous state-of-the-art while using a smaller backbone, ResNet-101. For the overall average accuracy, HandyPose achieves a mPCK of 81.75%, increasing the previous state-of-the-art. Most of the improvement of HandyPose is due to its higher capacity to precisely detect keypoints at lower thresholds, increasing the PCK@0.01 by 9.3% compared to the previous state-of-the-art (from 39.46% to 43.13%). HandyPose is an overall more accurate framework that achieves most of its gains in the fine refinement of joints detections for tight thresholds.

In contrast to other methods relying in multi-stage frameworks [11,21], and [22], HandyPose is able to detect with higher accuracy hand joints in a single iteration network. HandyPose improves the accuracy by 6.1% to its nearest competitor for the most traditional PCK with threshold of 0.02, and an even larger 17.8% for more precise hand pose estimation in a less forgiving threshold of 0.01, attesting to the more precise alignment of HandyPose to the exact joint locations.

Examples of HandyPose detections for the CMU Panoptic Hand dataset [13] are shown in Fig. 7. It is noticeable that HandyPose addresses with higher accuracy occluded joints, the most challenging component of hand pose estimation in general and for this dataset.

5.3. Experimental results on MPII+NZSL dataset

We next performed our experiments on the MPII+NZSL [14] Our HandyPose framework outperformed the current SOTA methods by a significant margin as reported in Table 5.

Similar to results from the previous dataset, HandyPose outperforms the state-of-the-art, achieving an overall mPCK of 56.39%, increasing the accuracy from other methods by 4.8%. Significant improvements are also present for the traditional PCK with threshold of 0.02 by a margin of 7.2%, reaching 41.66%. The MPII+NZSL dataset presents more challenging images in the wild, having in addition to the high incidence of occlusion a great amount of vari-

Table 4
Results for 2D hand pose estimation and comparison with other state-of-the-art-methods for the CMU Panoptic Hand Dataset [13].

CMU Panoptic Hand Dataset									
Method	Params (M)	GFLOPs	PCK @0.01	PCK @0.02	PCK @0.03	PCK @0.04	PCK @0.05	PCK @0.06	mPCK
HandyPose (ours)	47.5	29.5	**43.13%**	74.61%	87.85%	92.81%	**95.28%**	**96.84%**	**81.75%**
10-head R-SiaPose-HG [11]	-	-	39.46%	**77.22%**	**88.45%**	**92.97%**	94.85%	96.09%	81.48%
UniPose [7]	47.5	29.2	36.60%	70.32%	84.81%	90.60%	93.72%	95.64%	78.61%
10-head R-SiaPose-CPM [11]	-	-	26.62%	65.80%	81.60%	88.02%	91.39%	93.36%	74.47%
R-SiaPose-CMU [11]	-	-	24.94%	62.08%	77.83%	84.91%	88.78%	91.34%	71.64%
AGMN [21]	-	-	23.90%	60.26%	76.21%	83.70%	87.72%	90.27%	70.34%
R-MGMN [22]	-	-	23.67%	60.12%	76.28%	83.14%	86.91%	89.47%	69.93%
AGMN Sep. Trained [21]	-	-	21.52%	56.73%	73.75%	82.06%	86.39%	89.10%	68.25%
CPM [6]	31.4	163.7	22.88%	58.10%	73.48%	80.45%	84.27%	86.88%	67.67%

