2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Dual Regression for Efficient Hand Pose Estimation

Dong Wei[1], Shan An[1,*], *Senior Member, IEEE*, Xiajie Zhang[1], Jiayi Tian[1],
Konstantinos A. Tsintotas[2], Antonios Gasteratos[2], *Senior Member, IEEE*, and Haogang Zhu[3]

***Abstract*— Hand pose estimation constitutes prime attainment for human-machine interaction-based applications. Real-time operation is vital in such tasks. Thus, a reliable estimator should exhibit low computational complexity and high precision at the same time. Previous works have explored the regression techniques, including the coordinate regression and heatmap regression methods. Primarily incorporating ideas from them, in this paper, we propose a novel, fast and accurate method for hand pose estimation, which adopts a lightweight network architecture and a post-processing scheme. Hence, our architecture uses a *Dual Regression* strategy, consisting of two regression branches, namely the coordinate and the heatmap ones, and we refer to the proposed method as DRHand. By carefully selecting the branches' characteristics, the proposed structure has been designed to exploit the benefits of the two methods mentioned above while impoverishing their weaknesses to some extent. The two branches are supervised separately during training, and a post-processing module estimates their outputs to boost reliability. This way, our novel pipeline is considerably faster, reaching 44.39 frames-per-second on an NVIDIA Jetson TX2 graphics processing unit, offering a beyond real-time performance for any custom robotics application. Lastly, extensive experiments conducted on two publicly-available datasets demonstrate that the proposed framework outperforms previous state-of-the-art techniques and can generalize on various hand pose scenarios.**

I. Introduction

Hand pose estimation is crucial for hand gesture recognition as well as for various practical applications, including human-machine interaction [5]–[9], virtual reality [10]–[13] and augmented reality [14]–[16]. In particular, a hand pose estimator aims to detect the 21 or more hand-landmarks (*i.e.*, the fingertips and the joints of each finger), and subsequently return the respective coordinates. In the literature, hand pose estimation mostly leans on 3D landmarks extracted through the depth sensors [17]–[22]. Nevertheless, due to the sensing distance restrictions and the low resolution of typical depth maps, such approaches do not allow satisfactory generalization, while showing great computational complexity. Recently, by leveraging deep convolutional neural networks (CNNs) trained on labeled data, there has been significant progress in restraining this problem. Deep learning architectures have been proved remarkably efficient in capturing descriptive features from high dimensional sensory inputs in a wide variety of computer vision challenges [23], [24]. However, hand pose estimation remains challenging due to the high similarity existing among fingers and the nuisance of self-occlusion. In this work, we focus on $2D$ hand pose estimation from monocular color images in real-time applications.

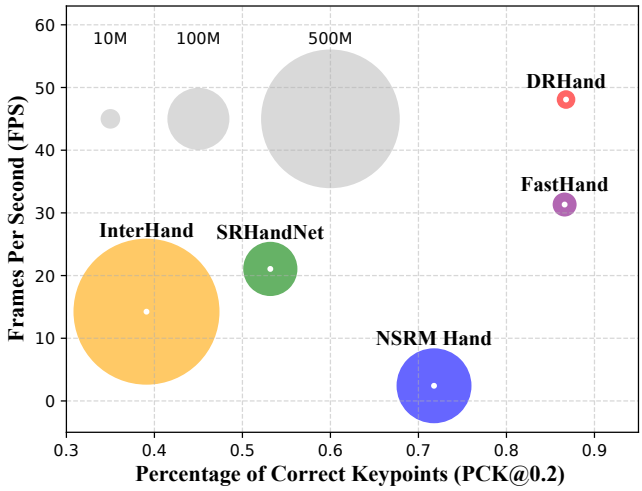

Fig. 1: Comparison between the proposed framework DR-Hand and other state-of-the-art methods, namely SRHand-Net [1], NSRM Hand [2], InterHand [3] and FastHand [4]. The frames-per-second(FPS) are calculated on an NVIDIA GeForce 940MX, while the results of correct keypoint (PCK) are evaluated on the RHD dataset. The size of each circle indicates the respective model size. It is notably that amongst all, the proposed system achieves the highest performance, while requiring the least amount of learnable parameters.

