# Hand Pose Estimation Based on Deep Learning Depth Map for Hand Gesture Recognition

Naima Otberdout
LRIT-CNRST URAC 29
Mohammed V University In Rabat
Faculty of Sciences Rabat, Morocco.
Email: naimaotberd@gmail.com

Lahoucine Ballihi *
LRIT-CNRST URAC 29
Mohammed V University In Rabat
Faculty of Sciences Rabat, Morocco.
Email: ballihi@fsr.ac.ma

Driss Aboutajdine
LRIT-CNRST URAC 29
Mohammed V University In Rabat
Faculty of Sciences Rabat, Morocco.
Email:aboutaj@fsr.ac.ma

***Abstract*—Hand pose estimation plays an important role in many applications, especially in human-computer interaction. Therefore, this topic has matured quickly in recent years. In this work we focus on the hand pose estimation from a depth map using convolutional neural networks. We propose a method for hand pose estimation by formulating a regression problem whose solution is the $16$ hand joint locations. This method consists of two stages, the first one dealing a hand detection based on contours, the second one consists hand pose estimation using convolutional neural networks. In this paper, we provide an extensive quantitative and qualitative experiments using real word depth maps from ICVL dataset. We perform a comparative evaluation with the state-of-the-art approaches to show the effectiveness and the accuracy of our method. Moreover, we propose a new application for hand gesture recognition based on our hand pose estimation method. The experimental results reported on test sequences of ICVL dataset show that the proposed application yields interesting performances and gives a marked improvement in recognition rate.**

## I. Introduction

Capturing and estimating human hand movements is an old computer vision problem that has been studied since the nineties. However, it is still difficult to solve because of the large pose variations, the highly articulated structure, significant self-occlusions, viewpoint changes and data noises. In addition, real-time performance is often needed in most applications. The use of GPU acceleration or multi camera setups helps to overcome some of these challenges, but limits deployment to the general public. Recently, the emergence of depth sensors, including structured-light and time-of-flight sensors is one of the major motivations of this topic. But given the many challenges presented by the hand, even with depth sensors it is still very challenging to estimate the 3D hand pose from a single noisy depth map.

Hand pose estimation can be exploited in a wide range of applications such as robotic control [1] humanoid animation [2], sign language recognition [3], human-computer interaction [4] and visual interfaces [5]. Each of these applications requires different level of precision, for some applications, it is sufficient to recognize some poses from the same viewpoint, but most of them require more precision and robustness.

The introduction of RGB-D sensors and a multitude of practical applications have led many researchers to work on the topic of 3D hand pose estimation and have spurred new advances. In this paper, we tackle the problem of hand pose estimation from a single depth map using convolutional neural networks. The main contributions of our work are summarized as follows:

- We propose a method for hand detection based on contours to segment the hand from the background in a single depth map.
- Given the hand detected, we use a CNN architecture to train a new model for hand pose estimation.
- We exploit the success of our approach to develop a new application for hand gesture recognition. This application is able to recognize the number indicated by the hand pose in a depth image.

## II. Related work

Hand pose estimation is an old problem which was the goal of several studies within computer vision. Recently with the emergence of depth sensors, this field receives more and more attention and many proposed works have achieved good performance (at the level of both accuracy and runtime) by different methods. Generally, the first works proposed to solve this challenging topic were the marker-based methods which use gloves to estimate 3D hand pose [6]. These solutions were used for several years, but the markers are uncomfortable for users and they still less flexible, expensive and often inhibit free movement. Thus, marker-less methods which try to estimate the 3D hand pose just from a single image are attractive and receive more interest.

The figure 1 illustrates our representations of different stat-of-the-art approaches. Current marker-less methods for hand pose estimation can be generally divided into the following two classes:

