A Deep Regression Network with Key-joints Localization for Accurate Hand Pose Estimation

Stanley L Tito, Jamal F. Banzi, and Aloys N. Mvuma

***Abstract*—In this paper, a new method is proposed to improve hand joint regression in 3D hand pose estimation. The existing methods regress all joints together given a depth map. This causes misallocations of some hand joints, misuse of hand depth information, and have difficulties in estimating 3D coordinates accurately. In this paper, joint regression is performed in stages, such that highly flexible joints e.g. fingertip joints are regressed first followed by less flexible joints to avoid getting some errors while estimating all joints together. In practice, fingertip joints constitute relatively higher estimation errors than all other joints. Thus, we perform fingertip joint localization (2D joint estimation) after obtaining rough pose estimates from the pose estimator to locate fingertip joint positions. We then use these 2D joint estimates to generate the depth coordinates of the pose estimator. To further ensure the accuracy of the absolute pose hypothesis, we integrate a robust implicit shape-based hand detector with the deep regression pose estimator into one pipeline through a shared convolutional layer. Finally, a shared convolutional layer converts the 2D joint location to 3D poses. Consequently, our system can accurately estimate hand pose based on the prior knowledge of a well detected human hand and the properly located joint positions. Experiments were carried out on three publicly available datasets, ICVL, NYU, and MSRA. The proposed hand pose estimation system attains an accuracy of 96.4% at the threshold level of 40mm on the ICVL dataset, 92% on MSRA, and 89% on the NYU dataset illustrating the effectiveness of the proposed system over many state-of-art approaches.**

***Index Terms*—Deep learning, human-computer interaction, image analysis.**

I. Introduction

There is a growing interest in the hand pose estimation approach that can facilitate collaborative computing. The approach should improve interaction between humans and the cyber space. Various human-computer interaction applications including gaming, virtual reality, augmented reality, autonomous driving, automatic sign-language recognition, doctor's remote surgery and many others, rely upon the estimation of hand joint locations—i.e., hand pose estimation (HPE) from input visual data.

Indeed, there has been a large body of works [1]-[7] devoted to developing hand pose estimation system, thanks to the advance of depth sensors, which were developed for body pose estimation. Depth sensors improve robustness and reduce sensitivity to variations of illumination. Additionally, depth sensors can provide adequate 3D information of a hand geometry, which simplifies hand detection process, and therefore enhances the stability of hand pose estimation systems.

Despite the clear interest in hand pose estimation, it is still difficult to estimate human hands due to challenges faced during the estimation process. For example, estimating a human hand pose requires the location of the hand to be recognized and detected from an input image. During the hand detection process, a position of the hand has to be detected from the complex background of the image, which adds a series of problems such as viewpoints variations and high intra-class variation [8]. Similarly, the defective hand detectors cause failure in the subsequent hand pose estimation stage. In addressing this problem, early works only make assumptions that a human hand will just be an object appearing nearest from the camera in the image [9]-[11]. However, HPE systems made under such assumptions usually fail in many conditions such as, when multiple hands are being used and when a user is in a complex environment. It is therefore important to design a robust hand detector and integrate it with the hand pose estimation system to operate in all working conditions.

Similarly, the subsequent pose estimation stage is also challenging due to the following reasons. Firstly, the noise due to input depth image will certainly mislead the pose estimator and distort the final output results. Secondly, the flexibility of a human hand is relatively high, for example, a human hand may have up to 29 degrees of freedom (DoF) i.e. the variation of the finger joint flexibility which may compromise the accuracy of the hand pose estimation [12]. Specifically, joints in the palms such as wrist and finger roots have a lower DOF than the joints in the fingers i.e. fingertip joints [13]. This flexibility of the fingertip joints compromises the accuracy of the final estimated pose in many of the previous bodies of works [8], [9], [12], [14], [15].

Inspired by the fingers flexibility, this study proposes a HPE system that addresses the challenge of the high flexibility (high DoF) due to fingertip joints which causes poor results in the final pose estimates.

In reality, fingers flexibility, and poor depth quality around fingertips are improperly increasing the errors due to fingertip location. Nevertheless, many important human-computer interaction actions such as clicking, zooming, and scrolling depend heavily on the movement of fingertips. Considering that, this paper introduces a three-stage HPE system. In the first stage, a robust implicit shape-based hand detector is designed to detect the position of the hand using local features votes. Secondly, our model performs finger joint localization based on the root-center-angle algorithm to acquire the relative 2D

Manuscript received May 26, 2022; revised August 22, 2022.

Stanley L Tito and Aloys N. Mvuma are with Mbeya University of Science and Technology, Mbeya, Tanzania (e-mail: stanley.tito@must.ac.tz, anmvuma@gmail.com).

Jamal F. Banzi is with the Sokoine University of Agriculture, Morogoro, Tanzania (e-mail: jamalbanzi@sua.ac.tz).

doi: 10.18178/ijmlc.2022.12.6.1118

information of finger joint positions, while measuring joint separation through the root-center angle (RCA). Lastly, to improve the performance of the proposed system, we train a stable deep regression network containing a shared convolutional layer with the hand detector in one pipeline. We construct fast and deep neural network architecture for both hand detection and pose estimation such that the proposed system can be computationally efficient and with improved performance for a real-time application. Fig. 1 presents the general overview of the proposed hand pose estimation system. The contributions of the proposed approach are summarized as follows:

- The integration of the hand detector in line with the pose estimator to ensure the estimated pose is based on the prior knowledge of the hand rather than just assuming the nearest object to the camera to be the hand.
- To avoid errors of estimation due to finger's flexibility, this paper proposes a root center angle algorithm that performs finger joint localization after obtaining rough pose estimates. Localizing finger joints confers precise joint position of the final pose estimates.
- Lastly, to make full use of hand depth information, we introduce a new in-stage joint regression method, whereby highly vulnerable joints such as extended fingertip joints are firstly regressed, followed by other joints in their order of flexibility. This allows 2D information to concatenate with 3D depth co-ordinate for precise joint positioning.

