Hand LightWeightNet: an optimized hand pose estimation for interactive mobile interfaces

Jamal Firmat Banzi[1], Stanley Leonard[2]
[1]Department of Tourism and Recreation, College of Forestry, Wildlife and Tourism, Sokoine University of Agriculture, Morogoro, Tanzania
[2]Department of Computer Science and Engineering, College of Engineering and Technology, Mbeya University of Science and Technology, Mbeya, Tanzania

Article Info

Article history:

Received Jun 27, 2023
Revised Sep 7, 2023
Accepted Sep 12, 2023

Keywords:

Convolutional neural network
Deep learning
Feature extractor
Hand pose estimation
Human computer interaction

ABSTRACT

In this paper, a hand pose estimation method is introduced that combines MobileNetV3 and CrossInfoNet into a single pipeline. The proposed approach is tailored for mobile phone processors through optimizations, modifications, and enhancements made to both architectures, resulting in a lightweight solution. MobileNetV3 provides the bottleneck for feature extraction and refinements while CrossInfoNet benefits the proposed system through a multitask information sharing mechanism. In the feature extraction stage, we utilized an inverted residual block that achieves a balance between accuracy and efficiency in limited parameters. Additionally, in the feature refinement stage, we incorporated a new best-performing activation function called “activate or not” ACON, which demonstrated stability and superior performance in learning linearly and non-linearly gates of the whole activation area of the network by setting hyperparameters to switch between active and inactive states. As a result, our network operated with 65% reduced parameters, but improved speed by 39% which is suitable for running in a mobile device processor. During experiment, we conducted test evaluation on three hand pose datasets to assess the generalization capacity of our system. On all the tested datasets, the proposed approach demonstrates consistently higher performance while using significantly fewer parameters than existing methods. This indicates that the proposed system has the potential to enable new hand pose estimation applications such as virtual reality, augmented reality and sign language recognition on mobile devices.

Corresponding Author:

Jamal Firmat Banzi
Department of Tourism and Recreation, College of Forestry, Wildlife and Tourism, Sokoine University of Agriculture
Morogoro, Tanzania
Email: jamalbanzi@sua.ac.tz

1. INTRODUCTION

Hand pose estimation is the first step for various natural hand related interactions such as human-machine interaction, hand gesture recognition, and virtual space interaction which delivers greater user experience. Hand pose estimation allows users to manipulate virtual objects with direct natural hands instead of using wearable gadgets controlled via 2D interfaces. Indeed, there have been considerable research efforts in this domain for the past two decades. Hand pose estimation has been proven to be natural with the advent of holographic display [1].

Several approaches have been presented in the literatures to address challenges of this research domain with different objectives in place; some of the remarkable objectives are remote surgery [2], hand gesture recognition such as in automotive industry [3], sign language recognition [4], traffic sign analysis and in virtual space interaction such as augmented reality (AR) and virtual reality (VR). The advent of depth sensors [5], [6], advance in vision algorithms, and increase in processing power have revolutionized the research domain making it possible for vision based approach to comes into the mainstream of hand pose estimation especially for the AR systems where wearable glove or mask is not possible, primarily for casual usage and mobile applications. Knowing this, leading smart phone manufacturers have begun integrating a diverse array of depth sensors into their latest models, thereby expanding accessibility to VR and AR for users across the globe. For instance, Samsung included 3D time of flight (ToF) depth sensor in galaxy S23 ultra, Apple Inc. integrated direct (D-ToF) in the iPhone 13 to 15 series and had also introduced structured light (SL) depth sensor to iPhone X, and Huawei has started using the same type of sensor on P30 pro model. As the availability of these sensors is increasing, the scope of applications utilizing depth data is also expanding at a rapid pace.

Recently, deep learning has been deemed the gold standard in computer vision, hand pose estimation being one of them. Based on the task requirement perspective, a deep learning based approach to hand pose estimation generates joint coordinates directly from its fully connected layers [7], [8] or creates a probability heatmap of each joint following its location [9], [10]. Both approaches pass through complex stages from pre-processing to final pose estimates which require large computational resources along the way. This makes most algorithms to be confined to standard computers which are computationally expensive to run smoothly on mobile devices or other handheld devices. This work presents lightweight network (LightWeightNet) hand pose estimation (HPE), a hand pose estimation based on convolutional neural network finetuned and optimized to mobile phone processors to minimize the computational cost and allow mobile phone users enjoy an immersive experience. The first attempt was the work of [11], where a CrossInfoMobilenet was presented replacing a computational critical CrossinfoNet [12]. Herein, we present an improved version of CrossInfoMobileNet with an additional depth-wise separable convolutions which greatly lowers the computational cost of a general convolutional neural network (CNN) model used in MobileNet3 [13]. We further add a multi-spectral attention layer prior to its fully connected layer to reveal more frequent domain information and hence delivers the best performance of the network. Lastly, we pruned several parameters as explained in section 2 so that a lightweight HPE could run smoothly on a mobile phone when the hand is placed between 45° to 90° Infront of a hand-held device, as displayed in Figure 1.