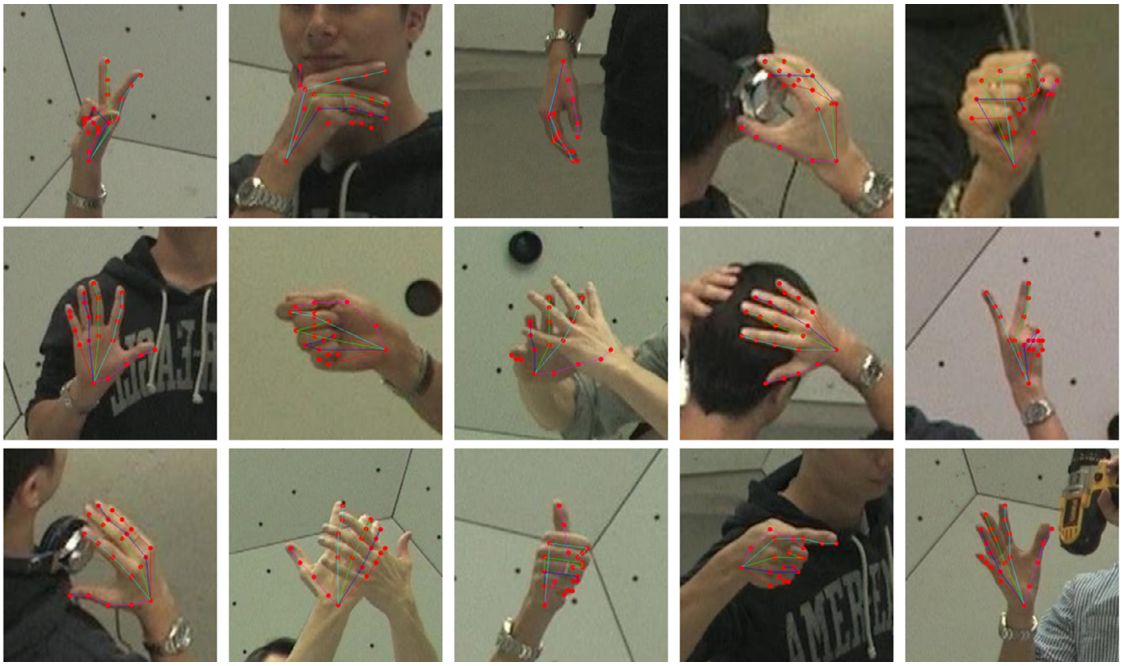

Fig. 7. Pose estimation examples from the CMU Panoptic Hand Dataset [13].

Table 5
Results for 2D hand pose estimation and comparison with other state-of-the-art-methods for the MPII + NZSL Dataset [14].

MPII + NZSL Dataset									
Method	Params (M)	GFLOPs	PCK @0.01	PCK @0.02	PCK @0.03	PCK @0.04	PCK @0.05	PCK @0.06	mPCK
HandyPose (ours)	47.5	29.5	**16.02%**	**41.66%**	**58.15%**	**68.12%**	**74.53%**	**79.90%**	**56.39%**
UniPose [7]	47.5	29.2	14.29%	38.85%	55.28%	65.14%	71.75%	77.52%	53.80%
10-head R-SiaPose-HG [11]	-	-	12.19%	33.34%	49.13%	59.86%	67.83%	73.69%	49.33%
10-head R-SiaPose-CPM [11]	-	-	8.40%	24.71%	39.33%	50.31%	59.04%	66.01%	41.30%
CPM [6]	31.4	163.7	8.05%	23.78%	37.74%	48.00%	55.65%	61.68%	39.15%

ability of images, resulting in a more difficult dataset for architectures to predict hand pose estimation.

Examples for hand pose estimation for images from the MPII part of the dataset and the New Zealand Sign Language part of the dataset are shown in Fig. 8 and Fig. 9, respectively. Images from the MPII part of the dataset present a higher challenge due to greater variation of the background in the wild, adding to the challenge of occlusion present in both parts of the dataset.

6. Conclusion

Hand pose estimation has drawn increasing attention during the past decade due to its similarity to full body pose estimation and usefulness in a wide range of applications including augmented reality, virtual reality, human-computer interaction, and action recognition. The high degrees of freedom in the human hand movements and frequent self-occlusion of hand joints make the task more challenging. In addition, the low resolution of the hand crops make multi-scale feature representations more challenging.

We presented the HandyPose framework for 2D hand pose estimation, consisting of a modular, end-to-end trainable network. We proposed a multi-level waterfall module and multi-level decoder to better leverage multi-level and multi-scale features and more accurately predict pose estimation without losing spatial and contextual information in the presence of occlusions of hand keypoints.