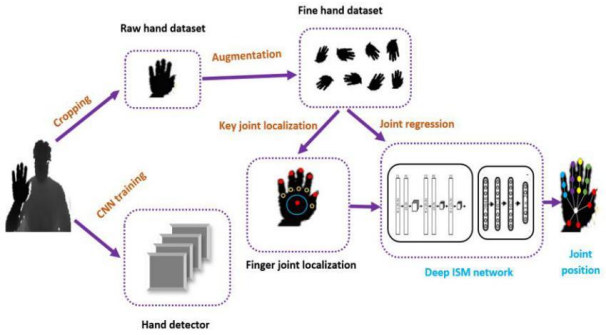

Fig. 1. Illustration of our hand pose estimation system showing both hand detector and pose estimator.

The rest of the paper is organized as follows: Related works are in section II. Section III presents the system overview. Sections IV and V describe the hand and fingers detection model and pose estimation. Section VI explains the detailed deep learning architecture for hand pose estimation. Section VII describes the setup of the experiment and results. Finally, section VIII concludes the work.

II. Related Works

This section highlights the recent methods and observes their significance to the proposed approach. For a comprehensive survey, we refer the reader to [1], and [2]. Generally, hand pose estimation approaches are categorized into two complementary paradigms: a generative approach and discriminative approach. Other approaches combine both discriminative and generative techniques to form hybrid methods.

Generative approach– The majority of early works were based on generative approaches whereby numerous gestures are developed and matched with the best suit depth image. These approaches faced tremendous overhead in rendering candidate poses which consequently breaks the optimization process. Martin *et al* [11] tried to address this challenge by recovering 3D hand poses from a monocular image using a generative Bayesian method. By synthesizing the corresponding hand silhouette Martin *et al* [11] project the image plane to measure its corresponding likelihood in a given model to extract background and hand skin pixels. The final hand pose is then refined iteratively through the minimization of negative log-likelihood. Latterly, Melax *et al* [10] used a rigid body simulation for optimization and then applied point to surface constraints that work similarly to ICP in an optimization process.

All of these generative-based optimization methods fail due to the local minima and break optimization process. This explains why discriminative methods seem to be ideal for human hand pose estimation.

Discriminative methods– majority of the recent works [12], [14], [15] are based on machine learning. These methods use deep convolutional neural networks for estimation and regression of joint positions. The work of Guo *et al* [16] recovered a 3D hand pose using a tree-structured Region Ensemble Network (REN) which divided the network outputs into regions and integrated output results from multiple regressors on each region. Latterly, the work of Madadi *et al* [17] use a hierarchical tree-like structure Convolutional Neural Network (CNN) that predicts diverse parts of the kinematic tree through which branches are trained to obtain the local poses as a subset of hand joints. However, the designed CNN architecture would vary depending on data annotation.

This shortcoming could be solved by the work of Forure *et al,* [18] which leverages alternative annotations methods from different datasets by adding a shared representation that improved predictive accuracy. Deng *et al* [19] introduced data representation of the input depth whereby the depth image is converted to 3D volume and 3D CNN is used to predict joint location. Nevertheless, this method has a shortcoming of low computational efficiency and increased ambiguity. Afterwards, Wan *et al* [20] use surface normal instead of the depth image which increases an additional overhead since surface normal are not easily accessible by the present depth sensors. Other approaches [21], [22] propose to estimate 2D heatmaps separately for different joints instead of estimating 3D joint locations directly. However, most of discriminative methods require complex post-processing to suit a kinematic model to the heat map.

Hybrid methods–Recently most hand pose estimation approaches [23]-[25] combine both discriminative and generative techniques to provide a smooth but robust hand pose estimation system. Hybrid methods combine discriminative methods and generative methods by first generating candidate poses using discriminative methods then utilize them as an initial state of the generative approaches to optimize the full hand poses. An early hybrid method was presented by Sridhar *et al* [26] whereby a person-specific model for body pose estimation was developed. The model was built for a hybrid system using five RGB cameras and a Time-of-Flight (ToF) camera. Latterly, Qian *et al* [14] applied hybrid method for real-time tracking of the hand using a simple hand model containing numerous spheres and then combine a gradient-based

discriminative approach with the stochastic optimization.

Then, Sharp *et al* [15] leverage a hand mesh into triangles and vertices to formulate multiple discriminative patch levels followed by a generative refinement of the final pose.

III. System Overview

The proposed approach aims to address the challenges facing hand pose estimation process as discussed in Section I. In doing so, a reliable and robust hand detector based on implicit shape model (ISM model) is proposed to detect a human hand.

A. ISM for Hand Detection

The ISM was firstly proposed by Leibe *et al* [26] to detect instant arbitrary object classes. It has been effectively applied to body pose estimation by Jurgen Muller *et al* [27]. The core idea in the implicit shape model is to represent the structure of an object by the classical distribution of its corresponding classes (elements) relative to the object center, represented as:

ISM (*C*, *O*) = (*C*, *P*), where *C* is a codebook containing feature vectors of the object, in our case joint locations that appends on the human hand. *O* is the object, in our case the hand and P is a probability distribution that confines the hand center and it specifically appears relative to the joint locations. The probability distribution *P* signifies the probability over the locations of the hand center for a given joint independently of all other joints. Having obtained joint location from the codebook, one can vote for the hand center to be at the location h relative to the position of the joint location v based on conditional probability $P(h|v)$. *P* is independent of all other joint locations, and $v = 1,\ldots j$ where *j* is the number of joints. Using ISM for hand detection is a good choice because of the following reasons:

- It ensures a scale-invariant, interest joint detectors and descriptors do not only have a position but a scale.
- It can withstand cases where fingers are partly occluded.
- ISM avoids modeling a mutual relationship between joint location and the center of the hand by a joint probability because this would need large numbers of training examples. Alternatively, it integrates hints from individual joint location by a voting scheme.

B. Generating Hand Codebook

The codebook C for hand features is generated to learn a set of joints showing instances of hand joint positions. For each of these joints, the absolute scale-invariant key points are computed and for each key point region, the feature vector $\boldsymbol{p(a)}$ is extracted to encode the corresponding joint information. Then, the feature vectors combined with absolute scale-invariant key points are clustered resulting in feature vector centroid, which is called visual words as shown in Fig. 2.