Current state-of-the-art works can be categorized into coordinate-based [2], [9], [25]–[27] and heatmap-based [1], [28]–[31] regression methods. The former directly regress the locations of each hand landmark with clear structural information (*i.e.*, geometric relationship) among the hand joints. The latter generates an $H \times W \times C$ confidence or score map, where C is the number of hand landmarks and $H \times W$ is the size of each landmark's confidence map. The points' value in the confidence map indicates the probability of the respective hand landmark with respect to its current position. The most likely location of one specific landmark

[1] Dong Wei, Shan An, Xiajie Zhang, Jiayi Tian are with Tech. & Data Center, JD.COM Inc., Beijing, 100108, China {weidong53, anshan, zhangxiajie, tianjiayi1}@jd.com;
[2] Konstantinos A. Tsintotas and Antonios Gasteratos are with the Department of Production and Management Engineering, Democritus University of Thrace, Xanthi, 67132, Greece {ktsintot, agaster}@pme.duth.gr
[3] Haogang Zhu is with the School of Computer Science and Engineering, Beihang University, Beijing, 100191, China haogangzhu@buaa.edu.cn
* Corresponding Author

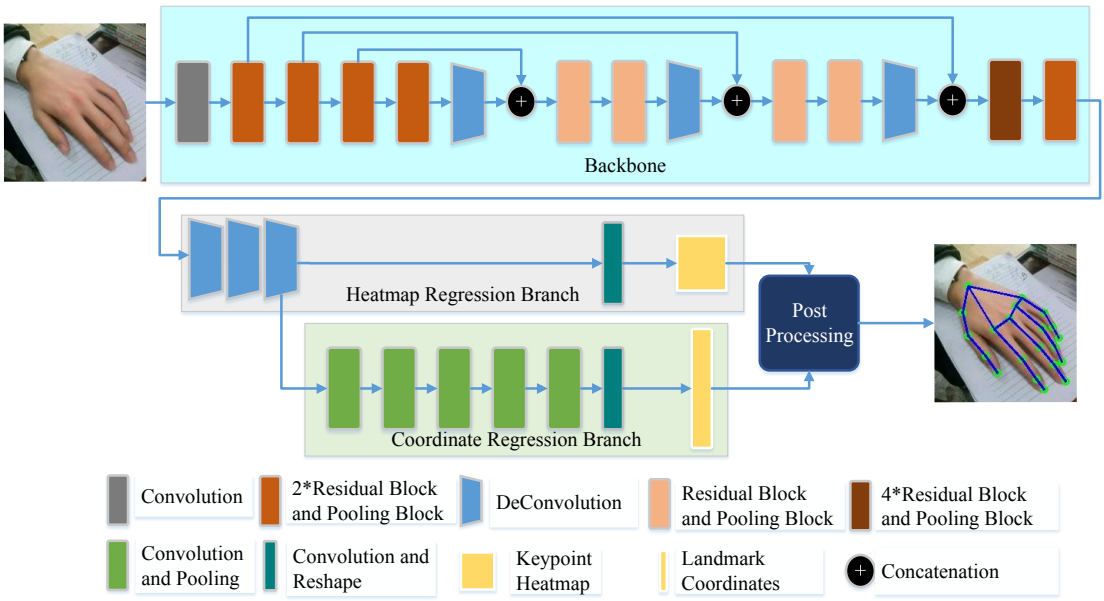

Fig. 2: Illustration of the proposed hand pose estimation network architecture. Given an input image, feature maps containing rich hand information are firstly obtained via the backbone network. Then through the heatmap regression branch the hand joints are predicted; the output of the last deconvolution layer is fed into the coordinate regression branch for predicting the coordinates of the hand-joints. The final results are obtained via a post-processing technique.

is selected as the position of the maximum value in the confidence map. Nevertheless, these two regression methods are both limited for applications and we describe the reasons below. The coordinate-based regression approaches should essentially regress the offset of each landmark relatively to the input image; however, the prediction of a long-distance offset is difficult and thus the network strains to converge. On the contrary, the output of the heatmap-based methods converges easier because the network only needs to generate the probability at each point to the different confidence maps. Yet, they may present a performance reduction due to their lack to memory the hand structure during training.

Nowadays, computational complexity is the primary concern for hand-pose estimation applications. The reason is that these systems are expected to perform in real-time. Unfortunately, previous hand pose estimation and gesture recognition solutions fail to achieve this goal when applied to mobile robots [29], [30], [32]–[35] as they are not designed for embedded devices; hence, they are not light enough to be employed in resource-constrained platforms.