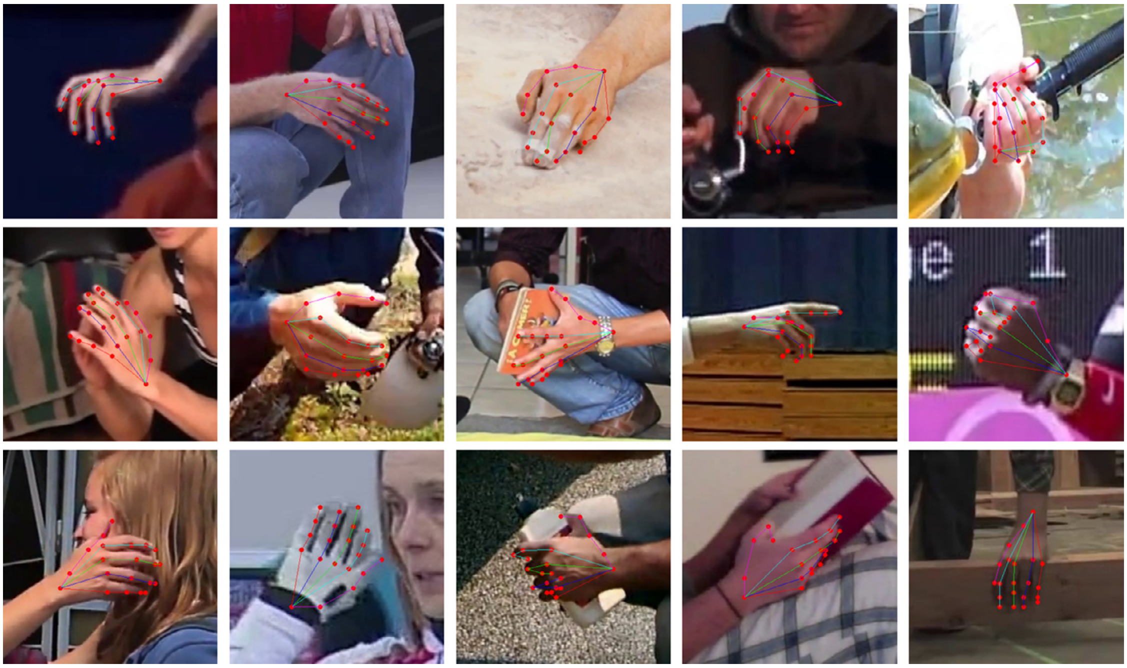

Fig. 8. Pose estimation examples from the MPII+ NSZL dataset [14].

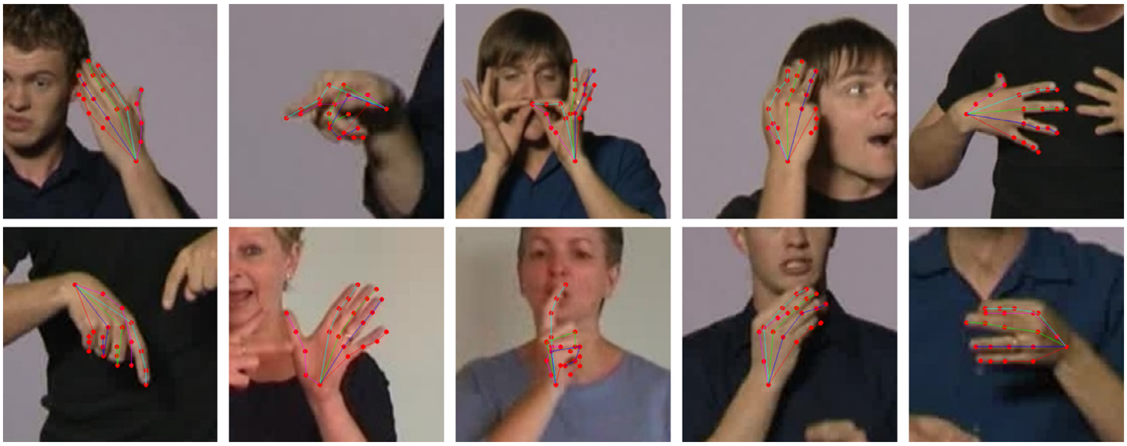

Fig. 9. Pose estimation examples from the New Zealand Sign Language (NZSL) dataset.

Our multi-level feature extraction approach deals more effectively with the spatial loss of resolution due to the small size of the input image and successive pooling, while achieving high accuracy and maintaining the size complexity and modularity of the network. HandyPose achieves state-of-the-art results on two hand pose datasets and set the foundation for future work on 3D pose estimation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was supported in part by the National Science Foundation grant ♯1749376.