**Model-based approaches**: These approaches [7]–[11] use a 3D hand model which is fitted against the input image.

978-1-5090-4062-9/17/$31.00 © 2017 IEEE

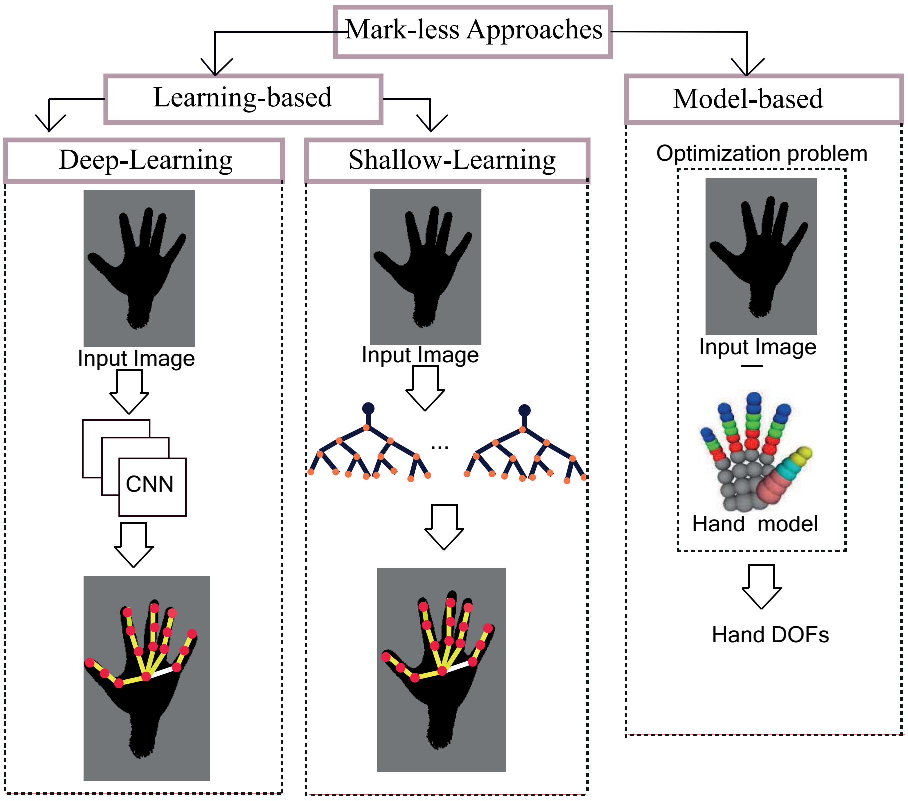


Fig. 1: Overview of the state-of-the-art: We have divided the exist approaches into three classes: *Model-based* approaches that use a hand model which is fitted against the input image. *Learning-based* approaches which try to learn direct model from the image space to the pose space using either Deep Learning techniques or shallow-Learning ones.

They are generally formulated as optimization search and use a loss function that measure the differences between the hand pose in the input image and that of 3D hand model. By minimizing this loss function, we can find the parameters (degrees of freedom (DOFs)) of the 3D model reproducing the observed movements in the input image. Model-based approaches differ mainly by (1) the different visual features extracted from the images, (2) the 3D hand model used, (3) the optimization techniques used to solve the estimation problem, and finally (4) the technique used to initialize the optimization problem and guarantee a fast convergence. This type of solutions provide more accuracy than Learning-based approaches, However they suffer from initialization problem that is often solved using the pose estimated in previous images [5]. In addition, these approaches achieve low frame rates (12 FPS in [11] ) and need to be accelerated using a GPU.

**Learning-based approaches**: These approaches aim to directly estimate the positions of the hand joints from a depth map without using a 3D hand model. They try to learn a model that match the input image with the hand pose desired using machine learning algorithm. Studies prove that the learning-based approaches provide better real-time performance compared to model-based approaches, as the process of extracting the image features is faster. However, the main challenge of learning-based approaches is that they heavily rely on the quality of learning data, and they require a large datasets to model a large number of possible poses. Being interested in the deep-learning we divide Learning-based approaches into two classes:

1) **Shallow-Learning based approaches:** In the state-of-the-art Random Forest is one of the most shallow-learning algorithm used for hand pose estimation. For example, Tang et al. [12] use Latent Regression Forest to estimate the hand pose. To do so, the image is divided recursively according to a Latent Tree Model to two regions, until each region corresponds to a single joint of the hand. Finally Latent Regression Forest is used to estimate the position of each joint. The difference between this work and Li et al. [13] is to replace the latent tree model used to guide research in latent regression forest in [12], by the segmentation index points , which are more flexible. Tang et al. [14] propose Semi-supervised Transductive Regression Forests, which start to classify the hands viewpoint,and then, the target joints and the offset between the joint location and the image center.

2) **Deep-Learning based approaches:** Even more recently, other works are motivated by the success of Deep learning. Actually, deep convolutional neural networks (CNNs) have

made impressive progresses on the majority of recognition tasks including image classification, object detection, semantic segmentation [15] and especially on human pose estimation [16], [17] Thus, a lot of recent approaches try to overcome the difficulties of hand pose estimation by using CNNs. Given a depth map Oberweger et al. [18] evaluate several architectures for CNNs to predict the 3D joint locations of a hand and Suau et al. [19] use convolutional neural networks to infer 2D heat-maps corresponding to joint positions, followed by inverse kinematic approach for 3D pose recovery from a 2D image. However these approaches are less efficient in the presence of occlusions or missing depth values. Oberweger et al. [20] use CNN to predict an initial estimate of the 3D pose which is used to synthesize an image , then they derive a pose update using synthesize image with the input depth map . The update is applied to the pose and the process is iterated. Their results show the accuracy of this method, but it is tested just on a GPU. Ge et al. [21] firstly project the input depth image onto 3 orthogonal planes, these multi-view projections are used to regress for 2D heat-maps which estimate the joint positions on each plane and then fused to produce final 3D hand pose estimation.