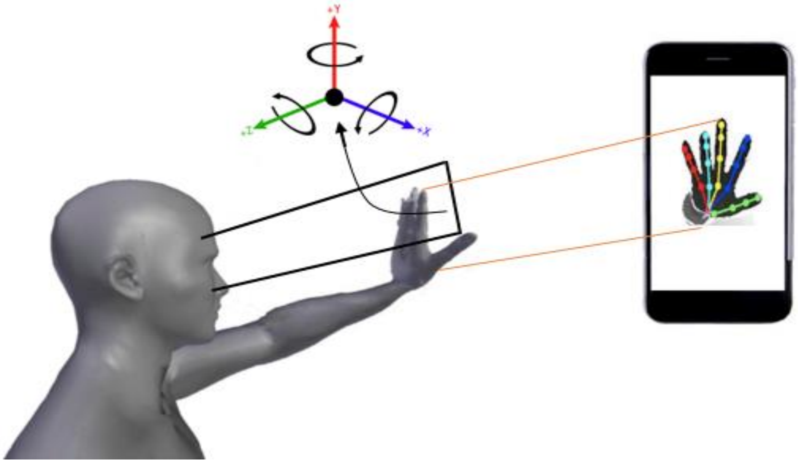

Figure1. Set up of the developed hand pose estimation system

From the Figure 1, a hand was set at the initial position of the interacting device i.e., mobile phone. Hand configuration allows rotation on three axes in ranges x, y, z with respect to realistic positions. This further network improvement presented in this paper has significantly increased both the performance of the network and the accuracy of the HPE system. To demonstrate the performance of the proposed approach, a thorough evaluation was conducted on the three often used publicly available hand pose datasets. The gist of our work is presented in the summary below: i) Integrating the latest version of MobileNetV3 [13] to CrossinfoNet [11] along with multi-spectral attention mechanism in one pipeline while retaining multidomain information during parameter reduction and hence improve performance; ii) Transforming a rectified linear unit (ReLU) activation to activate or not (ACON) activation for additional performance; and iii) Improving the whole training procedure of the network to cover additional information from the available data while using few parameters to allow a network to run in mobile devices framework.

The rest of the paper is organized: section 2 introduces the method we use, approaches to hand pose estimation based on the LightWeightNet architecture, our modifications and evaluation of the proposed architecture with the recent approaches on public benchmark datasets. Section 3 concludes the paper.

2. METHOD

In this paper, an architecture based on modelling human hand kinematics and morphology is described. This architecture is based on a model-based approach and is a good choice since it allows logical divisions between different parts of the hand such as fingers and palm. It is therefore possible to extract multiple regions per finger as in [5], to locate different finger functions while considering hand kinematic constraints. The foundation of this architecture comes from the previous hand model-based approaches that utilizes residual network (ResNet) blocks throughout its architecture [14]. To expand its usefulness [6], draws inspiration from previous hand model-based approaches and utilizes ResNet block to perform a multitask information sharing mechanism. Afterwards, Xiong *et al.* [15] applied multitask information sharing approach to human pose estimation where two fully trained convolutional neural network were fused with different resolutions to improve accuracy. However, training deep neural network has been time and memory consuming, making it impossible to apply to mobile devices.

In mobile devices application, it is imperative to reduce the number of deep neural network (DNN) model parameters to achieve a lesser memory usage while maintain promising performance. In an effort to reduce the numbers of parameters, SqueezeNet [16] succeeded to achieve AlexNet [17] level accuracy with 50 times lesser parameters. Thereafter large bodies of work presented LightWeightNet [18], [19] with small models attempting to make them suitable for mobile devices. Ge *et al.* [20] pioneered to propose a weighted average network that utilized few numbers of parameter but could attain desired accuracy. Their network provides the best performance in terms of inference speed and model size. The Ge's network achieved a small size by significantly reducing the parameter count to less over two million. Approximately 66% of these parameters were allocated to the latent heat regression network, while the feature extraction network utilized roughly half a million parameters. Thereafter, Liu *et al.* [21] proposed a regression network containing similar features and an additional fingertip refinement module using neighboring points of the similar finger location. In achieving parameter reduction, Ge *et al.* [22] proposes a modified PointNet network to estimate 3D hand joint positions from a 3D point cloud.

In recent literature, optimization is conducted by minimizing the number of multiply-add (MAD) operations instead of directly reducing the number of parameters of the network. This opens a new way of reducing network size of different models to suit mobile devices platform. The most recent CNN with mobile backbones is from the MobileNetV3 network developed by [13]. The MobileNet architecture has advanced intensively in the last 3 years and since then, three versions have been released [13], [18], [23]. The architecture is modular as it is for ResNet that's means the number of layers and blocks can be adjusted to control the performance and accuracy of the network. In the next section we explain how we explore MobileNetV3 adjust some layers to fit with our requirements, optimize and present it in a very lightweight form while maintaining intended accuracy.

2.1. The LightWeightNet architecture

The LightWeightNet was developed from the original CrossInfoNet [11], with feature extraction performed using ResNet-50 residual blocks [14]. The extracted features accounts for 61% of the multiply-add operations of the entire network. Since those residual blocks were not designed to specifically run on mobile devices, Du *et al.* [11] replaced them by a more recent and more efficient design; i.e., MobileNetV3 [13]. In their paper, they utilized less multiply-add operations by 79% and less parameters by 28% compared to CrossInfoNet version. This paper presents a newly developed feature extractor for LightWeightNet which comes as an improvement of CrossInfoMobileNet [13]. The CrossInfoMobileNet is chosen because of its success to utilize 70% less multiply-add operations (70.2×10^6 vs. 74.4×10^6) and 27% less parameters compared to CrossInfoNet version.

2.2. Proposed approach

Herein, we propose a LightWeightNet version of CrossInfoMobileNet suitable for running in a mobile device called LightWeightNet model presented in Figure 2. The model attempts to address the problem of accurate hand joint location of a human hand via hand pose estimation. The first part of the

Figure comprises a feature extraction module, in which heat-maps are incorporated as constraints to enhance the learning of feature maps and obtain all initial joint features.

