Depth Adaptive Deep Neural Network for Semantic Segmentation

Byeongkeun Kang, Yeejin Lee, and Truong Q. Nguyen, Fellow, IEEE

Abstract—In this work, we present the depth-adaptive deep neural network using a depth map for semantic segmentation. Typical deep neural networks receive inputs at the predetermined locations regardless of the distance from the camera. This fixed receptive field presents a challenge to generalize the features of objects at various distances in neural networks. Specifically, the predetermined receptive fields are too small at a short distance, and vice versa. To overcome this challenge, we develop a neural network which is able to adapt the receptive field not only for each layer but also for each neuron at spatial locations. To adjust the receptive field, we propose the adaptive perception neuron and the in-layer multiscale neuron. The adaptive perception neuron is to adjust the receptive field at each spatial location using the corresponding depth information. The in-layer multiscale neuron is to apply the different size of the receptive field at each feature space to learn features at multiple scales. By the combination of these neurons, we propose the three fully convolutional neural networks. We demonstrate the effectiveness of the proposed neural networks on the novel hand segmentation dataset for hand-object interaction and publicly available RGB-D dataset for semantic segmentation. The experimental results show that the proposed method outperforms the state-of-the-art methods without any additional layers or pre/post-processing.

I. INTRODUCTION

Depth perception, which is one of the crucial abilities in the human visual system, allows human to perceive the distance to an object and to understand the world in three dimensions. The human visual system uses the perceived depth information to robustly estimate the size/shape of objects in three dimensions. The three-dimensional information helps to better understand the objects and scenes along with other cues such as color information. Thus, depth information plays a key role in understanding the visual world.

As depth information is crucial for understanding the visual world, many researches have been conducted to acquire accurate depth information efficiently in both hardware systems and software systems. In hardware-based solutions, advanced depth sensors, such as Microsoft Kinect and light detection and ranging (LiDAR) sensors have been developed to capture better quality depth information with portability and low cost [1]–[3]. In software-based solutions, disparity estimation algorithms using single/multiple cameras have been explored to estimate accurate depth cues in shorter processing time [4]–[6]. Owing to these successes in both communities, depth information has been widely usable in many computer vision applications such as human pose estimation [7], [8], indoor scene understanding [9], and autonomous driving [10], [11].

After perceiving depth and/or color information, a machine processes the perceived information to understand the visual world. One of recently popular framework for learning visual information is the deep neural network, which is loosely inspired by the neurons of a human brain. As computing capability of machines has increased dramatically, deep neural networks have attained a huge improvement in understanding visual information and shown the state-of-the-art performance in many tasks such as image classification [12]–[14], object detection [15]–[19], and semantic segmentation [20]–[24].

Because of the importance of depth information and the improvements using deep neural networks, it has been speculated that incorporating depth information with neural networks has the advantage in understanding visual information. In most researches on deep neural networks using depth information, the depth map has been treated as an image equivalent input to the networks [25]–[29]. In such networks, the neurons share the predetermined receptive fields in a convolutional layer, which hinders the networks from learning common representations of an object. Considering that a pinhole camera captures an object at different distances, the camera captures the same object in different sizes, as demonstrated in Fig. 1. The illustration implies that a neural network can possibly learn/extract different features for the same object at various distances, yielding

Fig. 1. Illustration of the captured images and the proposed neural networks. The size of a captured object on the image plane varies with the distance from the object to the camera.

B. Kang, Y. Lee, and T. Q. Nguyen are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093 USA (e-mail: bkkang@ucsd.edu, yel031@ucsd.edu, tqn001@eng.ucsd.edu).

This work is supported in part by NSF grant IIS-1522125.
the confusions of recognizing objects. Hence, we propose the novel deep neural networks that learn common features for the same object by leveraging depth information (Section III-B). The proposed neural networks perceive the same region of the object regardless of the distance from the camera to each pixel as described in Fig. 2. This is achieved by the adaptive perception neuron in Section III-C1. The adaptive perception neuron adjusts the size of the receptive field at each spatial location corresponding to the distance from the camera. The adjustment requires a coefficient to decide the ideal correlation between the size and the distance. Since the optimal coefficient differs depending on the objects, it is beneficial to achieve the capability of utilizing multiple coefficients in a layer. This is achieved by the proposed in-layer multiscale neuron in Section III-C2. The in-layer multiscale neuron learns/extracts diversely scaled representations in a layer by applying a different size of the receptive field at each feature representation. The adjustment of the receptive field is applied using the sparse convolution (dilated convolution) as demonstrated in Fig. 5. In Section IV, we verify the effectiveness of the proposed method on two tasks: hand segmentation for hand-object interaction and indoor semantic segmentation. We collect new challenging dataset1 including hand-object interaction for hand segmentation and use publicly available NYUDv2 dataset [9] for indoor semantic segmentation.