## III. Methodology and Problem formulation

Our work can be divided into two stages. firstly, the detection stage aims to extract the region of interest from the depth image. Here, our goal is to find the hand center location C with $C = (u, v, d)$ such that $(u, v)$ is the 2D position of the hand center and $d$ its depth. Then, we extract from the image a cube centered on C. This cube after preprocessing, will be the input of the next stage.
For the hand pose estimation stage, the problem can be divided into estimating the 3D position of a discrete set of hand joints (16 joints in our case). Therefore, we aim to estimate the pose P represented by the 3D coordinates of N=16 joints illustrated in figure 2 such that: $P = \{j_i\}_{i=0}^{i=N}$ and $j_i = (u_i, v_i, d_i)$.

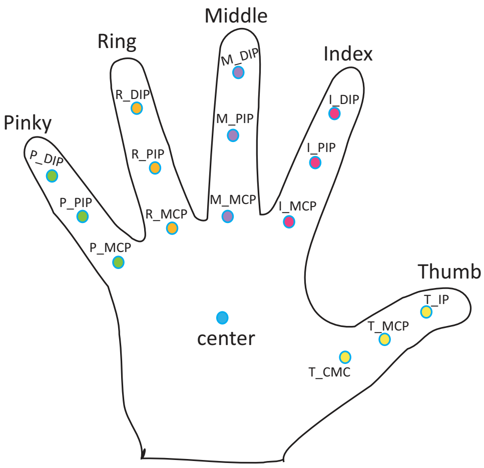


Fig. 2: The 16 hand joints estimated in our work.

Figure 3 provides an overview of our work. Given a hand depth map, we start by detecting the region of interest (ROI) in the image using a segmentation based on contours. We process this initial region to obtain a patch image of normalized depth values. Given this patch, we use a CNN architecture to estimate the hand pose which is defined by the set of 16 joint locations.

For the remnant of this paper, we discuss the region of interest extraction in section IV. Following this, we present the CNN architecture used for hand pose estimation in section V, then, we evaluate our model and present our results, quantitatively and qualitatively in section VI. Finally, in section VII, we present our application of hand gesture recognition and its evaluation.

## IV. Hand detection

Before estimating the hand pose, we have to extract the region of interest from the depth image. In the state-of-the-art, one of the most simple hand segmentation method is depth thresholding [18], it assumes that the hand is the nearest object to the camera. It is simple, but does not cover all situations. There are several other segmentation methods. Tompson et al [19] use random forest that assigns each pixel a probability if it belongs to the hand or background and more recently, Riegler et al [22] use a mean shift segmentation to find the hand center by iterations.

In this work we use a contours-based segmentation. We assume that the hand occupies a large area and represents the largest contour in the image. This method is not affected by the background color as skin color-based hand segmentation approaches nor the distance of the hand to the camera as the depth thresholding approach. However, it works well when the hand is the biggest object in the image.
Once we detect the hand contour as the larger contour in the image, we consider its center as the hand center desired and its size as the hand size. Following this, we extract from the image, the region of interest which is represented by the cube of hand size centered on the hand center detected. Convolutional neural networks need an input of a predefined size, therefore, most of the deep-learning based methods needs a preprocessing stage to prepare the input of CNN. As preprocessing, we resize the ROI extracted to a uniform patch of size $128 \times 128$ and we normalize the depth values to [-1, 1] with background pixels set to 1 as in [19].