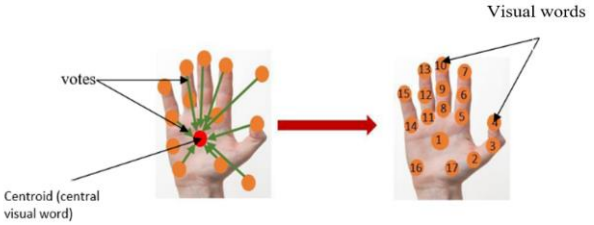

Fig. 2. ISM feature votes for a central hand location.

C. Vote Weighting

We now match hand joints against visual words of the codebook and use the relative position and scale information to cast votes for the center of the hand relative to the detected joints. For a vote generation, each joint of an observation hand, $O=(\Delta x,\Delta y,z)_w$ is associated with the visual word that matches the joint. A vote is generated where the location and scale of the joint corresponds to the center of the hand but it is adapted to the joint scale j. Considering the 3D voting space, a joint feature $J=(j_x, j_s, j_z)$ that matches to a visual word w casts a vote $V=(v_r, v_x, v_y, v_z)$ based on the offset vectors $O=(\Delta x, \Delta y, z) \in \theta_w$. Hence

$$V_r = \frac{1}{|\theta_w|} P(w|\mathrm{d}) \tag{1}$$

$$V_x = j_x + \Delta x \frac{j_z}{o_z} \tag{2}$$

$$V_y = j_y + \Delta y \frac{j_z}{o_z} \tag{3}$$

$$V_z = \frac{j_z}{o_z} \tag{4}$$

The value of the v_r determines the influence of the single vote assigned to each joint during normalization. Some other research methods [27], [16] assign the number of offset vectors for one joint feature $\mathbf{w_1}$ which would result into significant variation when compared to the offset vectors collected for another joint feature $\mathbf{w_2}$. We dismiss this method because it could bring bias, as it considers a hand as one object while in our context we refer to a hand as a set of objects. Therefore, for perfect hand detection, we present a soft matching technique whereby the descriptor vector **d** of a detected hand (W) is matched to a set of joints w such that $W=\{w_1,\ldots,w_M\}$ and the Euclidean distance between the descriptor vector **d** and the word centroid vector **c** is below the threshold.

$$W=\{w : \| c-d \|_2 \le \Theta, i \in \{1,\ldots\ldots | C |\}\} \tag{5}$$

To compensate for the different number of matching feature word *M*, we define the vote weight v_r as:

$$v_r = \frac{1}{|\theta_w|}\frac{1}{M} \tag{6}$$

Furthermore, to find the probability that the descriptor vector **d** matches the set of joints *w*, we define the weighted vote based on probability distribution $P(w|\mathrm{d})$ as

$$v_r = \frac{1}{|\theta_w|} p(w|\mathrm{d}) \tag{7}$$

Therefore, the voting mass of 1 has been distributed to all possible interpretation $W=\{w_1,\ldots,w_M\}$ of the descriptor vector **d**.

D. Hand Detection

For hand detection, the locations of high vote density in the 3D space are identified. This could be the center of the hand.

Having recognized center of the hand, we compute the sum of the vote weight (features) along each voting joint and transfer these features to the detection network. Fig. 3 shows the structure of the hand detector.

The number of output values from the code generator network is 2W plus 4W where W indicates the number of "code words". The 2 value represents the probability of a feature being a joint or not joint and 4 means their spatial positions (i.e. x, y, z, r). The number of joints with higher votes are forwarded to the detection network by a region of interest (RoI) pooling, which is a max-pooling layer that produces small and fixed sized output in accordance with the feature map of the desired RoI.

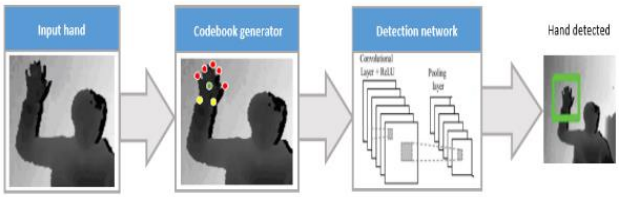

Fig. 3. The structure of the hand detector consisting of codebook generator and the detection network.

A codebook contains a number of activated codewords which are casting votes towards the center of the hand.

Therefore, to match with the previously learned probability distribution P_i, we cast votes at different locations (x, y, z) and scales. Locations in this 3D voting space with a large number of assigned votes are considered as detected instances of the hand.

$$p(a) = \frac{1}{W_b} \sum_{k=1}^{|v|} v_r^k K\left(\frac{\| v_{3-a}^k \|^2}{b} \right) \tag{8}$$

$$W = \sum_{k=1}^{|v|} v_r^k \tag{9}$$

where v is the set of all votes $v^k = (v_r^k, v_x^k, v_y^k, v_z^k)$ and $(k = 1, \ldots |v|)$ cast from all features for the RoI. | v | is the number of total votes cast, $v_3^k = (v_x, v_y, v_z)$ is the 3D vote space location of a vote v^k, W is the sum of weights of all votes cast, and b is the bandwidth (smoothing parameter) of a kernel while K indicates the kernel function. The kernel used is essentially a Gaussian kernel given as:

$$p(a) = \frac{1}{W_{b(s)}} \sum_{k=1}^{|v|} v_r^k K\left(\frac{\| v_{r-a}^k \|_2}{b(s)} \right) \tag{10}$$

Where the vote space locations $a=(x, y, z)$ and the vote density $p(a)$ is above some minimum threshold θ. We consider $a' = (x, y)$ as a detected instance of the hand at an image position (x, y) and define hypothesis score $s(a') = p(a)$ for this detection.

To this end, we have detected the location of the hand. The next step is to crop a hand image and obtain a fixed size which will be used as an input of the pose estimator network. We carefully extract a 3D cube as in [8], [28] to capture the real coordinates around a hand, while ensuring the formed image attains scale-invariance and background subtraction. This is more significant because the cube is extracted with reference to the hand center. The proposed hand detector is therefore very robust and reliable since it extracts the hand from a 3D cube instead of directly resizing the image which results in scale-invariance and background subtraction.