In summary, the contributions of our work are as follows:

- We propose the depth-adaptive neural networks having the adaptive perception neurons and the in-layer multiscale neurons.
- We propose the adaptive perception neuron to learn/extract depth-adaptive representations.
- We propose the in-layer multiscale neuron to learn/extract variously scaled representations in a convolution layer.
- We verify the effectiveness of the proposed networks on the task of semantic segmentation.

II. RELATED WORKS

Deep neural networks using depth map. Most researches using depth maps in deep neural networks used a raw depth map as an image equivalent. For instance, a raw depth map was given as a direct input to the networks in hand pose estimation [25]–[27], human pose estimation [28], and fingerspelling recognition [29].

Alternatively, Gupta et al. proposed the geocentric embedding for a depth map to learn better representations in convolutional neural networks [30]. Specifically, the geocentric embedding encodes horizontal disparity, height above ground, and angle with gravity (HHA) for each pixel. The work showed that using the HHA encoded images, convolutional neural networks can learn better features for object detection and segmentation.

The networks we introduce are distinct from the works in [25]–[30]. First, our proposed method utilizes depth information in convolution layers rather than converting a raw depth map into a better representation in a preprocessing stage. In other words, our method does not require any additional preprocessing to manipulate the raw data. Second, our proposed method can take any type of input (e.g., color image, depth map, etc) to learn feature representations by giving the corresponding depth information as shown in Fig. 4.

Semantic segmentation. Long et al. proposed fully convolutional neural networks (FCN) for semantic segmentation by converting fully connected layers to convolution layers in the neural networks for image classification [20], [21]. The networks take an input of arbitrary size and produce the correspondingly-sized output.

Additional efforts have been made to improve the performance in [22]–[24], [31]. Zheng et al. proposed the convolutional neural networks that incorporate the strength of conditional random field (CRF)-based probabilistic graphical modeling. They formulated CRF as recurrent neural networks (RNN) and attached the RNN after FCN [31]. Chen et al. improved semantic segmentation using convolution with up-sampled filters, atrous spatial pyramid pooling, and fully connected CRF [22], [23]. Yu et al. proposed an additional context module to aggregate multiscale information without losing resolution [24].

Unlike other methods, our approach improves the performance of neural networks using depth information without adding additional layers. In addition, the proposed networks can incorporate any aforementioned additional layers for further improvement.

Hand segmentation for hand-object interaction. Most algorithms for hand segmentation segment hands using skin color in color images. Oikonomidis et al. and Romero et al. segmented hands by thresholding skin color in the hue-saturation-value (HSV) color space [32]–[34]. Wang et al. used the learned probabilistic model constructed from the color

---

1https://github.com/byeongkeun-kang/HOI-dataset
III. PROPOSED NETWORKS

The goal of this work is to learn depth-invariant representations in deep neural networks using depth information. To achieve this goal, we propose novel fully convolutional neural networks conceiving the adaptive perception neurons and the in-layer multiscale neurons as shown in Fig. 4. The adaptive perception neuron is proposed to adjust the receptive field using the depth information at each spatial location. The in-layer multiscale neuron is designed to learn features in different scales at each feature space (or channel) in a layer.

In Section III-A, we introduce key notations for networks. The overall architecture of the proposed neural network is developed in Section III-B. We provide the detailed explanation of the adaptive perception neuron in Section III-C1 and the in-layer multiscale neuron in Section III-C2. In Section III-D, the details of the training procedure for the proposed networks are derived. Finally, we provide the mathematical proof of the depth invariant property of the proposed networks in Section III-E.

### A. Notation

Let $X^\ell \in \mathbb{R}^{c_\ell \times h_\ell \times w_\ell}$ and $X^{\ell+1} \in \mathbb{R}^{c_{\ell+1} \times h_{\ell+1} \times w_{\ell+1}}$ be the matrices representing an input and an output of a certain layer $\ell$ (either convolution, pooling, softmax, or loss layer), where $c$, $h$, and $w$ denote the number of feature spaces (channels), height, and width, respectively. Also, let $D^\ell \in \mathbb{R}^{h_\ell \times w_\ell}$ be the (pooled) depth map in the convolution layer $\ell$ whose spatial resolution corresponds to the spatial resolution of the input $X^\ell$ (see Figs. 3 and 4). The size of $D^\ell$ is determined by pooling, convolution, and padding in the previous layers.