## V. Hand Pose Estimation

After extracting the region of interest from the depth map in the detection stage, we use CNN to estimate the hand pose by directly regressing the 3D joint locations simultaneously. The architecture used (figure 4) is approximately the same as that used by Oberweger et al. [18]. It contains 2 convolutional layers using 8 kernels of size $5 \times 5$. Each one is followed by max-pooling layer of size $4 \times 4$ for the first layer and $2 \times 2$ for the second one. Following this, we use a third convolutional

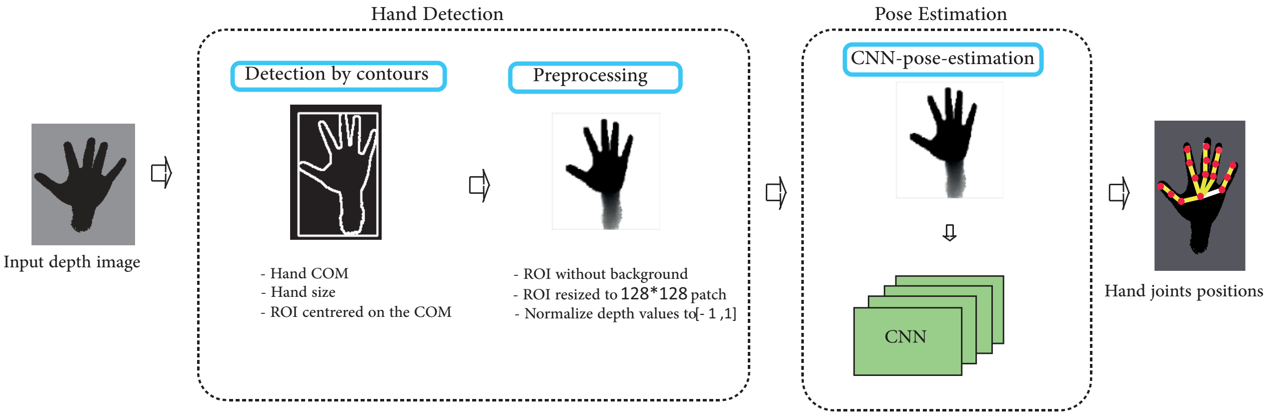


Fig. 3: Overview of the proposed method

layer with 8 kernels of size 3 × 3, and finally 2 fully connected layers containing 1024 neurons and followed by a dropout layer with dropout ratio 0.3. The stride of each layer is set to 1 with no padding, and all layers use Rectified Linear Unit (ReLU) as activation function.

Between the fully connected layers and the output layer that should contain 3N neurons representing the 3D coordinates of all joints, we add a layer with less neurons than needed to represent the full pose. This layer that we called PCA-layer is used for dimensionality reduction, it exploits the correlation of different joint locations to represent the pose in a small dimension using Principal Components Analysis.

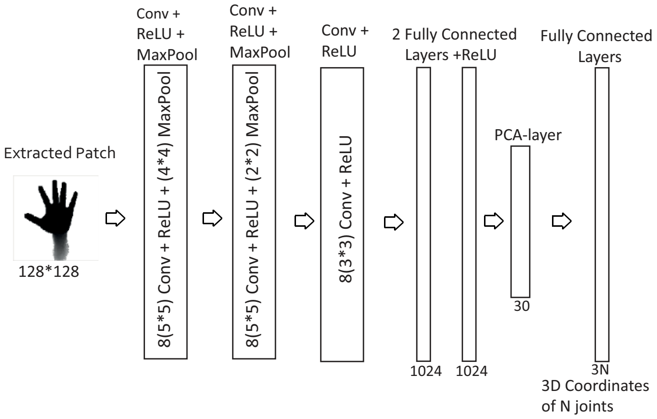


Fig. 4: CNN architecture used for hand pose estimation. Numbers under the four last layers indicate their neurons number.

Principal Components Analysis (PCA) has been used by Wu et al [23] to well represent the correlation of different joint positions. Introducing this method is beneficial for two reasons: it enforces the output result to respect certain hand geometric constraints, in this way it ensures the geometric validity of estimated poses and prevents estimating impossible ones, at the same time it allows to represent the full pose in low dimension.

PCA is used with CNN as an additional layer with less neurons than needed to represent the full pose $(<< 3N)$. This layer exploits the correlations between joint positions to minimize the number of neurons in the layer. Many works have proved that introducing PCA in CNN is beneficial compared to direct regression in the full pose space.

More recently Zhou et al. [24] have replaced PCA by using a model based deep learning approach that fully exploits the hand model geometry, they add a new layer to the CNN architecture that realizes the non-linear forward kinematics. This layer is considered as an intermediate representation of the hand with its DOFs, it transforms the predicted joint angles to their locations. Our comparison to state of the art shows that this intermediate pose representation is roughly equivalent to the use of PCA for accuracy and pose validity.

**Details of learning:** The CNN used is implemented in Python using Theano library. We have trained our model on 22K depth images from ICVL dataset.We have used back-propagation algorithm with stochastic gradient descent and Huber loss function, which compute the difference between the prediction and the ground truth. Table I provides the meta-parameters values used for training. To overcome the overfitting, we have used a regularization term for weight decay and dropout in fully connected layers and we stopped the training after 100 epochs.