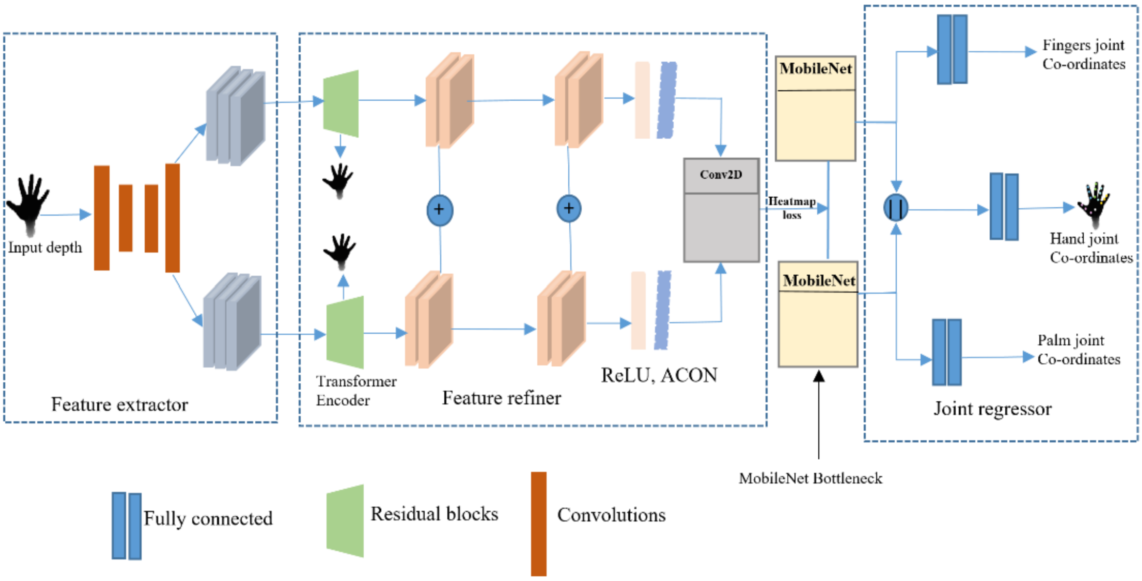

Figure 2. LightWeightNet overview showing feature extraction, feature refinement and joint regression stages which an input hand depth is passing before joint positioning

The subsequent section involves a feature refinement module, which further breaks down the task into two sub-tasks: one focusing on estimating the palm joints and the other on estimating the finger joints. The whole process can be summarized as follows; Given a preprocessed hand depth image, a ResNet is primarily used to encode the input depth and generate individual hand frame embeddings. Afterwards, a transformer encoder learns the contextual sequential both temporal and angular information as in many recent literatures [10], [24], [25] on computer vision applications. Then the transformer generates 2D hand joint locations as an offset. The 2D joint estimate is converted to 3D hand poses by the regression layer connected directly to fully connection layer which maps the hand and the joint position to produce the desired output.

Specifically, a fully connected layer takes the input as a set of features extracted from the hand depth image and produces the output as a set of predicted joint angles or coordinates that define the pose of the human hand. By using a fully connected layer as the final layer of the network, the model can learn to map the high-dimensional input features to the low-dimensional output pose space. In the next sub-sections, the main components of lightweight pose estimator are discussed in details.

2.3. Network architecture

The whole network pipeline is presented in Figure 2. This network is a newly developed architecture obtained by integrating CrossInfoNet and MobileNetV3 into one pipeline. The architecture generally comprises of three components: feature extraction, feature refinement and joint regression. It combines the structural design of both CrossInfoNet and MobileNetV3 as in [11] but we made further modification from the feature extraction stage to build a more efficient architecture as explained in the subsequent sections. The modified architecture has an improved performance of approximately 1.5 times compared to recent CrossInfoMobileNet.

2.4. Feature extraction

Recently, the MobileNet architectures demonstrated remarkable efficiency and outstanding performance in many vision based tasks with a minimal data volume than classical network backbone, such as ResNet [14]. In this study, the local and global high dimensional information present in the input depth data are efficiently extracted using a series of residual block with top-down approach as in many recent deep learning and neural network literatures [16], [26]. We completely redesigned feature extractor backbone to ensure a reduced weight of the network and achieve the superior accuracy. In feature extraction stage, we firstly utilize an inverted residual block to achieve a balance between accuracy and efficiency in limited parameters as shown in Figure 3. We leverage an inverted residual block bottleneck consisting of input, and extension. Our design leverages 5×5 kernels instead of 7×7 used in literature [27]–[29], to guarantee the light weighted network while retaining feature information contrary to many applied MobileNetV3 series [13]. We then replace ReLU activation with the new ACON activation function which has lesser computational requirement and hence increases speed of model training while ensures transformed parameters retains its features. Finally, we integrate the model with a co-ordinate attention mechanism module before 1×1 convolution at the end of bottleneck, instead of the squeeze-and-excitation SE block used by literature [30]–[32]. The inclusion of coordinate attention mechanism module enhances the reading of joint features in the pose estimator, surpassing the performance of the conventional spatial attention mechanism modules, such as SE-block or convolutional block attention module (CBAM).

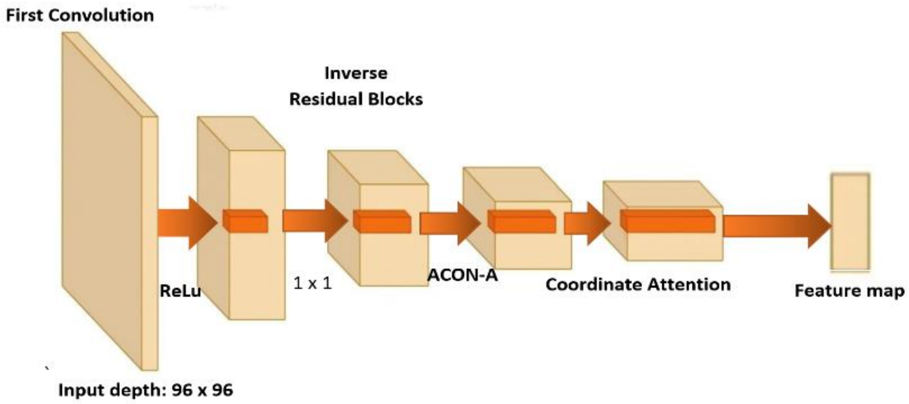

Figure 3. Feature extractor part of the proposed network

Figure 3 presents feature extractor of the LightWeightNet. Each block represents the output tensor of each layer. The thickness represents the number of channels in 2D and the last two blocks are tensors processed by the co-ordinate and multispectral attention mechanism. To this end, our architecture demonstrated superior memory efficiency and experimental outcomes compared to the traditional residual modules. Moreover, our architecture leverages the feature map connectivity to further enhance the capabilities of the feature extractor module. By establishing effective connections to the feature maps, we are able to extract and integrate information more efficiently, leading to improved performance in hand feature extraction. This feature map connectivity plays a crucial role in achieving better results compared to traditional approaches relying solely on residual modules.