The output $X^{\ell+1}$ of the convolution layer $\ell$ is computed by convolving the input $X^\ell$ with a shared weight matrix $W^\ell \in \mathbb{R}^{c_{\ell+1} \times c_\ell \times k_h^\ell \times k_w^\ell}$ and by adding a bias vector $b^\ell \in \mathbb{R}^{c_{\ell+1}}$, where $k_h^\ell$ and $k_w^\ell$ denote the dimensions of kernels along the height and width directions. In a typical convolution layer, the
output $X^{t+1}_{t,m,n}$ of the $t$-th output feature space at the spatial location $(m, n)$ is computed as

$$X^{t+1}_{t,m,n} = f\left(\sum_r \sum_u \sum_v W^t_{r,u,v} X^t_{r,m+u,n+v} + b^t_r\right). \quad (1)$$

where $r \in [0, c^t - 1]$ and $t \in [0, c^{t+1} - 1]$ are the indices for the feature spaces of the input and the output, respectively, $u \in [-k_h/2, -k_h/2 + k_h - 1]$ and $v \in [-k_w/2, -k_w/2 + k_w - 1]$ are the indices for the weight matrix $W^t$ along the height and width direction, and $f(\cdot)$ is a transfer function (e.g. rectified linear unit (ReLU), etc.).

**B. Architecture**

We propose three networks (namely, Architecture A, B, C) according to the locations of applying the adaptive perception neuron and the in-layer multiscale neuron to the frontend module in [24]. The frontend module was selected as our baseline model since it showed better performance than the FCN [20], [21] without any additional layers. In the Architecture A, the proposed neurons are plugged into the first convolution layer to verify the improvement with minimum changes from the frontend module. In the Architecture B, the proposed neurons are applied to three convolution layers to demonstrate the improvement of applying the proposed neurons to the multiple layers. In the Architecture C, the proposed neurons are employed in all the convolution layers to achieve depth-invariance proved in Section III-E. The top figure in Fig. 4 shows the entire architecture of the proposed networks from the input layer to the output layer. The lower figure describes the detailed composition of the proposed Architecture A and B from the input layer to the fifth convolution layer.

We train the proposed neural networks by back-propagating the multinomial logistic loss $e_a$ while penalizing the increment of weights using the $L_2$ regularization (denote as $e_b$) [42]. Thus, the total loss $e$ is the sum of $e_a$ and $e_b$ (i.e. $e = e_a + \lambda e_b$), where $\lambda$ is the decay factor. To compute the multinomial logistic loss $e_a$, we apply the softmax function that transfers the input $X^t \in \mathbb{R}^{c_t \times h^t \times w^t}$ from the last convolution layer to the output $X^{t+1} \in \mathbb{R}^{c_t \times h^{t+1} \times w^{t+1}}$, where $c_t$ denotes the total number of classes. In softmax layer, the spatial resolutions of the input $X^t$ and the output $X^{t+1}$ are equivalent (i.e. $(h^{t+1}, w^{t+1}) = (h^t, w^t)$). The softmax output of the $r$-th feature space at the spatial location $(m, n)$ is defined as

$$X^{t+1}_{r,m,n} = \frac{e^{\exp(X^t_{r,m,n})}}{\sum_r e^{\exp(X^t_{r,m,n})}}. \quad (2)$$

The output $X^{t+1}_{r,m,n}$ is equivalent to the predicted probability of being the class $r$ at the spatial location $(m, n)$. Then, the
multinomial logistic loss $e_a$ is the weighted sum over the logistic outputs of $X^{ℓ+1}$:

$$e_a = -\frac{1}{h^ℓw^ℓ} \sum_r \sum_m \sum_n I(r = L_{m,n}) \log(X^{ℓ+1}_{r,m,n}), \quad (3)$$

where $I(\cdot)$ is an indicator function and $L \in \mathbb{R}^{h^{ℓ+1} \times w^{ℓ+1}}$ is a target class label matrix.