TABLE I: Meta parameters values used in training.

| **Meta parameters** | **Value** |
|---|---|
| Batch size | 128 |
| Momentum | 0.9 |
| Weight decay | 0.001 |
| Learning rate | 0.01 |
| Epoch number | 100 |
| PCA components | 30 |

## VI. Experimental Results

### A. Dataset and Evaluation Metrics

**Dataset**: To evaluate our work we have used the hand pose ICVL dataset [12], which contain 180K fully annotated depth maps from 10 different subjects. This dataset is captured by Intel Creative Interactive Gesture Camera, which is a time-of-flight depth sensor. This depth sensor captures depth images at a lower noise level than structured-light sensors.

For training, ICVL hand poses dataset provides $22K$ original depth images and by applying in-plane rotations to this set, the final dataset contains 180K training images representing a various hand poses. All these images are annotated with N=16 hand joint locations. For testing, ICVL dataset provides two test sequences, each one contains over 800 hand depth maps annotated with 16 hand joint locations.

**Evaluation metrics**: For evaluation, we have used the two metrics generally used in the state-of-the art of this topic as in [18]. These metrics are: (1) the mean distance error which represents the difference between the estimated location and the ground truth in (mm), and (2) the fraction of test frames for which all joints are within a threshold error.

**Comparison with the state of the art**: To evaluate our work we have compared our model to DeepPrior model proposed by Oberweger et al. [18] and the model of Zhou et al. [24]. These two recent baselines use also convolutional neural networks to estimate the hand pose. Moreover, we compare our model results to the work of Tang et al. [12] who use Regression Forest to estimate the hand pose from a single depth map.

### B. Hand Pose Estimation

To evaluate our model accuracy, we conduct a comparison between the results of this model and the state of the art approaches. We first show the results of detection stage evaluation. Following this, we present the quantitative results of our model compared to tree baselines [12], [18], [24] on both ICVL test sequences. Finally, we show our qualitative results.

*1) **Hand detection Evaluation**:* As we divided the hand pose estimation process in two stages, we want to evaluate each stage separately to measure the accuracy of the used hand detector and its effect on the model. To evaluate the hand detection stage, we compare our model results obtained by using the annotated hand center from ICVL dataset, with our model results using the hand center detected as presented in section IV. Figure 5 shows the results of this evaluation.

*2) **Model Evaluation**:* In this part we aim to evaluate our model by comparing its results to those of DeePrior model [18], and other state of the art approaches. Figures 6 7 and tables II III show the results of this evaluation.

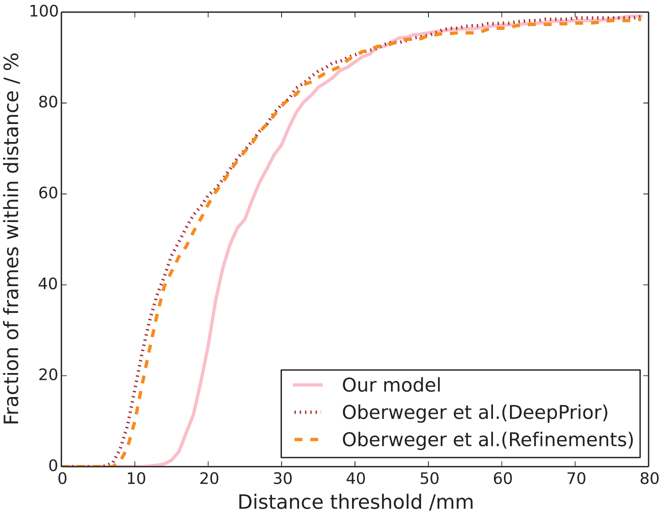


Fig. 6: Fraction of frames within a given distance error. Comparison between our model, DeepPrior model and the model of [18] using refinements on the first ICVL test sequence.

TABLE II: Mean error on the first sequence of ICVL dataset.

| Method | Mean error(mm) |
|---|---|
| Our model | 14 |
| DeepPrior [18] | 10 |
| Refinements [18] | 10 |

### C. Result Analysis

**Quantitative results**: The evaluation of the detection stage indicates that our hand detection method limits the performances of the model. According to the figure 5a the hand center incorrect detection adds about 7 mm to the model