Moreover, the proposed hand detector is implemented using a deep neural network to capture high level functions of the input image which ultimately improves detection accuracy. The performance of the proposed hand detector is demonstrated in the experiment section.

IV. Hands Key Joint Localization

In this section, we explain how to find points of the human hand which are local extrema. Specifically, we need to identify hand points that are the best representation of the region of interests i.e. the neighborhood of the points in different scales. To locate these key points, we need to iterate over each pixel and compare it with all its neighbors. We introduced a novel technique for this task RCA, which is the extension of the [29] algorithm. Using this technique important key points can be recognized and finally estimated with a degree of assurance. The following subsections explain the process in details.

A. Fingertip Localization

Fingertip is the point that defines the end of the finger, with a high degree of flexion. For a given image depth, in Fig. 4(a), to locate the fingertip we first measure the tilt angle of the arm using principal component analysis and then make an arms orientation to cope with the rotation variation.

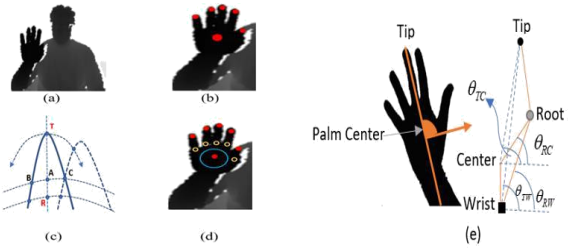

Fig. 4. Schematic plot showing the location of finger root and fingertip.

Hand image (b) The localized fingertips (c) The schematic plot of finger root location (d) location of finger root (e) palm center location

The palm center is regarded as the point with the maximum distance to the closest palm boundary. In this paper, we use distance transform to locate the palm center, obtain the wrist line, and calculate the palm radius (which is the minimal distance from the palm center to the boundary) as shown in Fig. 4(c).

Fingertip localization starts with scanning of the palm boundary using the distance to palm center and then calculate the curvature value of the points with the local maximum distance, which can be recognized as the fingertip in Fig. 4(b). By forming the global and local constraints, the noisy extreme points can be eliminated and only the stable ones can be attained. Then, the curvature values of the acquired stable points are applied for verification. Thus, the individual points with the maximum distances to the palm center and suitably large curvature are recognized as the fingertips.

B. Finger Roots Localization

For effective fingertip location, the corresponding finger root needs to be located. The localization of the finger root follows the following steps:

Firstly, we scan along the palm boundary both to left and right sides of every fingertip T as shown in Fig. 4(c), to obtain point AB whose distance to palm center is the local minimum value or less than a targeted threshold value. Secondly, we locate the midpoint A which is also referred to as the centerline of the finger. Using coordinates of the fingertip and that of the midpoint, the orientation of each finger can be calculated. Then the fingertip and the midpoints are connected using a line segment TA. Finally, we extended the line until it intersects the circle at the point R, which is referred to as a finger root.

C. Finger's Separations

Once the finger root is successfully localized as shown in Fig. 4(d), it is now important to obtain angle information to categorize each finger. Fingers separation are performed using four angle representations i.e. root-center-angle, tip-center-angle, tip-root-angle, and root-wrist-angle as presented in Fig. 4(e). These angles have different ranges, such that wrist to fingertip or wrist to root has many small angles than other angles with respect to palm center. This leads to a weaker distinction between fingers. The root-center-angle is less sensitive to the particular movement of the fingers than the tip-center-angle. It is therefore considered that a root-center-angle is a good choice to discriminatively identify each stretched finger because it is invariant to palm orientation. Also, the angle is less sensitive to the particular movement and therefore, will not be affected by changes in viewpoints.

Given N number of fingers, the vector for the root-center angle is given as $\theta = [\theta_1, \theta_2, \theta_3 ... \theta_N]^T$ (11)

The difference between every two root-center angles can sufficiently provide significant information and facilitate fingertip locations. Thus, to every *A* absolute angle, there are C_A^2 relative angles (the distinction of two root-center-angles) of which the number of non-zero items is

$$N_A = A + C_A^2 = \frac{A(A+A)}{2} \quad (12)$$

These feature vectors can be expanded to the fixed size by adding zero items. Nevertheless, root-center-angle cannot solely provide a sample distinction to the fingers. This is because the root-center-angle of the two fingers can be equal and hence can cause misclassification. Therefore, considering other factors such as the length of the finger is essentially significant. In this paper, we also consider the relative length of the finger which is the quotient of tip-root-length and the palm radius, to handle scale variances as in [10]. The relative finger length vector is therefore presented as $l = [l_1, l_2, ... l_N, 0, ... 0]^T \in R^{5x1}$ (13)

where *N* represents the number of fingers.

V. Hand Pose Estimation Using Deep ISM

This section describes the process of estimating joint location in the testing stage. Ideally, we detect the location of a human hand from an input image by an implicit shape model hand detector. Then, we extract a hand from the detected implicit shape called codebook through cropping. Eventually, a well-processed cropped image is sent into the pose estimator network for generating pose configuration.

A. Generating 3D Pose Hypothesis from ISM

The well-processed cropped image produced by the proposed hand detector is used to generate pose hypothesis. Our deep ISM network is pre-trained and therefore, could directly obtain the results by forwarding the input into the network, see Fig. 5. However, the output due to the hand detector input image is the true fixed positions of the hand joints in the real world or camera coordinates, while the desired output should be the hand pose with joint positions. So, the output is in an unsuitable state at the bounding box due to the relative position.

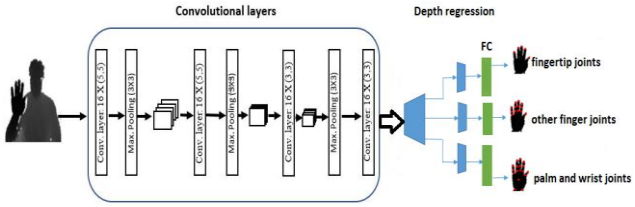

Fig. 5. A deep network architecture for the proposed system.