References

[1] Y. Wang, C. Peng, Y. Liu, Mask-pose cascaded CNN for 2D hand pose estimation from single color image, IEEE Trans. Circuits Syst. Video Technol. 29 (11) (2019) 3258–3268, doi:10.1109/TCSVT.2018.2879980.

[2] N. Santavas, I. Kansizoglou, L. Bampis, E. Karakasis, A. Gasteratos, Attention! a lightweight 2D hand pose estimation approach, IEEE Sens J 21 (10) (2021) 11488–11496, doi:10.1109/JSEN.2020.3018172.

[3] Y. Wang, B. Zhang, C. Peng, Srhandnet: real-time 2D hand pose estimation with simultaneous region localization, IEEE Trans. Image Process. 29 (2020) 2977–2986, doi:10.1109/TIP.2019.2955280.

[4] Y. Zhang, S. Mi, J. Wu, X. Geng, Simultaneous 3D hand detection and pose estimation using single depth images, Pattern Recognit Lett 140 (2020) 43–48, doi:10.1016/j.patrec.2020.09.026.

[5] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision - 14th European Conference, ECCV 2016, Proceedings, Springer Verlag, Germany, 2016, pp. 483–499, doi:10.1007/978-3-319-46484-8_29.

[6] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4724–4732, doi:10.1109/CVPR.2016.511.

[7] B. Artacho, A. Savakis, Unipose: Unified human pose estimation in single images and videos, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[8] E. Souza dos Reis, L.A. Seewald, R.S. Antunes, V.F. Rodrigues, R. da Rosa Righi, C.A. da Costa, L.G. da Silveira Jr., B. Eskofier, A. Maier, T. Horz, R. Fahrig,

Monocular multi-person pose estimation: a survey, Pattern Recognit 118 (2021) 108046, doi:10.1016/j.patcog.2021.108046.

[9] B. Doosti, S. Naha, M. Mirbagheri, D.J. Crandall, Hope-net: A graph-based model for hand-object pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[10] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. Zhou Tianyi, J. Yuan, A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image, in: Proceedings of the IEEE Conference on International Conference on Computer Vision (ICCV), 2019.

[11] D. Kong, H. Ma, X. Xie, SIA-GCN: A spatial information aware graph neural network with 2D convolutions for hand pose estimation, in: British Machine Vision Conference BMVC, 2020.

[12] B. Artacho, A. Savakis, Omnipose: a multi-scale framework for multi-person pose estimation, arxiv:2103.10180 (2021).

[13] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T.S. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, Y. Sheikh, Panoptic studio: a massively multiview system for social interaction capture, IEEE Trans Pattern Anal Mach Intell (2017).

[14] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: New benchmark and state of the art analysis, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[15] B. Artacho, A. Savakis, Waterfall atrous spatial pooling architecture for efficient semantic segmentation, Sensors 19 (24) (2019) 5361, doi:10.3390/s19245361.

[16] D. Gupta, B. Artacho, A. Savakis, Vehipose: a multi-scale framework for vehicle pose estimation, in: Applications of Digital Image Processing XLIV, volume 11842, International Society for Optics and Photonics, 2021, p. 118421K.

[17] C. Keskin, F. Kiraç, Y.E. Kara, L. Akarun, Hand pose estimation and hand shape classification using multi-layered randomized decision forests, in: ECCV, 2012, pp. 852–863. https://doi.org/10.1007/978-3-642-33783-3_61

[18] R.Y. Wang, J. Popović, Real-time hand-tracking with a color glove, ACM Trans. Graph. 28 (3) (2009), doi:10.1145/1531326.1531369.

[19] B. Stenger, A. Thayananthan, P.H.S. Torr, R. Cipolla, Model-based hand tracking using a hierarchical bayesian filter, IEEE Trans Pattern Anal Mach Intell 28 (9) (2006) 1372–1384, doi:10.1109/TPAMI.2006.189.

[20] J. Han, L. Shao, D. Xu, J. Shotton, Enhanced computer vision with microsoft kinect sensor: a review, IEEE Trans Cybern 43 (5) (2013) 1318–1334, doi:10.1109/TCYB.2013.2265378.

[21] D. Kong, Y. Chen, H. Ma, X. Yan, X. Xie, Adaptive graphical model network for 2D handpose estimation, in: British Machine Vision Conference BMVC, 2019.