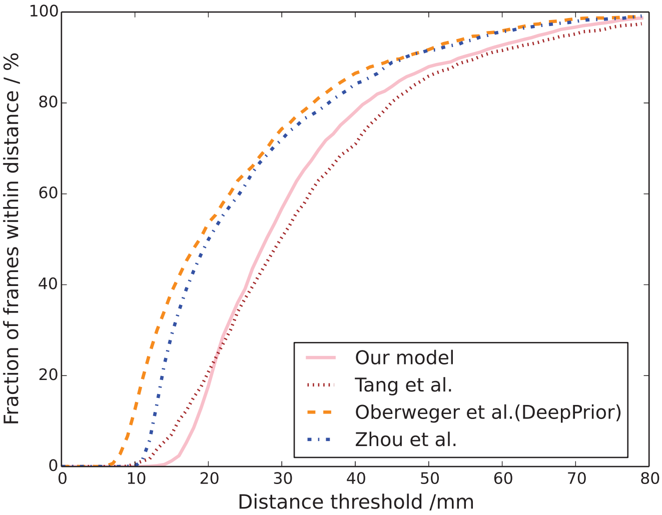
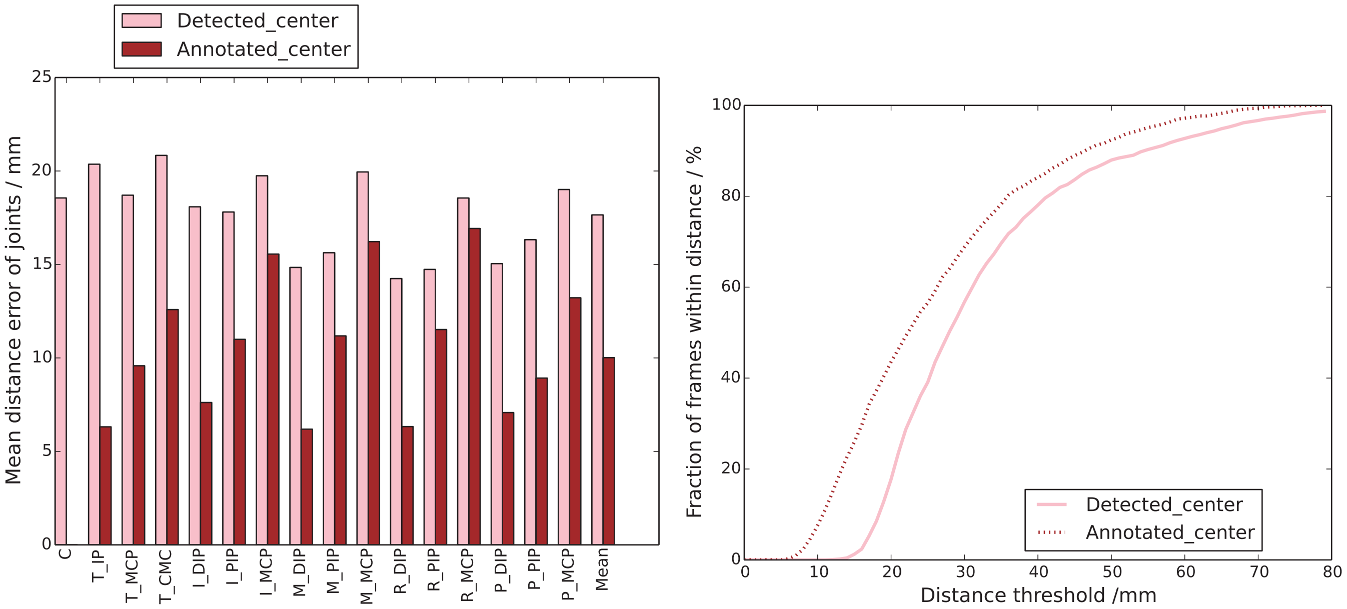


Fig. 7: Fraction of frames within a given distance error. Comparison with state of the art approaches on both ICVL test sequences.



(a)

(b)

Fig. 5: Hand detection evaluation: Comparison of results obtained by using the annotated hand center ($annotated - center$) to the results obtained using the detected hand center($detected - center$). 5a provides mean joint errors, and 5brepresents the fraction of frames for which all max joint error are within a given distance.

TABLE III: Mean error on the test sequences of ICVL dataset.

| Method | Mean error(mm) |
|---|---|
| Our model | 17 |
| DeepPrior [18] | 11 |
| Zhou et al. [24] | 12 |
| Tang et al. [12] | 14 |

mean error. This proves the detection stage importance in the hand pose estimation process, and indicates that we can significantly enhance our approach by improving our detector accuracy. However, even it limits the model performance, our hand detection method provides good results according to the comparison with the state of the art and the qualitative results.