2.4.1. ACON activation

The ACON activation function is a relatively new activation function proposed in 2021 by the research team [33]. ACON stands for "active convolutional networks," and it is designed to improve the performance of convolutional neural networks (CNNs) by addressing some of the limitations of existing activation functions, such as ReLU, parametric rectified linear unit (PReLU), and Swish. The ACON activation function has been shown to be effective in improving the performance CNNs on various computer vision tasks, including 3D hand pose estimation. The recent study by Xiong *et al*. [15] proposed the use of the ACON activation function and they achieved promising results. Motivated by the function $max(x_1, x_2, x_3 \ldots x_n)$ that defines the range of a maximum value of specific n parameters, Tompson *et al*. [34] utilized ACON activation to determine a new smooth and differentiable approximation function. In this paper, we define ACON function transformation with an improved transformation parameter by (1).

$$S_\beta(x_1, \ldots, x_n) = \frac{\sum_{i=1}^{n} x_i e^{\beta x_i}}{\sum_{i=1}^{n} e^{\beta x_i}} \tag{1}$$

where β is a transformation parameter. Transformative parameters behave inversely proportional with the opposite extremities. For example, when β approaches infinity, the scaling factor S_β attains its maximum value. Similarly, when β approaches 0, S_β approaches arithmetic mean. Now considering its behavior, an addition structure from the commonly used activation functions such as linear activation are substituted to the general activation function S_β. The basic structure for linear activation is given as $max(\eta_a(x), \eta_b(x))$, where $\eta_{a,b}(x)$ represents a linear function such as ReLU function with states $max(x, 0)$. Then an appropriate smooth differentiable function of $max(\eta_a(x), \eta_b(x))$ is constructed and simplified as in [31] to obtain (2).

$$S_\beta(\eta_a(x), \eta_b(x)) = \eta_a(x) - \eta_b(x)\sigma[\beta(\eta_a(x) - \eta_b(x))] + \eta_b(x) \tag{2}$$

where σ is the sigmoid function. Substituting $\eta_a(x) = x$ and $\eta_b(x) = 0$, the approximate fitting of the ReLU function can be seen as $S_\beta(x, 0) = x\sigma(\beta x)$ which is same as swish function [25]. This function was defined as ACON-A function by the work of Wan *et al.* [26]. With this regard, it is alleged that additional maximum based activation function of the ReLU activation functions can also be transformed into the ACON family. For example, using PReLU function $f(x) = max(x, 0) + p.min(x, 0)$ where p is the learnable parameter with initial value of 0.25. If we considering two learnable parameters, p_1 and p_2 the ACON function is given as (3):

$$f_{ACON} = S_\beta(p_1x, p_2x) = (p_1 - p_2)x\sigma[\beta(p_1 - p_2)x] + p_2x \tag{3}$$

by strategically tuning hyperparameters to toggle between active and inactive states, this activation function effectively learns the linearity and non-linearity gate across the entire activation function range of a network. Therefore, the ACON activation function can be referred to as a combination of a sigmoid function and a linear function with learnable parameters. The sigmoid function scales the input x based on its statistics, while the linear function applies a shift and scaling to the output.

The parameters p_1, p_2 and beta, are learned during the training process, which allows the network to adaptively scale the input based on its statistics and improve its performance. Choosing ACON activation is a commendable decision for the following reasons; Firstly, it introduces an additional learnable parameter, *w*, that enables the network to adaptively scale the input based on its statistics. This helps to reduce the impact of outliers and improves the overall robustness of the network. Secondly, the ACON activation function allows the network to learn different scaling factors for positive and negative values of the input. This can be useful for tasks where the input has a skewed distribution or where the positive and negative values have different meanings. Thirdly, the ACON activation function can be easily integrated into existing CNN architectures without requiring major changes. It can be used as a drop-in replacement for existing activation functions, such as ReLU, PReLU, and Swish. Finally, the ACON activation function has been shown to improve the performance of CNNs on several benchmark datasets, including ImageNet, CIFAR-10, and CIFAR-100. It has been shown to outperform existing activation functions on tasks that require high model capacity or where the input has a skewed distribution.

2.5. Joint regression

Joint regression is the final stage of the lightweight hand pose estimation system aiming at estimating the location of each joint in a hand, in this case finger joints and a palm. The input for joint regressor as in is the finger feature map with tensor size $F_{mF} \in \mathbb{R}^{168\times6\times6}$ and the palm feature map with size $F_{mP} \in \mathbb{R}^{168\times6\times6}$, and the output is a refined feature map. This refined feature map is formed as a combination of palm and finger joints to obtain a single unified prediction of the entire hand joints as in [23]. For easy of processing, each feature map is firstly flattened from a $\mathbb{R}^{168\times6\times6}$ into $\mathbb{R}^{6048}$ before it is sent to fully connected layer with ACON activation to produce an output product of a vector size 640 which is same as in [11]. During experiments, we noticed that lowering the vector size to 640 has the same results as the original 1024. However small size has significantly reduced complexity and computation time. Further splitting of the feature map both for the finger joints and palm joints in the fully connected layers produces a palm joint prediction of size $\mathbb{R}^{63}$ and a finger joint prediction of size $\mathbb{R}^{15\times3}$ respectively. Therefore, joint regression stages leave the hand joints that are of vector size of 1280, 640 features from the palm and 640 from the finger joints respectively. The final predicted hand joints of size $\mathbb{R}^{21\times3}$ is finally passed to fully connected layer which locate all the 21 joints in place.