### C. Depth-adaptive Multiscale Convolution Layer

As observed in Fig. 1, an object appears to be different sizes in the image plane depending on its distance from the camera. The generalization performance of the trained networks using the depth-variant features may not be sufficiently good because learning a common representation is challenging. As such, it is necessary to learn depth-invariant features for neural networks in order to achieve better generalization performance. To this end, we propose the depth-adaptive multiscale convolution layer containing the adaptive perception neurons and the in-layer multiscale neurons. The adaptive perception neuron in Section III-C1 adjusts its receptive field to offset the change of the spatial size of objects on captured images. The receptive field adjusted by the adaptive perception neuron is clearly sub-optimal because the ideal correlation between the size and the distance varies over objects (e.g. due to different sizes). Hence, we develop the in-layer multiscale neuron in Section III-C2 that effectively controls over individual objects. The in-layer multiscale neuron extracts the diversely scaled depth-invariant features by tuning a parameter that determines sparsity at each feature representation.

Given a depth map as an input of the networks, unlike color images, the intensity (value) of an object on the depth map is dependent on the distance from the camera. This implies that the networks may learn intensity-variant features for the same object. To avoid this misguiding, we propose to employ depth difference (relative depth) in Section III-C3 as an input for the feature extraction.

1) **Adaptive perception neuron**: The proposed adaptive perception neuron determines its size of receptive field based on the depth information at each spatial location while other methods [22]–[24] used the predetermined receptive field in a convolution layer. Thus, the proposed networks having such adaptive perception neurons can apply different receptive field at each spatial location. Specifically, we increase the receptive field for objects at close distance and decrease it for objects at long distance to compensate the variation of objects’ size on captured images.

To determine the receptive field of each neuron, the depth map $D^ℓ$ is fed to the adaptive perception neuron. The size of the receptive field $S^ℓ \in \mathbb{R}^{h^ℓ \times w^ℓ}$ at a spatial location $(m,n)$ inversely increases to the depth from the camera $D^ℓ_{m,n}$, as follows:

$$S^ℓ_{r,m,n} \propto \frac{1}{D^ℓ_{m,n}}. \quad (4)$$

Applying the $S^ℓ$ for the convolution layer $ℓ$, the adaptive perception neuron takes different entries of the input $X^ℓ$ corresponding to the $S^ℓ_{r,m,n}$ as demonstrated in Figs. 3 and 5. Thus, the output in (1) is replaced by:

$$X^{ℓ+1}_{r,m,n} = f \left( \sum_r \sum_u \sum_v W^ℓ_{r,u,v} X^ℓ_{r,m+u,m+n+v} + b^ℓ \right). \quad (5)$$

2) **In-layer multiscale neuron**: Conventionally, learning/extracting features in various scales is advantageous in achieving higher accuracy by learning variant features. To learn features in multiple scales, the neural networks comprised of multiple neural networks were proposed in [25], known as the multiscale neural networks. In this type of neural networks, each constituting neural network takes an input in different resolution and learns features in various scales. However, these networks are structurally complex and require higher computational complexity. Thus, we propose the in-layer multiscale neuron that takes only an input and learns features with multiple scales in a neural network (see Fig. 6). The proposed in-layer multiscale neuron learns features at various scales by having a different parameters for the sparsity at each feature representation (channel).

The in-layer multiscale neuron determines sparsity at each feature space $r$ using the multiscale parameter $p^ℓ_r$, whereas the adaptive perception neuron in the previous section spatially

![Fig. 5. An example of applying the different sizes (sparsities) $S$ of the receptive field at each spatial location $(m,n)$. Suppose the indices of the matrix start from the top-left corner with $(1,1)$, and the kernel size is $3 \times 3$. The figure shows the cases of $S^ℓ_{r,1,3} = 1$, $S^ℓ_{r,5,8} = 2$, and $S^ℓ_{r,5,16} = 3$.](image)

![Fig. 6. The in-layer multiscale neuron. This neuron is able to learn features at different scales in a layer.](image)
determines the sparsity depending on the depth \( D_{m,n}^\ell \). The parameter \( p_r^\ell \) is determined as follows:
\[
p_r^\ell = \frac{s_r^\ell}{\prod_{r \in \mathcal{E}} z_r^\ell} \cdot \left[ \frac{1}{|\mathcal{T}|} |h^\ell_{w,1}| \sum_{d \in \mathcal{T}} \sum_{m} \sum_{n} D_{m,n}^d \right] \cdot q^\ell
\]
where \( s_r^\ell \) is the scaling factor for each feature space (channel) \( r \), \( z_r^\ell \) is the stride of pooling layers \( \ell' \in \mathcal{L} \) up to the current layer, \(|\mathcal{T}|\) represents the number of data in the training dataset \( \mathcal{T} \), and \( q^\ell \) is the dilation parameter from the ancestor architecture.