Figure 6 and table II show the comparison of our model to the models of Oberweger et al. [18] on the first ICVL test sequence. Table II shows that our model does not exceed 14 mm as mean error on all joints, while this error attains about 10 mm for both other models. Figure 6 shows that the models of Oberweger et al. are more accurate, but the three models do not exceed 35 mm as maximal error for all joints in roughly 80% of the images.

Moreover, figure 7 indicates that our model performance remains comparable to the state of the art and provides good results. Indeed, although it is less accurate than DeepPrior model, it provides more accurate results compared with Tang et al. [12]. The accuracy of our model is also proven by our qualitative results 8.

**Qualitative results**: According to our qualitative results presented in figure 8 our model is able to estimate the general hand pose with good precision. By the use of PCA layer, the majority of the estimated poses are kinimatically valid and respect the hand geometry. However, the model is less efficient with complicated poses that show many occlusions between hand joints.

Figure 8 shows also some failure cases in the two last columns, this happens mostly when the image is noisy or has missing depth values, without forgetting the problem of incorrect annotations of ICVL dataset for some samples.

## VII. Hand gesture recognition

Based on our neural network model, we have implemented a hand gesture recognition application to recognize the number indicated by the hand pose. Our application principle is simple: given the predicted hand joint positions and computing the Euclidean distances between the hand center and the fingertip of different fingers, we can compute the number of unfolded fingers. This number corresponds to a number between 0 and 5 Indicated by the hand pose. A finger is considered as unfolded if it satisfies this condition:

$$\sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} > T$$

with $(x_c, y_c)$ and $(x_i, y_i)$ are the 2D positions of hand center and fingertip, respectively. T is a defined threshold. The threshold used is very critical as using a bad threshold greatly decrease the recognition rate, thus we performed a

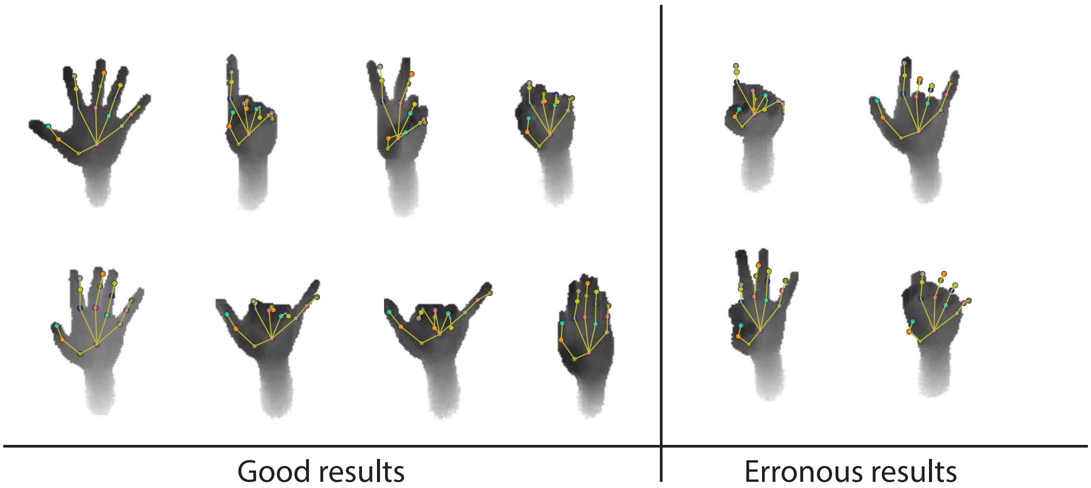


Fig. 8: Qualitative results: The predicted joint locations are shown on the depth images. The good results (left) show that our model is able to predict the hand pose and preserve its topology, but remains less efficient with complicated poses which present many occlusions and missing depth values (right).

large number of experiments varying this threshold until finding the best performing one. The experiment shows that this threshold depends on the finger length, thus we used different thresholds for different fingers. To obtain the threshold of each finger, we used our neural network model to estimate the distances between the hand center and fingertips in roughly 30 images in which the hand pose indicates different numbers. Using these distances we observed for each finger the maximum distance between the hand center and the fingertip when the finger is folded, and the minimum of this distance when the finger is unfolded. Finally, we choose for each finger a threshold between its maximum and minimum distance from the hand center.

For more accuracy we can use a term of normalization l such that the condition becomes:

$$\frac{\sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}}{l} > T$$

With l represents the largest distance between the hand center and a fingertip, it is used to normalize the distance. This normalization is important in order to be invariant to different hand sizes. Our application can be used in several ways, for instance, we can use it with Kinect depth maps captured in real time to teach children how to count, or we can use it to control a PowerPoint presentation. Moreover, this application has many uses in video games.