2.6. Transfer learning

Deep learning techniques, specifically CNNs, have an advantage over traditional methods in that they can be reused for similar tasks. We utilized a transfer learning approach to confirm the general usability of our model. Transfer training has a similar workflow to initial training, with the exception that the first four layers of the model are pre-trained, and the output layer is not. We uptake the advantage of transfer learning in deep learning to adopt some of the layers of MobileNet, drop others and apply to our presented HPE domain. To this end, the transfer learning strategy was applied to evaluate the general use representation of the trained model.

2.7. Experiments

Our experiments are based on the Tensorflow2.4.0 framework, cuda11.1, 11400F CPU and RTX3090 GPU. Our network was trained batch-wise using Adam optimizer with a learning rate of 10^{-3}, and a decay rate of 0.00001. During training, the best accuracy was recorded when the epoch size reached 256. We use NYU [35], imperial college vision lab (ICVL) [36] and Microsoft Research Asia (MSRA) [37] datasets for validation. Given its strong scalability and extensibility, we conducted experiment under the conditions specified by Du *et al.* [11], training with 256 epochs and evaluating our results in comparison to the latest advancement in the field. We demonstrate the effectiveness of our methods in ablation studies. On the first round, we train our network on the MSRA [37] dataset. The training results which include training error, and the finest epoch were recorded. The experimental architectures of our LightWeightNet architecture were compared to the closest method CrossInfoMobileNet and the results of the testing errors are depicted in Figure 4.

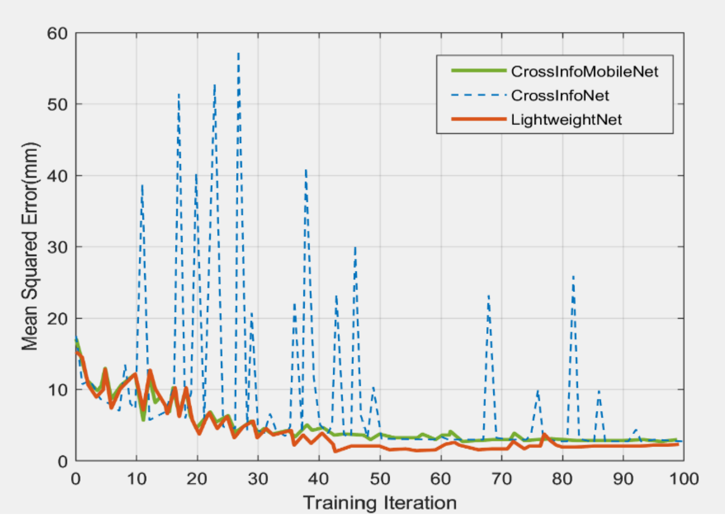

Figure 4. Mean squared error computed after each iteration during training

From Figure 4, it can be observed that the proposed LightweightNet is more stable during training process of the chosen subject of MSRA dataset compared to both CrossinfoNet and the CrossInfoMobileNet. This proves that the use of ACON produces improved output than ReLU and H-Swish as proposed by Du *et al.* [11]. The improved stability of the ACON is attributed to its unique feature of performing adaptive learning of which fixed activation such as ReLU and H-Swish cannot (best viewed in color).

2.7.1. Datasets

In this section, we present three datasets that were utilized in our experiments. These datasets have been widely used by recent researchers [11], [38]–[41], and as a result, we will conduct a comprehensive comparison with other works on these three datasets to assess the performance of our proposed system. The first dataset, NYU, is a publicly available dataset released by New York University [34]. It comprises 72,757 training sets and 8,252 testing sets of RGB-D images captured using the structured light-based sensor PrimeSense Carmine 1.09 from three different viewpoints. Each frame in this dataset is annotated with precise ground truth hand pose configurations and exhibits a wide range of pose variations. For our experiment, we solely utilize the depth data from a single camera. In contrast, the Imperial College Vision Lab (ICVL) dataset, provided by the ICVL, was captured using an Intel Creative Interactive Gesture Camera. This publicly available dataset [35] comprises approximately 180,000 depth frames in the training set, featuring diverse hand poses recorded from 10 different subjects. The test set consists of two sequences, each containing around 700 frames. Each hand pose in the dataset is annotated with 16 joint coordinates. The depth images in this dataset are of high quality, with few or no missing depth values, and exhibit sharp outlines with low noise levels. Despite a recent report highlighting limited pose variability and potential annotation inaccuracies [41], the ICVL dataset remains highly suitable and optimal for hand pose estimation systems. Microsoft Research Asia (MSRA) dataset [37], consists of approximately 76,000 depth frames captured using a time-of-flight camera. This dataset includes sequences from 9 subjects, with each subject dataset containing 17 gestures. Notably, the MSRA dataset contains data from 9 subjects obtained from a different source, and it excels particularly in the evaluation of finger joints.

2.7.2. Evaluation metrics

To assess the accuracy of the proposed system's estimation results, we employ two distinct evaluation criteria. The first metric measures the percentage of successful frames, while the second metric evaluates the mean error for each joint of the entire hand. There criteria have been used in recent many Hand pose estimation approaches:

- The fraction of the sample error distance within a specified threshold. This metric calculates the percentage of successful frames where the error distance for each joint falls below the defined threshold. It is important to note that this criterion may be more susceptible to ambiguity, as a single incorrect joint estimation can influence the overall evaluation of the hand pose.
- Mean error distance of various joints, along with their corresponding average. This criterion is commonly used in the literature on hand pose estimation [40]–[42] due to its simplicity in calculating joint errors. It provides valuable insights into the overall accuracy of the system by measuring the average error distance across different joints.