[22] D. Kong, H. Ma, Y. Chen, X. Xie, Rotation-invariant mixed graphical model network for 2D hand pose estimation, in: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1535–1544, doi:10.1109/WACV45572.2020.9093638.

[23] J. Malik, I. Abdelaziz, A. Elhayek, S. Shimada, S.A. Ali, V. Golyanik, C. Theobalt, D. Stricker, Handvoxnet: deep voxel-based network for 3D hand shape and pose estimation from a single depth map, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020).

[24] C.-H. Yoo, S. Ji, Y.-G. Shin, S.-W. Kim, S.-J. Ko, Fast and accurate 3D hand pose estimation via recurrent neural network for capturing hand articulations, IEEE Access 8 (2020) 114010–114019, doi:10.1109/ACCESS.2020.3001637.

[25] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2D pose estimation using part affinity fields, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1302–1310, doi:10.1109/CVPR.2017.143.

[26] T. Simon, H. Joo, I. Matthews, Y. Sheikh, Hand keypoint detection in single images using multiview bootstrapping, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[27] B. Xiao, H. Wu, Y. Wei, Simple baselines for human pose estimation and tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[28] B. Cheng, B. Xiao, J. Wang, H. Shi, T.S. Huang, L. Zhang, HigherHRNet: Scale-aware representation learning for bottom-up human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[29] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi:10.1109/CVPR.2016.91.

[30] S. Ren, K. He, R. Girshick, J. Sun, Faster r-CNN: towards real-time object detection with region proposal networks, IEEE Trans Pattern Anal Mach Intell 39 (6) (2017) 1137–1149, doi:10.1109/TPAMI.2016.2577031.

[31] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5686–5696, doi:10.1109/CVPR.2019.00584.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, u. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS'17, Curran Associates Inc., 2017, p. 60006010.

[33] Y. Chen, H. Ma, D. Kong, X. Yan, J. Wu, W. Fan, X. Xie, Nonparametric structure regularization machine for 2D hand pose estimation, in: IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 370–379, doi:10.1109/WACV45572.2020.9093271.

[34] J. Song, L. Wang, L. Van Gool, O. Hilliges, Thin-slicing network: A deep structured model for pose estimation in videos, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5563–5572, doi:10.1109/CVPR.2017.590.

[35] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[36] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, 2015 IEEE International Conference on Computer Vision (ICCV) (2015) 1520–1528.

[37] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, in: International Conference on Learning Representations (ICLR), 2016.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans Pattern Anal Mach Intell 40 (4) (2018) 834–848, doi:10.1109/TPAMI.2017.2699184.

[39] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arxiv:1706.05587 (2017).

[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

Divyansh Gupta received his B.Eng from Academy of Business and Engineering College, India in Computer Science. Gupta worked as a Senior Software Engineer in a product based firm working on web automation applications. Also, worked on developing methods for Facial Recognition and Pose Estimation during internships. He is currently pursuing his M.S. in Computer Engineering at the Rochester Institute of Technology in Rochester, NY. His research interests include Computer Vision, Deep Learning, and Pose Estimation.

Bruno Artacho received his B.Eng from Sao Paulo State University (2015) and his M.Eng. from Memorial University of Newfoundland (2017), both in Electrical Engineering. Artacho worked at Google and Amazon developing methods and applying his research on Pose Estimation and Segmentation for a diverse set of products; and Transport Canada as part of the Unmanned Aerial System Task Force to assess risk and update the Canadian Air Traffic Policy. He is currently pursuing his Ph.D. in Engineering at the Rochester Institute of Technology in Rochester, NY. His research interests include Computer Vision, Machine Learning, and Human Pose Estimation.

Andreas Savakis is Professor of Computer Engineering and Director of the Center for Human-aware Artificial Intelligence (CHAI) at Rochester Institute of Technology (RIT). He received his Ph.D. in Electrical and Computer Engineering from North Carolina State University. Prior to joining RIT, he was Senior Research Scientist at Kodak Research Labs. His research interests include computer vision, deep learning, machine learning, domain adaptation, object tracking, human pose estimation, and scene analysis. Dr. Savakis has coauthored over 120 publications and is co-inventor on 12 U.S. patents.