As a consequence, we had to perform some post-processing techniques to generate the smooth final pose of the hand. In performing some post-processing techniques, we first specify the range of the predicted value within the bounding box to be [-0.5,0.5]. However, some of the output of the pose estimator may surpass this value.

Therefore, we apply a normalization equation to the estimated value exceeding the range values to make sure that the output results are confined inside the boundary. Using the normalization equation, both maximum and minimum values are estimated in the three axes (*x*, *y*, *z*). For example, just similar to using the x-axis. The upper and lower values are expressed as:

$$\begin{aligned} X_{max} &= \max(\max(x_1 ... x_j), 0.5) \\ X_{min} &= \min(\min(x_1 ... x_j), 0.5) \end{aligned} \quad (14)$$

Where x_j represents the *x* position of joint 1 and *j* is the total number of joints. Finally, the equation (15) normalizes all the predicted values using the length of the axis.

$$(x_1 ..... x_j) = \frac{(x_1 ..... x_j)}{X_{max} - X_{min}} \quad (15)$$

To this end, all the estimated results which surpass the range of [-0.5,0.5] will be normalized. Additionally, the results which are inside the range will stay fixed. After the normalization of the predicted values, the hand pose can now be generated in the camera coordinates. To reduce the numbers of misallocated joints, we apply neighbor pixel votes to refine finger joint location. All the finger joints are re-estimated except finger roots which are in the palm area. Every finger joint other than root joints will get votes from the foreground pixels around them and therefore the average votes each joint obtains is set to be its location post-refinement.

VI. System Implementation Details

In this section, we explain details about the deep ISM

network. The network is based on the pre-trained model of DeepLab [30] and [31]. The training process for deep ISM is in stages starting from hand detector network, finger detection, and finally pose estimator network. Initially, we train a hand detection network. Then we conduct finger localization using the root-center-angle algorithm. Lastly, we train the hand detector network again using the convolutional network approach such that the two networks can operate by sharing the same convolutional layers. This will ease the training process and reduce computational time.

A. Training Deep ISM Network

Regression of joints is thoroughly performed in stages such that joints with high flexibility are regressed first and those with less flexibility are regressed last. In the first stage, all joints of five fingers are regressed with reference to the position of palm, while in the second stage, only locations of palm joints are regressed. This includes the wrist, finger roots, and palm centers.

We train the proposed deep ISM network regressively with the learning time steadily decreased. The parameters of deep ISM are constrained by the objective function specified by the following equation:

$$\arg\min \| C(E_p Z(t) \| + \| z \|_2^2 \quad (16)$$

where Z represents the parameters within the network, $Z(t)$ is the training samples in the dataset, C is the training loss function, and E_p is the estimated pose at time t from deep ISM network. The initial training rate is set to 0.0001 dropped by a ratio of 0.9 after every 100k iterations. One input image is augmented to different directions and fed into our network and trained for 2millions iterations. Regularization is by the weight decay rate, which is provided as = 0.001.

VII. Experiments

This section describes the experiments conducted to validate the proposed hand pose estimation system. The experimental setup is briefly presented and the dataset used in the experiment is then introduced. Empirical results are compared with different state-of-art methods which are recently published to evaluate the performance of the proposed system. The significance of integrating hand detector with pose estimator was also validated through experiments. Finally, we demonstrate the advantage of localizing finger joints in improving the accuracy of hand pose estimation.

A. Setting Experiment

In the experiment, we use Caffe [38] as a deep learning framework because Caffe provides an efficient implementation in the prediction. It is the fast way to apply regression problems and it supports many new layer functions such as the ISM layer. ISM layer can be customized and supported by Caffe. In simplifying the training process, Caffe is integrated with Cpp (C++) OpenCV library which is also linked with MATLAB using Mex-function because Caffe model can be well integrated with C++ to make predictions.

B. Datasets

Herein, we introduce three datasets that were used in the experiments. These datasets are publicly available and have been used by many recent researchers [20], [25], [35], [37]. Therefore, a thorough comparison with other bodies of research work will be carried out on these three datasets to evaluate the performance of the proposed system. The NYU [32] datasets use one handshape in training data and two handshapes in test data, one of which is from the training set. NYU dataset contains 8252 testing sets and 72,757 training sets of RGB-D images. The frames contained in the dataset are well annotated with the precise ground truth of pose configuration, and therefore exhibit a great variability to different poses.

On the other hand, the Imperial College Vision Lab (ICVL) [33] datasets involved ten pose-signers with similar hand sizes, and all are annotated with the single handshape model. The ICVL dataset contains about 180k depth frames training set having different hand poses obtained from 10 dissimilar subjects. The test set contains two sequences, each with approximately 700 frames. Each hand pose has 16 annotations of joints.

Microsoft Research Asia (MSRA) [5] datasets contain 9 subjects from a different source and it best performs for finger joint evaluation. The MSRA dataset contains about 76K depth frames. The dataset has sequences of 9 different subjects with 17 signs for each subject. We use 8 subjects for training and evaluate on the left 1 subject with the repeated practice for all subjects.

C. Evaluation Metrics

In this paper, we use four different evaluation criteria to evaluate the estimation results. The first two metrics evaluate joint locations in the fingers and the last two evaluate the fraction and mean error of the whole hand, namely:

1) All finger errors are calculated as an average Euclidean distance for present joints in each finger given in millimeters (mm).
2) All fingertips errors calculated from an average Euclidean distance for fingertip in each finger also in mm.
3) The fraction of sample error distance within a threshold. This criterion measures the percentage of success frames whose error distance to each joint is less than a certain threshold. This is considered as the most ambiguous evaluation criterion because the single wrongly located joint may decline the judgment of the entire hand pose [8].
4) Mean error distance of different joints and their corresponding average. This is the most preferred criteria in the literature of hand pose estimation. It has been used for comparison by many research works due to its simplicity in evaluation.