In this paper, we propose a novel network architecture for hand pose estimation based on a dual regression strategy that integrates the coordinate-based and heatmap-based methods. Considering their advantages and disadvantages, we aim to suppress the weaknesses of high complexity and limited interconnection information presented in the heatmap-based solutions, while we try to overcome the low accuracy of the coordinate-based methods. The proposed pipeline comprises the backbone network with the two regression branches, namely the coordinate and the heatmap, as well as a post-processing module. Our extensive experiments on two public benchmark datasets demonstrate the system's real-time inference and high prediction accuracy. As shown in Fig. 1, our proposed method has successfully managed to achieve state-of-the-art performance while giving consideration to memory cost and inference speed. The main contributions of this work are summarized as follows:

- A novel dual regression method for hand pose estimation (DRHand) combining coordinate-based and heatmap-based regression techniques for fast and accurate hand pose estimation.
- A lightweight framework capable of real-time execution on mobile robots with limited computing resources.
- An exhaustive experimental evaluation protocol on the NVIDIA Jetson TX2 graphics processing unit (GPU), demonstrating our framework's effectiveness compared with state-of-the-art methods.

The rest of the paper is organized as follows: at first, in Section II, we review the related literature. Then, in Section III, the proposed DRHand is described in detail, while in Section IV we present the evaluation process and the experimental results. Finally, in Section V, we summarize the conclusions and discuss our research plans.

II. Related Work

In this section, a focused discussion on the most representative approaches for hand pose estimation is presented. We firstly provide a brief review of the coordinate-based regression methods, whereas the heatmap-based ones follow. MediaPipe Hands [27] uses a single shot multi-box detector

(SSD) [25], palm detector and a hand-landmark model to predict high-fidelity $2.5D$ landmarks. A large-scale, multi-view dataset recorded using a green screen and augmented with a complex artificial background is provided by FreiHAND [26]. In addition, a CNN is trained to predict hand poses with a particular generalization ability. Nonparametric structure regularization machine (NSRM) [2] proposes a cascade multi-task architecture, which can jointly learn hand structure and key-point representations. In [9], a single-stage CNN with a "self-attention" module is presented to regress the landmarks' coordinates directly. The utilized attention augmented inverted bottleneck block is used to represent global constraints and correlations between hand-landmarks.

Heatmap-based regression approaches show an improved accuracy compared with the coordinate ones. Multiview bootstrapping [29] follows the structure of convolutional pose machines (CPMs) [28] to predict hand-landmarks. Its performance is improved via $3D$ triangulation obtained by a multi-camera setup. The authors in [30] propose synthetic data generation for monocular $3D$ hand-tracking and deal with challenging occlusions. A network consisting of a hand mask stage followed by a pose prediction one, improves the hand pose estimation accuracy [31], whereas an encoder-decoder network allows for the prediction of hand-bounding box and hand-landmarks simultaneously [1]. InterHand2.6M, a large-scale dataset for hand pose estimation consisting of $2.6M$ single and interacting hand frames with annotations, and InterNet, a $3D$ interacting hand pose estimation network, are provided in [3].

III. Method

A. Network structure

The proposed network architecture is illustrated in Fig. 2 and its implementation details are demonstrated in Fig. 3. Aiming to provide an efficient and low-complexity model with robust feature representation, we have made two notable improvements in our proposed network architecture.

The first one is the usage of the depthwise separable convolution [36] and the corresponding additional operation. Compared with the vanilla convolution [36], depthwise separable convolution can process the same calculation on high-dimensional input data with lower computation cost. Thus, in our proposed DRHand, we adopt the standard residual block [37] with depthwise separable convolution for the primary unit of our backbone network. At the same time, to overcome the possible degradation problem presented in the deep networks, we also add a max-pooling layer, which shares the same input of the current depthwise separable convolution. Then the output of the max-pooling layer and depthwise separable convolution are summed together to feed into next layer as shown in Fig. 3(a). This process acts as a residual block to avoid the degradation problem.

The second one is the concatenation operation between the high-level and low-level features, which share the same resolution. Concatenation operation is widely used in computer vision tasks, such as semantic segmentation [38], [39], to combine the texture information in the lower layers and semantic information in the high layers, and thus can help to obtain a more robust feature representation. We apply this concatenation operation three times in our backbone network as shown in Fig. 2 and Fig. 3(a).

The detailed design for each module in our proposed DRHand is shown in Fig. 3, and the simplified version is demonstrated in Fig. 2. There are total four different types of block or layer in our proposed framework as can be seen from Fig. 3, including "Conv2D", "DSConv2D", "MaxPool" and "DeConv2D". The numbers after each block name in the block, namely "$(k, k), D$", are the detailed parameters of each block, in which (k, k) is the filter size, and D is the output dimension for the current block. In particular, "Conv2D" is a 2-dimensional vanilla convolution block. "DSConv2D" is a depthwise separable convolutional residual block. The output and the input of the depthwise separable convolution are added together and then fed into the next layers. "MaxPool" is a max-pooling layer, and "DeConv2D" is a deconvolutional block. Except for the last "Conv2D" in the heatmap regression branch and the last "DSConv2D" in the coordinate regression branch as illustrated in Fig. 3, all other convolutions in "Conv2D", "DSConv2D" and "DeConv2D" are followed by a BatchNorm layer and a ReLU activation layer.