**Evaluation**: To evaluate our application, we have tested it on both test sequences of ICVL dataset. This dataset, as stated above, contains different poses of the hand and not only the counting poses. For this reason, we have used only 700 depth images of these test sequences which correspond to the poses indicating a clear number. Following this, we have annotated this sequence by the number indicated by the hand pose in each image to test our application results.

On this sequence, we attain a recognition rate of $80\%$, which indicates that our application managed to recognize the number indicated by $560$ poses and failed in $140$ cases. These failure cases are due to the annotation error of ICVL dataset which contain many incorrectly annotated samples. Indeed, Oberweger at al [18] evaluated the accuracy of ICVL annotation and found that about $36\%$ of the poses from the test set have an annotation error of at least $10mm$. This annotation error has a bad influence on the estimated pose and consequently on the accuracy of our application. Figure 9 shows some results of our application.

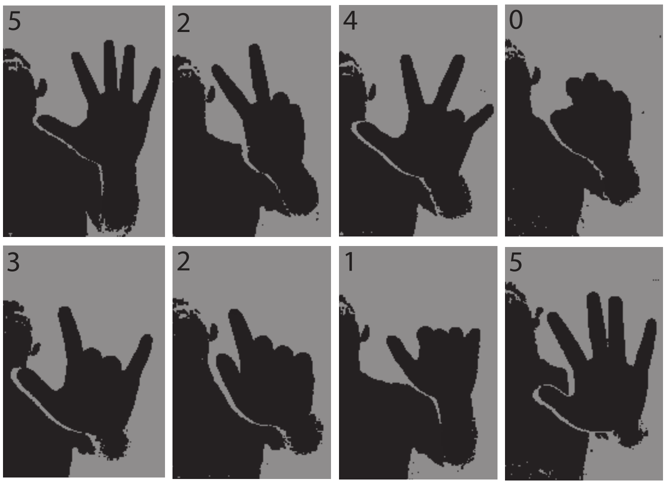


Fig. 9: Some illustrations of our application results on ICVL hand pose dataset. The number in the top-right of each image represents the estimated number indicated by the fingers.

## VIII. Conclusion and perspectives

In this work we have presented a regression method for hand pose estimation based on convolutional neural networks. This method relies only on a single depth map and does not require any complex image acquisition setup or special markers. Extensive experimental results demonstrate that our approach provides good performance. Moreover, we have exploited the success of this approach to develop an application of hand gesture recognition which attains 80% as recognition rate. As future work, we will first plan to enhance this approach by using a new CNN architecture and concentrate on improving the quality of the hand detector by introducing CNN in detection stage in order to further improve our detection precision.

## References

[1] A. Gustus, G. Stillfried, J. Visser, H. Jörntell, and P. van der Smagt, "Human hand modelling: kinematics, dynamics, applications," *Biological cybernetics*, vol. 106, no. 11-12, pp. 741–755, 2012.

[2] S. Sueda, A. Kaufman, and D. K. Pai, "Musculotendon simulation for hand animation," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 83.

[3] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *European Conference on Computer Vision*. Springer, 2012, pp. 852–863.

[4] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[5] S. Melax, L. Keselman, and S. Orsten, "Dynamics based 3d skeletal hand tracking," in *Proceedings of Graphics Interface 2013*. Canadian Information Processing Society, 2013, pp. 63–70.

[6] J. Park and Y.-L. Yoon, "Led-glove based interactions in multi-modal displays for teleconferencing," in *16th International Conference on Artificial Reality and Telexistence–Workshops (ICAT'06)*. IEEE, 2006, pp. 395–399.

[7] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1106–1113.

[8] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1862–1869.

[9] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2456–2463.

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

[11] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3456–3462.

[12] D. Tang, H. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3786–3793.

[13] P. Li, H. Ling, X. Li, and C. Liao, "3d hand pose estimation using randomized decision forest with segmentation index points," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 819–827.

[14] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3224–3231.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[16] S. Li, W. Zhang, and A. B. Chan, "Maximum-margin structured learning with deep networks for 3d human pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2848–2856.

[17] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[18] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," *Computer Vision Winter Workshop (CVWW)*, 2015.

[19] 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.

[20] 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.

[21] 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," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3593–3601, 2016.

[22] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, "A framework for articulated hand pose estimation and evaluation," in *Scandinavian Conference on Image Analysis*. Springer, 2015, pp. 41–52.

[23] Y. Wu, J. Y. Lin, and T. S. Huang, "Capturing natural hand articulation," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 426–432.

[24] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei, "Model-based deep hand pose estimation," *IJCAI*, 2016.