The \( p_r^\ell \) is interpreted as three factors: one is the scaling factor \( s_r^\ell \) with the mean \( \left[ \frac{1}{|\mathcal{T}|} |h^\ell_{w,1}| \sum_{d \in \mathcal{T}} \sum_{m} \sum_{n} D_{m,n}^d \right] \) of the depth maps in the training dataset, another is the factor \( 1/z_r^\ell \) regarding pooling layers, and the other is the dilation parameter \( q^\ell \) from the ancestor architecture. The scaling factor \( s_r^\ell \) with the mean of the depth determines different sparsities at each feature space considering the mean of the depth. Precise parameters for \( s_r^\ell \) is explained in Section IV. The term \( 1/z_r^\ell \) compensates the decrement of the spatial resolution of the feature map, caused by pooling layers. That is, the size of the receptive field is decreased as pooling layer reduces the spatial resolution. The term \( q^\ell \) is to retain the dilation parameter from the ancestor architecture.

Finally, the size of receptive field is determined combining the adaptive perception neuron and the in-layer multiscale neuron. The size \( S_{r,m,n}^\ell \) at a feature space \( r \) and a spatial location \((m,n)\) is as follows:
\[
S_{r,m,n}^\ell = \frac{p_r^\ell}{D_{m,n}^\ell},
\]
where denominator is contributed by the adaptive perception neuron, and numerator is from the in-layer multiscale neuron.

3) Depth difference: In practice, values on a depth map vary as the distance from the camera changes. For instance, objects at different distances are represented with different intensity levels. However, the relative distance between these objects is constant regardless of their distance from the camera [7, 8, 43]. Consequently, we instead use the relative depth to measure distance-independent depth in the first convolution layer. The relative depth is computed as the difference between the depth at the receptive field and the depth at the center location of the receptive field. Replacing a depth map by the relative depth map, (5) is rewritten as
\[
X_{t,m,n}^2 = f\left( \sum_r \sum_u \sum_v W_{t,r,u,v}^1 \left( X_{r,m+n,v}^{1} + s_{r,m,u,v} + s_{r,m,v} - X_{r,m,v}^{1} \right) + b_1^{1} \right).
\]

Although the input \( X^{1} \) to the networks is replaced by the relative depth \( \left( X_{r,m+n,v}^{1} + s_{r,m,u,v} + s_{r,m,v} - X_{r,m,v}^{1} \right) \), the size \( S_{r}^{1} \) of the receptive fields is computed using the raw depth map \( D^1 \).

D. Backpropagation

To train the proposed networks, the loss \( \epsilon \) is propagated backward and used to update the weights. The weights are updated by minimizing the loss using the gradient \( \partial \epsilon / \partial W^\ell \), where the gradient \( \partial \epsilon / \partial X^\ell \) is required to back-propagate to the lower layer. Considering the total loss \( \epsilon \) is the sum of the multinomial logistic loss \( e_a \) and the regularization loss \( e_v \), the gradient of \( \epsilon \) with respect to \( W^\ell \) is represented as
\[
\frac{\partial \epsilon}{\partial W^\ell} = \frac{\partial e_a}{\partial W^\ell} + \frac{\partial e_v}{\partial W^\ell},
\]
and this is rewritten by the chain rule [42], [44], [45], as follows:
\[
\frac{\partial \epsilon}{\partial W^\ell} = \frac{\partial e_a}{\partial X^{\ell+1}} \frac{\partial X^{\ell+1}}{\partial W^\ell} + \frac{\partial e_v}{\partial W^\ell}.
\]

For the shared weight \( W_{t,r,u,v}^\ell \), the gradient of (10) is expanded as
\[
\frac{\partial \epsilon}{\partial W_{t,r,u,v}^\ell} = \frac{\partial e_a}{\partial W_{t,r,u,v}^\ell} + \frac{\partial e_b}{\partial W_{t,r,u,v}^\ell} = \sum_m \sum_n \frac{\partial e_a}{\partial X_{t,m,n}^{\ell+1}} \frac{\partial X_{t,m,n}^{\ell+1}}{\partial W_{t,r,u,v}^\ell} + \lambda W_{t,r,u,v}^\ell.
\]