1) The backbone network: The primary demand for an efficient method is a backbone network with the ability to extract robust features. As described before, we utilize the standard residual block as the basic unit. The residual block contains a convolutional layer, As shown in Fig. 3, except for the first "Conv2D" block, every other convolutional layer adopts the depthwise separable convolusion. The input data are firstly down-sampled using depthwise separable convolution and then are up-sampled with deconvolution. Moreover, feature maps, between the scales of $[32, 16, 8]$, are connected across layers as illustrated in Fig. 2 and Fig. 3. In this way, the low-level texture features and high-level semantic features are merged in a more reliable fashion, yielding to robust feature extraction. As input, the backbone network accepts a 256×256 image. Furthermore, the first convolutional kernel size is 3×3 while the size of the rest kernels is 5×5. Thus, a feature map of $8 \times 8 \times 288$ is generated for each incoming image after passing through the backbone network.

2) The heatmap regression branch: The input of this branch consists of the generated feature maps from the backbone network. Next, three successive deconvolution layers are adopted to make our model's forward speed as fast as possible as shown in Fig. 3(b). Subsequently, the final heatmap, which is the size of $64 \times 64 \times 21$, is acquired with the most minor calculations and number of parameters. Finally, an L_2 loss is added at the end of this branch, which is defined as:

$$\mathcal{L}_{heatmap} = \frac{1}{2}(W_1(H(B(\boldsymbol{x}))) - y)^2, \tag{1}$$

where $\boldsymbol{x}$ and y denote the incoming image data and its corresponding ground truth label, respectively. $B(*)$ stands

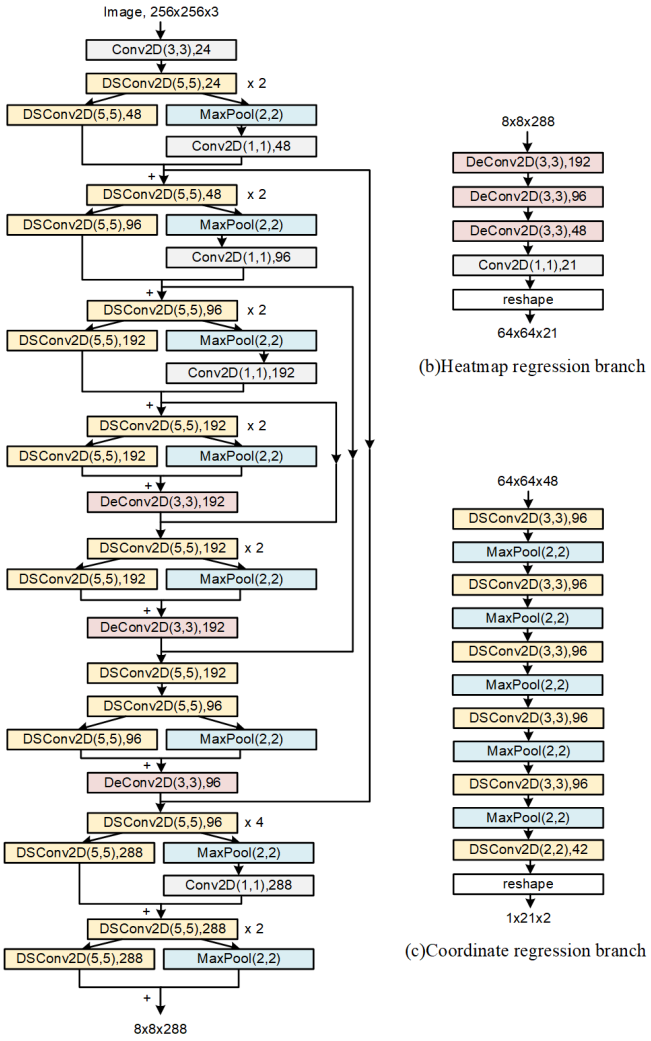

Fig. 3: Implementation details of our proposed backbone network. “Conv2D" indicates the vanilla convolution, “DSConv2D" indicates the depth-wise separable convolution, and “DeConv2D" indicates the deconvolution.

for the backbone network, $H(*)$ and $W_1(*)$ are the three deconvolution layers and the last convolution layer in the heatmap regression branch. Given the input image $\boldsymbol{x}$, the final output from the heatmap regression branch is denoted as $W_1(H(B(\boldsymbol{x}))$.