A. ICVL Evaluation

The test set of the ICVL datasets contains two sequences, each with approximately 700 frames. Each hand pose has 16 annotations of joints. During experiments, ICVL hand posture datasets were used to evaluate the proposed system estimation results, and compare the estimation results with seven state-of-the-art works namely: Deep prior ++ [35], Tang et al [33], Zhou et al [36], Crossing Nets [20], REN [16], BigHand [25], and Point Net [37]. The results for ICVL evaluation are presented in Fig. 6. Depicted from the figure, the proposed approach attains an accuracy of 96.4% at

threshold level of 40mm, demonstrating a consistent improvement of the proposed system in overall error threshold.

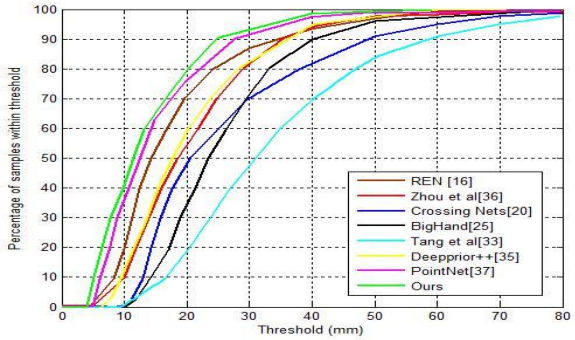

Fig. 6. The comparison with state-of-art on ICVL dataset.

On the other hand, the mean error distance of different joints and their average on ICVL hand posture datasets is presented in Fig. 7. In this paper we consider the mean error distance of only 11 joints in comparison using the fourth evaluation criteria since many works of literature [35], [8], [37] provide the results for only 11 joints. The experimental results demonstrate the supremacy of the proposed approach over many contending approaches, by proceeding with the lowest error and achieve 7.0 mm average error distance.

This is slightly equivalent to Point Net [37] which achieved state-of-art accuracy with an overall mean of 6.9 mm whereas others are greater than 7mm. This is the indication that localization of finger joints can greatly improve the accuracy of hand pose estimation systems.

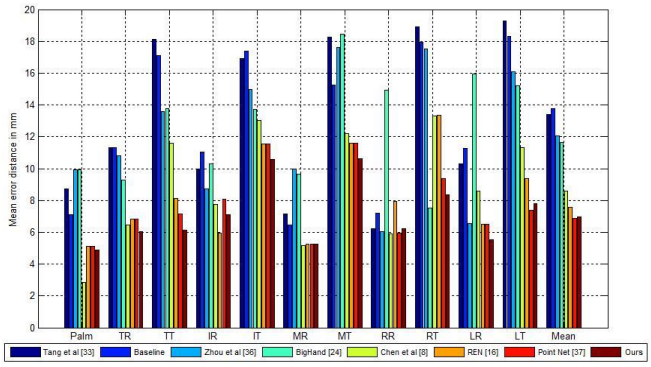

Fig. 7. The comparison of an average error distance of different joints on ICVL hand posture dataset.

D. MSRA Evaluation

The MSRA dataset [5] contains about 76K depth frames captured using a time-of-flight camera. The dataset has sequences of 9 different subjects with 17 signs for each subject. In this work, a leave-one-out cross-validation is preferred since it is the common evaluation procedure [35], [10]. Training of our deep network was in line with MSRA dataset, i.e. using 8 subjects for training and evaluate on the left 1 subject. We perform a repeated practice for all subjects and present the average errors over fingers in Fig. 8.

The marked red plot in Fig. 8 represents our method, which outperforms all other methods on the plotted metric, indicating that the proposed approach can handle multiple users' hands. For example, when the error threshold is 30mm, the proportions of good frames of the proposed approach are about (5%) better than Deep prior++ and (2%) better than Point Net [37]. The nearness of performance between our approach and the Point Net [37] is due to refinement of the finger joints location, especially fingertips which both methods considered.

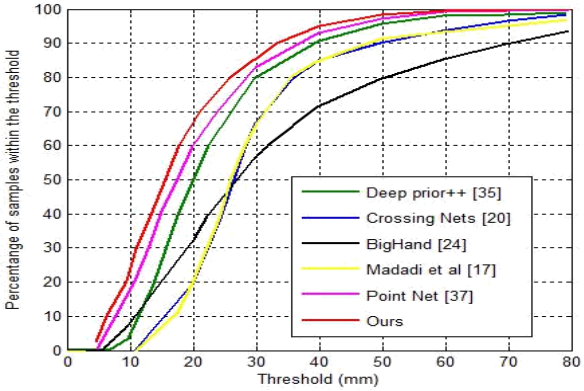

Fig. 8. Success frame comparison with the state-of-state on the MSRA dataset.

Finally, we compute all fingers errors using Euclidian distance for joints in each finger (in Millimeters), and fingertip error using average Euclidian distance for fingertip in each finger and report results in Fig. 9.

In both cases, our method performs better than the competing baselines. This significant improvement implies that the locations of fingertips can be clearly estimated and therefore inaccurate hand pose estimation due to fingertip errors can be alleviated.

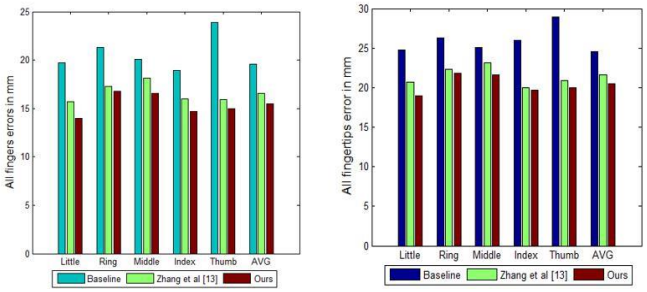

Fig. 9. Fingers and fingertip errors comparison with state-of-art methods on MSRA dataset.

E. NYU Evaluation

The NYU dataset [32] contains 8252 testing sets and 72,757 training sets of RGB-D images taken by the structured light-based sensor Prime sense Camine 1.09 from three dissimilar viewpoints. The frames contained in the dataset are well annotated with the precise ground truth of pose configuration, and therefore exhibit a great variability to different poses. For the purpose of the experiment, we only utilize depth data from a single camera. Considering the established evaluation standard [35], we selectively use depth images from viewpoints 1 and 14 joints for calculating values. We then present and compare the obtained estimated results with other state-of-the-art approaches and report results in Fig. 10. The figure, presents the results for test examples against max joint error below the threshold. Using evaluation criteria three (3), it can be seen that our proposed method outperformed the state-of-the-art methods by attaining 89% of success frames at an error threshold of 40m. Nonetheless, the difference to other methods is significantly small. This is because the dataset has small pose variations and also due to the error of annotation.