Recalling (5), since an output node has the input nodes \( (r,m) + S_{r,m,u,v} \) and \( n + S_{r,m,v} \) and the output node \( (t,m,n) \). The gradient of this specific connection is back-propagated as follows:
\[
\frac{\partial X_{t,m,n}^{\ell+1}}{\partial W_{t,r,u,v}^\ell} = X_{t,m,n}^{\ell} - X_{t,m,n}^{\ell+1} - X_{t,m,n}^{\ell} + b_2^{1}.
\]

To compute the first factor of \( e_a \), let’s first consider a specific connection between the input node \((r,m) + S_{r,m,u,v} \) and the output node \((t,m,n) \). The gradient of this specific connection is back-propagated as follows:
\[
\frac{\partial e_a}{\partial X_{t,m,n}^{\ell+1}} = \frac{\partial e_a}{\partial X_{t,m,n}^{\ell+1}} \frac{\partial X_{t,m,n}^{\ell+1}}{\partial X_{t,m,n}^{\ell}} = \frac{\partial e_a}{\partial X_{t,m,n}^{\ell}} \frac{\partial X_{t,m,n}^{\ell}}{\partial W_{t,r,u,v}^\ell}.
\]

In (13), the output node \( X_{t,m,n}^{\ell+1} \) is influenced by the multiple input nodes, then the gradient \( \partial e_a / \partial X^{\ell} \) is computed by the iterative accumulations over the feature spaces and the spatial locations, as summarized in Algorithm 1.

Finally, the weight matrix \( W^\ell \) is updated using the stochastic gradient descent algorithm with momentum [44] because we use small batch of training data to compute the gradients. At an iteration \( i \), suppose the current weight matrix is denoted as \( W^\ell,i \). Then, the weight matrix \( W^\ell,i+1 \) at the iteration \( i + 1 \) is updated considering the previous update and the computed gradient as follows:
\[
W^\ell,i+1 = W^\ell,i + \mu (W^\ell,i - W^\ell,i-1) - \gamma (\partial e / \partial W^\ell,i) \quad (14)
\]
where \( \mu \) and \( \gamma \) denote the momentum and the learning rate, respectively. The momentum \( \mu \) was chosen as 0.99, and the learning rate is explained in Section IV.
Algorithm 1 Gradient of loss with respect to input

Input: $\partial e_a/\partial X^{t+1}, W^t, S^t$
Output: $\partial e_a/\partial X^t$

initialize $\partial e_a/\partial X^t = 0$
for all $t, r, m, n, u, v$ do
$$\frac{\partial e_a}{\partial X_{r,m+S_{r,m,n,u,n}+S_{r,m,n,v}}} = \frac{\partial e_a}{\partial X_{t,m,n}} + W_{t,r,u,v}$$
end for

---

$X^t$ at distance $d$

1st Conv

Input at distance $d$

(a): Input at distance $d$

2nd Conv

- $x_1$
- $x_2$
- $x_3$
- $x_4$
- $x_5$
- $x_6$
- $x_7$
- $x_8$

3rd Conv

1st Conv

Input at distance $d/2$

(b): Input at distance $d/2$

---

E. Proof of Depth Invariance

In this section, we present the mathematical proof of the depth-invariance property of the proposed networks. We first simplify the convolution in (5) by considering a single channel one dimensional input and output. We, then, apply the proposed convolution to an input at different distances from the camera. By demonstrating that the outputs are equivalent regardless of the distances, we prove that the proposed convolution is depth-invariant.

Considering the network having a single channel (feature space), (5) is substituted as follows:

$$X_{m,n}^{t+1} = f\left(\sum_u \sum_v W_{u,v} X_{m+n,u+n,v} + b\right).$$

For the one-dimensional input, (15) is further simplified as

$$x_{m}^{t+1} = f\left(\sum_u w_{u} x_{m+u} + b\right).$$

Let’s first take an example in Fig. 7, showing the proposed convolution layers for the input at distance $d$ in Fig. 7(a) and at distance $d/2$ in Fig. 7(b). In the example, the size of kernel is set to 3, and the size $s$ of receptive field is 1 at distance $d$.