3) The coordinate regression branch: This branch receives the intermediate product from the heatmap regression branch before the last convolution layer. Since we aim to construct a pipeline with low computational complexity, we utilize only five consecutive deeply separable convolutional layers and one convolution with pooling layers. Thus, the coordinates of the 21 hand-landmarks are received with the minimum complexity. Wing loss [40] is also adopted, which is defined as:

$$\mathcal{L}_{coord} = \begin{cases} w \ln\left(1 + |z|/\epsilon\right) & \text{if } |z| < w \\ |z| - C & \text{otherwise} \end{cases}. \quad (2)$$

Given the input image $\boldsymbol{x}$, the L_1 distance between the output from coordinate regression branch and the corresponding ground truth label is denoted as z, where $z = |W_2(R(H(B(\boldsymbol{x})))) - y|$. R indicates the five convolution and pooling layers in the coordinate regression branch, and W_2 is the last convolution and reshape layer in the coordinate regression branch. w is a non-negative parameter limiting the range in $(-w, w)$, ϵ sets the curvature of the nonlinear region and $C = w - w\ln(1+w/\epsilon)$ is a constant that smoothly links the piecewise-defined linear and nonlinear parts.

B. Post processing

Owing to the proposed dual regression strategy, the system delivers two landmark results from the respective regression branches. In the post-processing module, both outcomes are estimated to determine the final output. As mentioned in Section I, the results from the coordinate regression technique show a straightforward geometric relationship among the joints; yet, their localization accuracy is poor. On the contrary, heatmap-based models present higher precision but lack of geometric interconnection information. To combine their advantages and provide better results based on the reliability of both branches' output, we propose the following scheme to determine the final outcome:

- Given an input image, each branch generates its landmark coordinates.
- The Euclidean distance between the two branches' predictions is calculated and referred to as $[d_1, d_2, ..., d_{21}]$.
- The distances are compared with a predefined threshold α, which is the length of a hand knuckle, *i.e.*, the median of the predicted length of all hand knuckles. Then for each hand-image, we choose the i-th output coordinate from the heatmap regression branch as our final i-th landmark prediction result only if $d_i < \alpha$. Otherwise, the i-th output of the coordinate regression branch is used as the final landmark prediction coordinate.

The reason behind the proposed post-processing scheme is that in most cases, the results of the heatmap regression are more accurate than the coordinate regression's results. Therefore, when the distance of a single hand landmark coordinate output from the two branches is relatively small, the result of the heatmap regression branch is closer to the ground truth label. However, the lack of structural information in the heatmap regression branch may prevent the system from accurately predicting every single landmark coordinate. As a consequence of this phenomenon, the Euclidean distance d_i between the two outputs can be considerable. Then, at this time, we choose the outcome from the coordinate regression branch as presumably being a more reliable one.

IV. Experiments

In this section, extensive experiments on two public datasets are conducted to evaluate and demonstrate the effectiveness of the proposed DRHand.

A. Experimental setup

1) Datasets: The YouTube3D Hands [41] and GANerated Hands [30] image-sequences have been utilized as training data. The former is a $3D$ hand pose estimation benchmark, containing 102 videos for training and 7 videos for testing. Since we aim for a $2D$ hand pose estimation, the $3D$ coordinate labels are projected onto the $2D$ space. This way, a novel dataset involving original images with their respective $2D$ labels is formed, ready for use in any $2D$ hand pose estimation scenario. We denote this new image sequence, containing 47,125 images, as YouTube2D Hands. The second dataset, GANerated Hands [30], is a benchmark composed of synthetic images with various gesture types and more accurate annotations when compared with other real-world datasets. This image sequence consists of 141,449 elements. We use STB [42] and RHD [32] benchmark datasets as our evaluation datasets; the images included in these two datasets are 6,000 and 2,727, respectively.

2) Implementation details: The proposed network was implemented using TensorFlow [43], and trained on four NVIDIA Tesla P40 GPUs. The whole network parameters were optimized through Adam optimizer [44]. The training process followed a two-stage procedure. At first, only the heatmap regression and the backbone network were optimized with an initial learning rate of $1e^{-3}$ and a batch size of 64. This training process lasts for 11 epochs till the heatmap regression branch converges. Subsequently, the coordinate regression branch was combined with the previously trained system for further optimization. Finally, the whole network was optimized with a lower learning rate of $1e^{-4}$ and a larger batch size of 256.