2.7.3. Performance analysis

The performance analysis of the proposed systems is presented through comparison to closely related state-of-the art architectures with their different variants in Table 1. The results are from the trained MSRA datasets, and it includes the following items; average mean joint error (*Error*), number of neurons (*Neurons*), number of architecture parameters (*Parameters*). As shown in Table 1, the first row displays the results obtained from CrossInfoNet, while the subsequent rows present the results from CrossInfoMobileNet. These rows include the plain architecture as well as variations with dropout (d) implemented in the joint regressor of their fully connected layers. In the last row, our architecture, LightWeightNet, is presented. From the table, it can be inferred that LightWeightNet is half the size of CrossInfoNet and 10% smaller than CrossInfoMobileNet. Despite this size reduction, LightWeightNet achieves the same level of accuracy as its predecessors on the MSRA dataset. For a more comprehensive comparison of our method with other state-of-the-art techniques, please refer to subsection 2.8.

Table 1. Performance overview of contending architectures and their variants

Architecture	Neurons	Errors (mm)	Parameters	Size (MB)
CrossInfoNet	1024	8.19	23,936,648	91.48
CrossInfoMobileNet	640	8.19	10,768,376	41.31
CrossInfoMobileNet-d0.4	640	8.48	10,768,376	41.31
CrossInfoMobileNet-1024	1024	8.19	16,765,304	64.22
CrossInfoMobileNet-512	512	8.25	8,900,472	34.18
CrossInfoMobileNet-128	128	8.41	3,689,976	14.30
CrossInfoMobileNet-64	64	8.52	2,878,904	11.22
LightWeightNet	640	8.17	8,880,801	36.04

2.8. Quantitative analysis

Qualitatively, our approach's performance is assessed on three publicly available datasets specifically designed for hand pose estimation. To evaluate our method, we primarily consider recently published works that are closely related to our research. For evaluation, we employ well-established evaluation metrics discussed in subsection 2.7.2. These metrics have been widely used in the literature, including in works such as [36], [37], and [42] to evaluate hand pose estimation. The values reported in the respective papers or measured from the accompanying graphs, if available, are presented in Table 2.

Table 2. Comparison of mean error distance for the state-of-art methods on three datasets

Algorithm	ICVL	NYU	MSRA
HandPointNet [22]	6.9	10.5	8.5
V2V-PoseNet [42]	6.28	8.42	7.59
CrossInfoMobileNet [11]	7.33	12.71	8.19
SHPR-Net [6]	7.22	10.78	7.76
LightWeightNet	7.02	10.07	6.9

2.8.1. Error analysis on NYU datasets

Although the test results for CrossInfoMobileNet in NYU were not reported, we experimented to see its performance and compare it with the presented LightWeightNet. The test results are presented in Figure 5. We opted to only compare the two baselines because of the similarity in our approaches. On the first few samples both our architecture and CrossInfoMobileNet [12] performed well with insignificant differences, however as the number of samples increased, our proposed method exhibited superior performance compared to the competing state-of-the-art architectures. For instance, when evaluating the test samples based on the maximum joint error below a specified threshold, our method achieved a remarkable 96% accuracy at a threshold level of 40 mm. In the Figure 5, the fraction of test samples whose distance between all estimated joints and ground truth is below a given threshold are presented (best viewed in color).

.

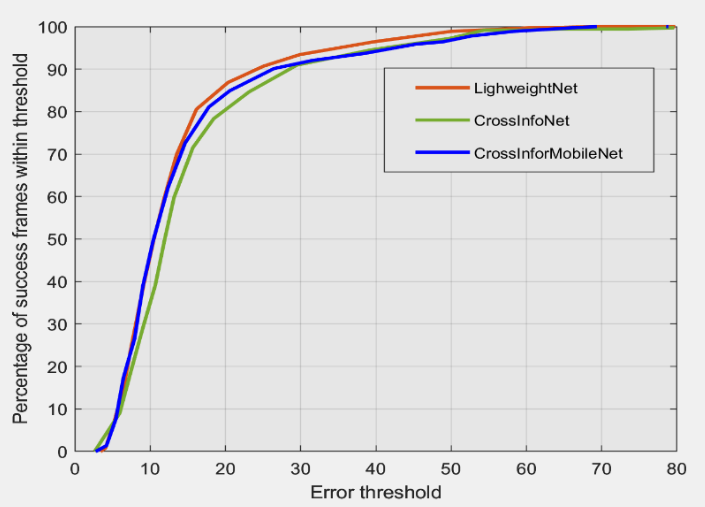

Figure 5. Comparison of the state-of-the-art methods on NYU dataset

2.8.2. Error analysis on ICVL

Similarly, on ICVL, again we request our readers to be aware that, CrossInfoMobileNet [11] did not provide the test results on ICVL, but we utilize the shared model to test its performance in ICVL so we test our hypothesis. The test results for all the three competing literatures are presented in Figure 6. Generally, and as in NYU, our proposed system outperformed both contending architectures especially when the number of samples are few.

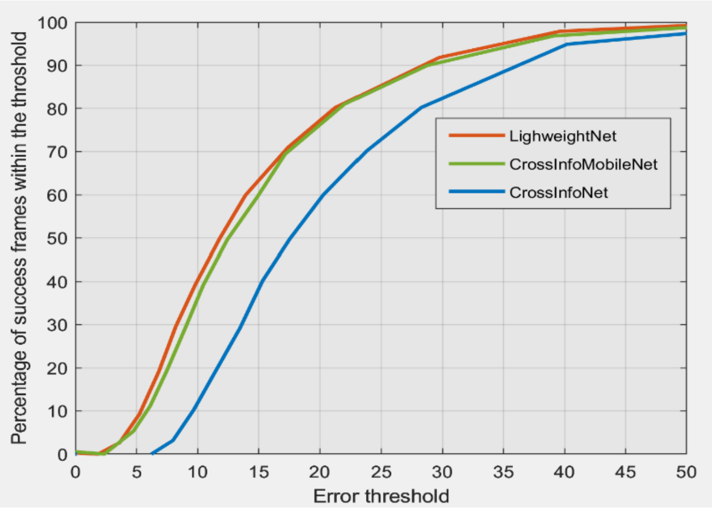

Figure 6. Comparison of the state-of-the-art methods on ICVL dataset (best viewed in color)

2.8.3. Error analysis on MSRA

Figure 7 demonstrates the mean joint error for each joint on MSRA dataset. We request our reader to note that testing was conducted only for the first subject as in [11] so that we can present a fair result. Our method LightWeightNet outperforms both other methods on the plotted metric indicating that it is lighter and more suitable for mobile devices. For example, when the error threshold is 20mm, the proportions of good frames for our method is about 98% better than CrossInfoMobileNet [11] which attains 96%, and CrossInfoNet which attain 94%. The closeness of performance between our method and CrossInfoMobileNet [11] is because of the optimization and training methods both approaches considered.