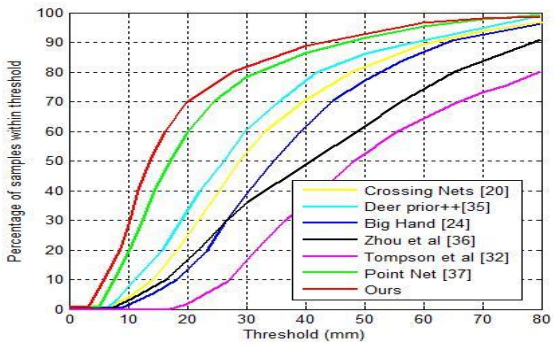

Fig. 10. Success frames comparison with the state-of-the-art methods on the NYU dataset.

Similarly, we measure the error distances of each joint and present the average error distances of each joint in Fig. 11. Then we compare our approach with the state-of-art methods, and the results show less error on average of all joints in our method than all other methods. Again, we obtained a slight difference with Point Net [37] For example, in overall average error distance, which is 10.0mm and that of Point Net [37] is 10.5mm. The reason for this slight difference has been explained in the previous sections.

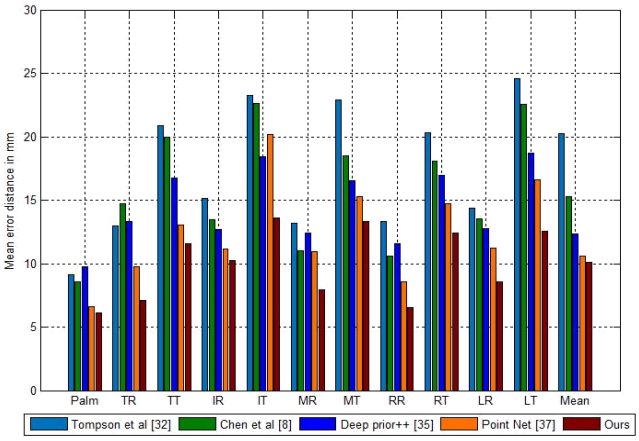

Fig. 11. Mean error distance comparison on the NYU dataset.

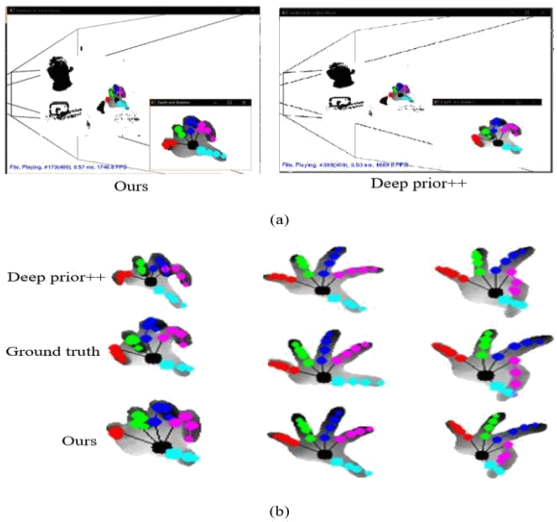

Fig. 12. Visualization of some estimated results from our hand pose estimation system, (a) ICVL dataset and (b) MSRA.

F. The Qualitative Analysis

Some qualitative results from some of the datasets used are shown in Fig. 12. In general, our hand pose estimation system provides significantly better results compared to many contending approaches. This can be attributed by the robust hand detector, the implicit shape model and ideal finger localization using a RCA algorithm. Additionally, the powerful deep model presented in this paper increases the prediction of highly accurate poses for complex articulations.

VIII. Conclusion

In this paper, an approach for a hand pose estimation system based on localization of finger joints that exerts a higher degree of freedom is presented. We demonstrated that accurate regression of finger joints delivers significant cues for joint estimation by lowering errors of estimation and therefore, improve hand estimation accuracy. In addition, we innovatively divided hand pose estimation process into three stages. Firstly, we detected the human hand using a robust and reliable hand detector based on implicit shape model.

This model uses scale-invariant interest point detectors and descriptors to provide not only a position but also a scale of the detected hand. Secondly, we perform a hierarchical regression, starting with the hand parts with high flexibility followed by hand parts that are less flexible.

Lastly, a complete end-to-end hand pose estimation system is implemented. The implementation is based on the well-designed hand detector integrated with the stable pose estimator into one pipeline. This conferred strong hand detection which ultimately improves the accuracy of the final pose estimates. Consequently, our system can be applied to multiple viewpoint systems and to any conditions of multi-users.

In the future, the focus would be on further improvement of accurate hand and fingers detection since these are the foundation of accurate hand pose estimation. Finally, the proposed system can be integrated with some machine learning approaches to develop a learning assistive tools for hearing disabilities, and other natural human-machine interaction systems that can achieve a greater user experience.

Conflict of Interest

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

Author Contributions

Author Stanley Leonard contributed to the main idea of the proposed study, conducted an experiments and write-up of the paper. Authors Jamal Banzi and Aloys Mvuma also contributed to the formulation of the study ideas and help to improve the presentation of the paper. Professor Aloys Mvuma was also an advisor for the entire duration of the study. However, all the authors had approved the final version of this paper.

References

[1] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 4, pp. 570-578, July 1993.

[2] E.Barsoum, "Articulated hand pose estimation review," arXiv preprint arXiv:1604.06195, 2016.