3) Evaluation metrics: Three different metrics are used to evaluate the proposed method comprehensively. In particular, the sum of squares error (SSE), the end-point error (EPE), and the probability of correct keypoint (PCK) within a normalized distance threshold are selected and defined as:

$$SSE = \frac{\sum_{s=1}^{N}(\sum_{i=1}^{21}((\frac{y_{si}-\widehat{y}_{si}}{\max(w,h)})^2))}{N}, \tag{3}$$

$$EPE = \frac{\sum_{s=1}^{N}(\sum_{i=1}^{21}\left\|\frac{(y_{si}-y_{s0})-(\widehat{y}_{si}-\widehat{y}_{s0})}{\max(w,h)}\right\|)}{21 \times N}, \tag{4}$$

$$PCK_{\sigma}^{i} = \frac{1}{N}\sum_{s=1}^{N}\mathbb{1}(\frac{\|y_{si}-\widehat{y}_{si}\|}{\max(w,h)} \leq \sigma), \tag{5}$$

$$PCK_{\sigma} = \frac{\sum_{i=1}^{21} PCK_{\sigma}^{i}}{21}, \tag{6}$$

The term y_{si} is the ground truth of landmark i and $\widehat{y}_{si}$ is the predicted coordinate, while i represents the landmark index, *i.e.*, $i \in [1..., 21]$, and s is the input hand image index. N denotes the number of samples in the dataset and w and h are the width and height of the original images, respectively. In Eq. 5, $\mathbb{1}(\cdot)$ denotes the indicator function and σ is a predefined threshold. The indicator function equals to 1 only when L_2 distance between the predicted landmark and the ground truth is less than σ, otherwise it equals to 0. In Eq. 6, PCK_{σ}^{i} represents the PCK_{σ} metric of landmark i with threshold σ. In our experiments, σ is set to 0.2.

B. Experimental results

Aiming to further elaborate the efficiency of the proposed framework, we attempt a comparison with other state-of-the-art pipelines, namely SRHandNet [1], NSRM Hand [2], InterHand [3], MediaPipe Hands [27] and FastHand [4].

1) System complexity evaluation: Table I presents the speed, in terms of frames-per-second (FPS), for the aforementioned methods. Note that MediaPipe Hands is not involved in this comparison since no open-source code for evaluation is provided. Nevertheless, for the performance comparison presented in Table III, we have reimplemented a demo version of this framework without the acceleration operations customized by the authors. Experiments are run on two devices, an NVIDIA GeForce 940MX GPU of a laptop and an NVIDIA Jetson TX2 GPU. Notably, our method outperforms other methods in inference time, exhibiting twice as high execution times as SRHandNet and NSRMHand. Furthermore, a model of 7.5MB is generated, which is 10.43% of the size for SRHandNet and 1.38% of InterHand. Though the proposed system is not comparable with MediaPipe Hands in terms of model size, we achieve better results on STB and RHD datasets (see Table III). Besides that, the proposed method still uses the lowest calculations of 0.95 GFlops. Yet, these metrics for SRHandNet and InterHand are hard to be obtained because details are not provided by the authors. The improvements of the inference time and space usage are attributed to the utilization of the depthwise separable convolution and the delicate design of our DRHand as discussed in Section III-A.

2) Comparison on benchmark datasets: As presented in Table III, the proposed DRHand and FastHand reach state-of-the-art performances. In particular, the former outperforms other techniques on the RHD dataset, where the SSE error is only $1/3$ of those in InterHand. A marginal improvement is achieved on the STB dataset, mainly because since most images in this dataset only contain one hand with fixed background and fixed camera angle. Thus, the STB dataset is well segmented, relatively simpler, and so its results are almost saturated. On the other hand, the RHD dataset is more challenging. Its synthetic images are widely different in background and camera angle, while two hands appear in the same image concurrently. Yet, our pipeline outperforms the other pipelines.

3) Qualitative results: A qualitative comparison is also provided in Fig. 4, wherein the last row provides the ground truth annotations. It is worth noting that our DRHand presents the closest outcomes to the ground truth. In particular, the prediction results of SRHandNet fail to detect all 21 joints of the second and third samples, while NSRM Hand and InterHand give some joint coordinates incorrectly, as can be seen from some predicted joints' locations. Compared with our previous work FastHand, we managed to improve

TABLE I: Comparison in inference runtime (frames-per-second - FPS) of different hand pose estimation methods.

Model	SRHandNet [1]	NSRM Hand [2]	InterHand [3]	FastHand [4]	Proposed
NVIDIA GeForce 940MX GPU	21.06 FPS	2.42 FPS	14.24 FPS	31.33 FPS	**48.08 FPS**
NVIDIA Jetson TX2 GPU	19.16 FPS	3.65 FPS	7.77 FPS	25.05 FPS	**44.39 FPS**

TABLE II: Comparison in terms of model size and calculations of different hand pose estimation methods.