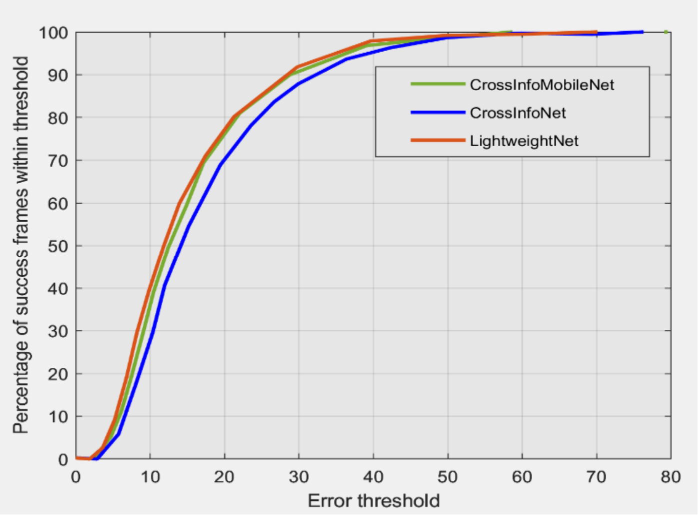

Figure 7. Comparison of the state-of-the-art methods on MSRA dataset (best viewed in color)

3. CONCLUSION

We presented our optimal approach for a hand pose estimation system suitable for running on a mobile device application. We showed that with a smaller number of parameters, over 38.6% less, we can still attain similar or better accuracy than the non-optimal methods in the literature. The development of lightweight hand pose estimation models for mobile devices has significant potential for various applications such as mobile gaming, augmented reality, virtual reality, and more importantly sign language recognition. With the increasing demand for more sophisticated human-computer interfaces, the ability to detect hand poses accurately and efficiently on mobile devices is becoming more critical. The results presented in this paper demonstrate the feasibility of using lightweight deep-learning models for real-time hand pose estimation on mobile devices. Such models can achieve high accuracy while requiring relatively low computational resources, making them suitable for deployment on mobile devices with limited processing power. Moving forward, further research can explore the potential for integrating these models with other mobile applications and evaluating their performance in different real-world scenarios. Achieving precise regression of finger joints offers crucial cues for accurate joint regression, resulting in reduced errors and improved accuracy in hand estimation. With this information future research should include large-scale benchmark datasets that can enable data-driven model development and foster advancements in the hand pose estimation domain, explore further optimization techniques such as compression methods, and quantization techniques, which will aid in reducing model size while maintaining or improving accuracy. By pursuing these directions, we can further advance the HPE domain and enhance the overall user experience in various human-computer interactions, enabling its integration into a wide range of practical applications.

REFERENCES

[1] P.-A. Blanche, “Holography, and the future of 3D display,” *Light: Advanced Manufacturing*, vol. 2, no. 4, 2021, doi: 10.37188/lam.2021.028.
[2] A. Sinha, C. Choi, and K. Ramani, “DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 4150–4158, doi: 10.1109/CVPR.2016.450.
[3] L. Ge, H. Liang, J. Yuan, and D. Thalmann, “Robust 3D hand pose estimation in single depth images: From single-view CNN to

multi-view CNNs," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 3593–3601, doi: 10.1109/CVPR.2016.391.

[4] T. Y. Chen, M. Y. Wu, Y. H. Hsieh, and L. C. Fu, "Deep learning for integrated hand detection and pose estimation," in *Proceedings - International Conference on Pattern Recognition*, 2016, vol. 0, pp. 615–620, doi: 10.1109/ICPR.2016.7899702.

[5] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3D pose inference from synthetic images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4663–4672, doi: 10.1109/CVPR.2018.00490.

[6] X. Chen, G. Wang, C. Zhang, T. K. Kim, and X. Ji, "SHPR-Net: Deep semantic hand pose regression from point clouds," *IEEE Access*, vol. 6, pp. 43425–43439, 2018, doi: 10.1109/ACCESS.2018.2863540.

[7] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Proceedings - International Conference on Image Processing, ICIP*, 2018, pp. 4512–4516, doi: 10.1109/ICIP.2017.8297136.

[8] M. Rezaei, R. Rastgoo, and V. Athitsos, "TriHorn-Net: A model for accurate depth-based 3D hand pose estimation," *Expert Systems with Applications*, vol. 223, Aug. 2023, doi: 10.1016/j.eswa.2023.119922.

[9] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," *arXiv preprint arXiv:1502.06807*, 2015.

[10] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3316–3324, doi: 10.1109/ICCV.2015.379.

[11] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9896–9905.

[12] M. Šimoník and M. Krumnikl, "Optimized hand pose estimation CrossInfoNet-based architecture for embedded devices," *Machine Vision and Applications*, vol. 33, no. 5, Sep. 2022, doi: 10.1007/s00138-022-01332-8.

[13] A. Howard *et al.*, "Searching for mobileNetV3," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[15] F. Xiong *et al.*, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 793–802, doi: 10.1109/ICCV.2019.00088.

[16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *arXiv preprint arXiv:1602.07360*, 2016.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM 60*, no. 6, pp. 84–90, 2017.

[18] M. Sandler, M. Zhu, A. Zhmoginov, and C. V Mar, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[19] Y. Liu, J. Jiang, and J. Sun, "InterNet +: A light network for hand pose estimation," *Sensors*, vol. 21, no. 20, 2021.