[3] A. Erol, G. Bebis, M. Nicolescu, R. D Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, 52-73, 2007.
[4] A. Sinha, C. Choi, and K. Ramani, "Deephand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4150-4158.
[5] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1106-1113.
[6] X.Sun, Y. Wei, S. Liang, X. Tang and J. Sun, "Cascaded hand pose regression," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 824-832.
[7] P. Krejov, A. Gilbert, and R. Bowden, "Guided optimization through classification and regression for hand pose estimation," *Computer Vision and Image Understanding*, vol. 155, pp. 124-138, 2017.
[8] C. Zimmermann and T .Brox, "Learning to estimate 3D hand pose from single rgb images," in *Proc. International Conference on Computer Vision*, vol. 1, no. 2, p. 3, 2017.
[9] T. Y. Chen, P. W. Ting *et al*., "Learning a deep network with spherical part model for 3D hand pose estimation," *Pattern Recognition*, vol. 80, pp. 1-20, 2018.
[10] De La Gorce, M. Fleet, and N. Paragios, "Model-based 3D hand pose estimation from monocular video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1793-1805, 2011.
[11] S. Melax, L. Keselman, and S. Orsten, "Dynamics based 3D skeletal hand tracking," in *Proc. Graphics Interface*, pp. 63-70, Canadian Information Processing Society, 2013.
[12] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1297-1304.
[13] C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun, "Randomized decision forests for static and dynamic hand shape classification," in *Proc. IEEE Computer Society Conference on* Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, pp. 31-36.
[14] C. Zhang, G. Wang, H. Guo, X. Chen, F. Qiao, and H. Yang, "Interactive Hand Pose Estimation: Boosting accuracy in localizing extended finger joints," arXiv preprint arXiv:1804.00651, 2018.
[15] Q. Ye, S. Yuan and T. K. Kim, "Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation," in *Proc. European Conference on Computer Vision*, 2018, Springer Cham 2018, pp. 346-361.
[16] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim and D. Freedman, "Accurate, robust, and flexible real-time hand tracking," in *Proc. the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM*, 2015, pp. 3633-3642.
[17] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Region ensemble network: Improving convolutional network for hand pose estimation," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 4512-4516.
[18] M. Madadi, S. Escalera, X. Baró, and J. Gonzalez, "End-to-end global to local CNN learning for hand pose recovery in depth data," arXiv preprint arXiv: 1705.09606, 2017.
[19] D. Fourure, R. Emonet, E. Fromont, D, Muselet, N. Neverova, A. Trémeau, and C. Wolf, "Multi-task, multi-domain learning: application to semantic segmentation and pose regression," *Neurocomputing*, vol. 251, 68-80, 2017.
[20] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang, "Hand3D: Hand pose estimation using 3D neural network," arXiv preprint arXiv:1704.02224, 2017.
[21] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation" in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[22] N. Neverova, C. Wolf, F. Nebout and G. W. Taylor, "Hand pose estimation through semi-supervised and weakly-supervised learning," *Computer Vision and Image Understanding*, vol. 164, pp. 56-67, 2017.
[23] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 3316-3324.
[24] G. Poier, K. Roditakis, S. Schulter, D. Michel, H. Bischof, and A. A. Argyros, "Hybrid one-shot 3D hand pose estimation by exploiting uncertainties," arXiv preprint arXiv:1510.08039, 2015.
[25] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.K. Kim, "Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis," *In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* IEEE, 2017, pp. 2605-2613.
[26] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, and J. Yuan, "Depth-based 3D hand pose estimation: From current achievements to future goals," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[27] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using RGB and depth data," in *Proc. the IEEE International Conference on Computer Vision*, 2013, pp. 2456-2463.
[28] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, pp. 259-289, 2008.
[29] J. Müller and M. Arens, "Human pose estimation with implicit shape models," in *Proc. First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, pp. 9-14, ACM, 2010.
[30] X. Chen, C. Shi and B. Liu, "Static hand gesture recognition based on finger root-center-angle and length weighted mahalanobis distance," *Real-Time Image and Video Processing*, 2016, vol. 9897, p. 98970U, International Society for Optics and Photonics, 2016.
[31] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2018.
[32] W .Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," arXiv preprint arXiv:1605.08104, 2016.
[33] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG*), vol. 33, no. 5, p. 169, 2014.
[34] 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.
[35] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3D training data for fine hand pose estimation," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4957-4965.
[36] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3D hand pose estimation," In ICCV workshop, vol. 840, pp. 2, 2017.
[37] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," arXiv preprint arXiv:1606.06854, 2016.
[38] L. Ge, Y.Cai, J. Weng, and J.Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8417-8426.
[39] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. the 22nd ACM International Conference on Multimedia*, 2014, pp. 675-678.

Stanley Leonard received his B.Sc. and MSc. degrees from Dar es salaam Institute of Technology (DIT) and the University of Dodoma in Tanzania in the years 2009 and 2012 respectively. 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) in Tanzania. He has four publications, in international journals. His current research interests include artificial intelligence and machine learning techniques for human-computer interaction application domains.

Jamal Banzi is a lecturer at the Sokoine University of Agriculture. Dr. Banzi received BSc. 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 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, AI Assisted Sign Language Recognition for hearing disabled communication, and AI+ Agriculture.

Dr. Banzi has published more than 10 scientific articles in top-tier journals and conferences including IEEE, Journal of Machine learning and Applications, Journal of Intelligent and Autonomous Systems, ICCCV, ICIC, etc.

Aloys N. Mvuma received his BSc in electrical engineering degree from University of Dar es Salaam (UDSM) Tanzania in 1994, MSc in information science from Shimane University, Japan and Doctor of Engineering (Systems Engineering) from Hiroshima University, Japan in 2000 and his PhD in information science in 2003. He then joined School of Engineering, Hiroshima University as an assistant professor from March 2003 to 2005. He was promoted to an associate professor in 2013.

He was saving as a lecturer in the Department of Telecommunications Engineering at UDSM, Senior Lecturer at School of Informatics, College of Informatics and Virtual Education at University of Dodoma (UDOM) Tanzania, in the year 2003 and 2008 respectively. In 2019 he joined Mbeya University of Science and Technology (MUST) as a vice-chancellor.

His research interests include adaptive signal processing, digital communication systems and ICT for development. He has published over 30 conferences and 23 journal papers. He is a registered member of Engineers Registration Board (ERB) and Institute of Electrical and Electronic Engineers (IEEE).