Model	SRHandNet [1]	NSRM Hand [2]	MediaPipe Hands [27]	InterHand [3]	FastHand [4]	Proposed
Size (MB)	71.90	139.7	3.9	541.7	13.0	**7.5**
Calculations (GFlops)	–	102.76	–	–	2.89	**0.95**

TABLE III: Comparative results of the baseline methods against the proposed method. The green values denote the best result and the blue ones the second best. ↑ signifies the higher the better, whereas ↓ marks the lower the better.

Dataset	Metric	SRHandNet [1]	NSRM Hand [2]	MediaPipe Hands [27]	InterHand [3]	FastHand [4]	Proposed
STB [42]	$SSE \downarrow$	–	0.7078	0.9435	0.4853	0.3490	0.3494
	$EPE \downarrow$	–	0.1326	0.1522	0.1302	0.1317	0.1300
	$PCK@0.2 \uparrow$	0.8526	0.7246	0.7032	0.8245	0.8948	0.8910
RHD [32]	$SSE \downarrow$	–	2.5613	1.6343	1.9929	0.6368	0.6247
	$EPE \downarrow$	–	0.1953	0.2133	0.2630	0.0968	0.0940
	$PCK@0.2 \uparrow$	0.5317	0.7177	0.6927	0.3910	0.8661	0.8677

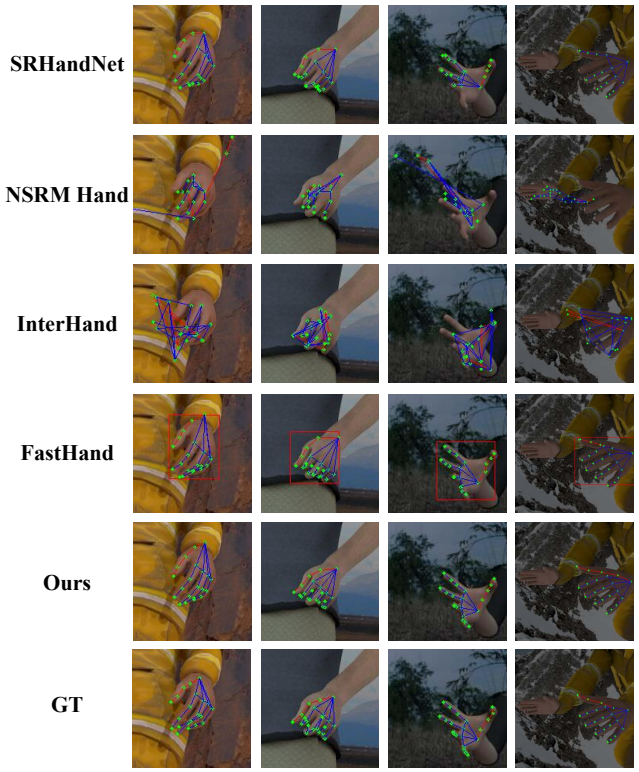

Fig. 4: Qualitative results of the proposed method compared with other approaches on the RHD dataset [32].

our previous achievement by providing more accurate results, according to the comparison of the second sample.

V. Conclusion And Future Work

This paper proposes a novel and lightweight network architecture for hand pose estimation, which is able to be employed on embedded platforms. Our method DRHand has been carefuly designed for efficient inference. As we combine heatmap-based and coordinate-based regression methods, we build a dual regression structure that exploits each technique's advantage and overcome their disadvantages at the same time. The branches' outputs are carefully selected through a post-processing technique, achieving this way to predict quickly and accurately the hand-landmarks.

In our future work, we intend to implement a more promising framework that would combine the coordinate and heatmap regression techniques more effectively, as well as to incorporate a new post-processing strategy to deal with the output of these two methods. Additionally, we aim to collect a novel dataset that will help the research community.

Acknowledgment

This work was supported by a grant from the National Key R&D Program of China (Grant No. 2021YFB2700300) In addition, this research has been partially co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH - CREATE - INNOVATE grant number [T2EDK-01862].

REFERENCES

[1] Y. Wang, B. Zhang, and C. Peng, "SRHandNet: Real-time 2d hand pose estimation with simultaneous region localization," *IEEE Trans. Image Processing*, vol. 29, pp. 2977–2986, 2019.

[2] Y. Chen, H. Ma, D. Kong, X. Yan, J. Wu, W. Fan, and X. Xie, "Nonparametric structure regularization machine for 2d hand pose estimation," in *IEEE Winter Conf. Applications of Computer Vision*, 2020, pp. 381–390.

[3] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "InterHand2.6M: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," *arXiv preprint arXiv:2008.09309*, 2020.