[20] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression PointNet for 3D hand pose estimation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11217 LNCS, pp. 489–505, 2018, doi: 10.1007/978-3-030-01261-8_29.

[21] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

[22] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet : 3D hand pose estimation using point sets supplementary material," *Cvpr*, pp. 3–5, 2018.

[23] B. Khasoggi, Ermatita, and Samsuryadi, "Efficient MobileNet architecture as image recognition on mobile and embedded devices," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 16, no. 1, pp. 389–394, Oct. 2019, doi: 10.11591/ijeecs.v16.i1.pp389-394.

[24] R. Mourad, C. Sinoquet, N. L. Zhang, T. Liu, and P. Leray, "A survey on latent tree models and applications," *Journal of Artificial Intelligence Research*, vol. 47, no. February, pp. 157–203, 2013, doi: 10.1613/jair.3879.

[25] L. Huang, J. Tan, J. Liu, and J. Yuan, "Hand-Transformer: Non-autoregressive structured modeling for 3D hand pose estimation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12370 LNCS, pp. 17–33, 2020, doi: 10.1007/978-3-030-58595-2_2.

[26] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3D regression for hand pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5147–5156, doi: 10.1109/CVPR.2018.00540.

[27] H. Qian, J. Xu, and J. Zhou, "Object detection using deep convolutional neural networks," in *Proceedings 2018 Chinese Automation Congress, CAC 2018*, 2019, pp. 1151–1156, doi: 10.1109/CAC.2018.8623808.

[28] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit, "Keypoint transformer: solving joint identification in challenging hands and object interactions for accurate 3D pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022, vol. 2022-June, no. 1, pp. 11080–11090, doi: 10.1109/CVPR52688.2022.01081.

[29] F. Guo, Z. He, S. Zhang, X. Zhao, and J. Tan, "Attention-based pose sequence machine for 3D hand pose estimation," *IEEE Access*, vol. 8, pp. 18258–18269, 2020, doi: 10.1109/ACCESS.2020.2968361.

[30] T. Hu, W. Wang, and T. Lu, *Hand pose estimation with attention-and-sequence network*, vol. 11164 LNCS. Springer International Publishing, 2018.

[31] M. Kocabas, C. H. P. Huang, O. Hilliges, and M. J. Black, "PARE: Part attention regressor for 3D human body estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 11107–11117, doi: 10.1109/ICCV48922.2021.01094.

[32] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[33] N. Ma, X. Zhang, M. Liu, and J. Sun, "Activate or not: Learning customized activation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8032–8042.

[34] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics*, vol. 33, no. 5, 2014, doi: 10.1145/2629500.

[35] D. Tang, H. J. Chang, A. Tejani, and T. K. Kim, "Latent regression forest: Structured estimation of 3D hand poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1374–1387, 2017, doi: 10.1109/TPAMI.2016.2599170.

[36] R. Du *et al.*, “Opportunistic interfaces for augmented reality: transforming everyday objects into tangible 6DoF interfaces using Ad hoc UI,” in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, Apr. 2022, pp. 1–4, doi: 10.1145/3491101.3519911.

[37] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, “Cascaded hand pose regression,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June, pp. 824–832, doi: 10.1109/CVPR.2015.7298683.

[38] Y. Wang, L. Chen, J. Li, and X. Zhang, “HandGCNFormer: A novel topology-aware transformer network for 3D hand pose estimation,” in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2023, pp. 5664–5673, doi: 10.1109/WACV56688.2023.00563.

[39] J. F. Banzi, Z. Ye, and I. Bulugu, “A novel hand pose estimation using dicriminative deep model and Transductive learning approach for occlusion handling and reduced descrepancy,” in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, Oct. 2016, pp. 347–352, doi: 10.1109/CompComm.2016.7924721.

[40] M. Oberweger and V. Lepetit, “DeepPrior++: Improving fast and accurate 3D hand pose estimation,” in *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, 2017, vol. 2018-Janua, no. October, pp. 585–594, doi: 10.1109/ICCVW.2017.75.

[41] J. Banzi, I. Bulugu, and Z. Ye, “Deep predictive neural network: Unsupervised learning for hand pose estimation,” *International Journal of Machine Learning and Computing*, vol. 9, no. 4, pp. 432–439, Aug. 2019, doi: 10.18178/ijmlc.2019.9.4.822.

[42] G. Moon and K. M. Lee, “V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map,” in *Proceedings of the IEEE conference on computer vision and pattern Recognition*, 2019, pp. 5079–5088.

BIOGRAPHIES OF AUTHORS

Jamal Firmat Banzi is a lecturer of artificial intelligence at the Sokoine University of Agriculture. Dr. Banzi obtained his BSc. degree in information systems from the University of Dodoma, Tanzania in 2011. He then went for a master’s study and obtained an MEng. signal and information processing engineering, at Tianjin University of Technology and Education in 2015. In the year 2019, he obtained a Ph.D. in information and communication engineering majoring in artificial intelligence from the University of Science and Technology of China (USTC). Dr. Banzi’s current research interests include cognitively inspired AI, AI+ healthcare, computer vision, deep learning, computer assisted sign language recognition for hearing-disabled communication, and seamless human machine interaction. He can be contacted at email: jamalbanzi@sua.ac.tz.

Stanley Leonard is currently a lecturer of computer science and engineering at Mbeya University of Science and Technology (MUST) in Tanzania. He received his bachelor engineering in computer from the Dar es Salaam Institute of Technology (DIT) in Tanzania in the year 2009. He then went for a master’s degree at the University of Dodoma (UDOM) in Tanzania in 2012 and obtained a master’s degree in telecommunications engineering. He is currently a lecturer and Ph.D. candidate researcher in the Department of Computer Science and Engineering at Mbeya University of Science and Technology (MUST), Tanzania. His current research interests include artificial intelligence and machine learning techniques for human-computer interaction application domains. He can be contacted at msuya@must.ac.tz.