[4] S. An, X. Zhang, D. Wei, H. Zhu, J. Yang, and K. A. Tsintotas, "Fasthand: Fast monocular hand pose estimation on embedded systems," *Journal of Systems Architecture*, vol. 122, p. 102361, 2022.

[5] K. Ehlers and K. Brama, "A human-robot interaction interface for mobile and stationary robots based on real-time 3d human body and hand-finger pose estimation," in *IEEE 21st Int. Conf. Emerging Technologies and Factory Automation*, 2016, pp. 1–6.

[6] O. Mazhar, S. Ramdani, B. Navarro, R. Passama, and A. Cherubini, "Towards real-time physical human-robot interaction using skeleton information and hand gestures," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2018, pp. 1–6.

[7] J.-Y. Chang, A. Tejero-de Pablos, and T. Harada, "Improved optical flow for gesture-based human-robot interaction," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2019, pp. 7983–7989.

[8] I. Kansizoglou, L. Bampis, and A. Gasteratos, "An active learning paradigm for online audio-visual emotion recognition," *IEEE Trans. Affective Computing*, 2019.

[9] 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.

[10] J. Segen and S. Kumar, "Gesture vr: vision-based 3d hand interace for spatial interaction," in *Proc. ACM Int. Conf. Multimedia*, 1998, pp. 455–464.

[11] R. O'Hagan, A. Zelinsky, and S. Rougeaux, "Visual gesture interfaces for virtual environments," *Interacting with Computers*, vol. 14, no. 3, pp. 231–250, 2002.

[12] M. Kolsch, *Vision based hand gesture interfaces for wearable computing and virtual environments*. University of California, Santa Barbara, 2004.

[13] K. M. Sagayam and D. J. Hemanth, "Hand posture and gesture recognition techniques for virtual reality applications: a survey," *Virtual Reality*, vol. 21, no. 2, pp. 91–107, 2017.

[14] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, "Fingartips: gesture based direct manipulation in augmented reality," in *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, 2004, pp. 212–221.

[15] W. Hürst and C. Van Wezel, "Gesture-based interaction via finger tracking for mobile augmented reality," *Multimedia Tools and Applications*, vol. 62, no. 1, pp. 233–258, 2013.

[16] T. Lee and T. Hollerer, "Handy ar: Markerless inspection of augmented reality objects using fingertip tracking," in *2007 11th IEEE Int. Symp. Wearable Computers*. IEEE, 2007, pp. 83–90.

[17] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect." in *The British Machine Vision Conference*, vol. 1, no. 2, 2011, p. 3.

[18] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-icp for real-time hand tracking," in *Computer Graphics Forum*, vol. 34, no. 5, 2015, pp. 101–114.

[19] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei *et al.*, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Ann. ACM Conf. Human Factors in Computing Systems*, 2015, pp. 3633–3642.

[20] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. IEEE Int. Conf. Computer Vision*, 2015, pp. 3316–3324.

[21] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3d hand pose estimation from single depth images using multi-view cnns," *IEEE Trans. Image Processing*, vol. 27, no. 9, pp. 4422–4436, 2018.

[22] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Self-supervised 3d hand pose estimation through training by fitting," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 10 853–10 862.

[23] S. An, H. Zhu, D. Wei, and K. A. Tsintotas, "Fast and incremental loop closure detection with deep features and proximity graphs," *arXiv preprint arXiv:2010.11703*, 2020.

[24] I. Kansizoglou, L. Bampis, and A. Gasteratos, "Deep feature space: A geometrical perspective," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2021.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Computer Vision*, 2016, pp. 21–37.

[26] 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 *Proc. IEEE Int. Conf. Computer Vision*, 2019, pp. 813–822.

[27] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[28] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 4724–4732.

[29] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153.

[30] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 49–59.

[31] Y. Wang, C. Peng, and Y. Liu, "Mask-pose cascaded cnn for 2d hand pose estimation from single color image," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 29, no. 11, pp. 3258–3268, 2018.

[32] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *Proc. IEEE Int. Conf. Computer Vision*, 2017, pp. 4903–4911.

[33] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *Proc. Eur. Conf. Computer Vision*, 2018, pp. 666–682.

[34] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz, "Hand pose estimation via latent 2.5 d heatmap regression," in *Proc. Eur. Conf. Computer Vision*, 2018, pp. 118–134.

[35] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3d hand shape and pose estimation from a single rgb image," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019, pp. 10 833–10 842.

[36] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[39] O. Ronneberger, P. Fischer, and 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.

[40] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2018, pp. 2235–2245.

[41] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2020, pp. 4990–5000.

[42] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang, "3d hand pose tracking and estimation using stereo matching," *arXiv preprint arXiv:1610.07214*, 2016.

[43] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.