Then, the output $x_3^2$ of the first convolution layer in Fig. 7(a) is

$$x_3^2 = f\left(\sum_{u=-1}^1 w_{u} x_{5+u} + b\right) = f\left(w_{-1} x_4 + w_{0} x_5 + w_{1} x_6 + b\right),$$

and the output $x_3^3$ of the second convolution layer is

$$x_3^3 = f\left(w_{-1} x_4 + w_{0} x_5 + w_{1} x_6 + b\right) + f\left(w_{-1} x_3 + w_{0} x_4 + w_{1} x_5 + b\right) + f\left(w_{-1} x_2 + w_{0} x_3 + w_{1} x_4 + b\right).$$

In Fig. 7(b), the distance from the camera decreases to $d/2$, thus the size of the object on an image plane is doubled comparing to the size at $d$ (see Fig. 1). Let $\hat{x}$ denote the input at distance $d/2$ and suppose $\hat{x}_5$ corresponds to $x_5$. Then, $\hat{x}_{5+2v}$ is equivalent to $x_{5+v}$ for $\forall v \in \mathbb{Z}$ (e.g. $x_6 = \hat{x}_7$ for $v = 1$). Since its receptive field increases to 2 by the relation of (7), the output $\hat{x}_3^2$ of the first convolution layer is consequently equivalent to $x_3^2$:

$$\hat{x}_3^2 = f\left(\sum_{u=-1}^1 w_{u} \hat{x}_{5+2u} + b\right) = f\left(w_{-1} \hat{x}_{3} + w_{0} \hat{x}_{5} + w_{1} \hat{x}_{7} + b\right) = x_3^2.$$ 

and the output $\hat{x}_3^3$ of the second convolution layer is

$$\hat{x}_3^3 = f\left(w_{-1} \hat{x}_{2} + w_{0} \hat{x}_{3} + w_{1} \hat{x}_{4} + b\right) + f\left(w_{-1} \hat{x}_{1} + w_{0} \hat{x}_{2} + w_{1} \hat{x}_{3} + b\right) + f\left(w_{-1} \hat{x}_{0} + w_{0} \hat{x}_{1} + w_{1} \hat{x}_{2} + b\right).$$

We conclude from this simple example that the proposed convolution extracts depth-invariant activations.
From the fact that $\hat{x}_{m+n}^t$ is equivalent to $x_{m+n}^t$, the demonstration is generalized as
\[
\hat{X}_{t',m,n}^{\ell+1} = f\left(\sum_{r',u'} \sum_{v'} W_{t',r',u',v'} \hat{X}_{t',m+n, u', n+S_{m,n}^t}^\ell + b^t\right) \\
= f\left(\sum_{r',u'} \sum_{v'} W_{t',r',u',v'} \hat{X}_{t',m+(S_{m,n}^t/g) u', n+(S_{m,n}^t/g) v'}^\ell + b^t\right) \\
= f\left(\sum_{r',u'} \sum_{v'} W_{t',r',u',v'} X_{t',m+n, u', n+S_{m,n}^t}^\ell + b^t\right) \\
= X_{t',m,n}^{\ell+1},
\]
and are predicted to the class $j$, and $c_t$ be the total number of classes.

Pixel accuracy = \[\frac{\sum_i n_{ij}}{\sum_j \sum_i n_{ij}}\],
Mean accuracy = \[\frac{1}{c_t} \sum_i \left(\frac{n_{ii}}{\sum_j n_{ij}}\right)\],
Mean IoU = \[\frac{1}{c_t} \sum_i \left(\frac{n_{ij}}{\sum_j n_{ij} + \sum_j n_{ji} - n_{ii}}\right)\],
FW IoU = \[\frac{1}{\sum_i \sum_j n_{ij} \sum_i \left(\frac{n_{ij} n_{ii}}{\sum_j n_{ij} + \sum_j n_{ji} - n_{ii}}\right)}\],
Precision = \[\frac{n_{11}}{n_{11} + n_{01}}\],
Recall = \[\frac{n_{11}}{n_{11} + n_{10}}\],
\[F_1 = \frac{2n_{11}}{2n_{11} + n_{01} + n_{10}}\],
where for hand segmentation, class 1 is hand, and class 0 is others.

IV. EXPERIMENTS AND RESULTS

The proposed neural network was tested on two applications: hand segmentation for hand-object interaction and indoor semantic segmentation. The experimental results verify that the proposed neural network outperforms the frontend module in [24] without any additional layer or pre/post-processing.

For comparison, we report pixel-wise accuracy, mean accuracy, mean intersection over union (IoU), and frequency weighted (FW) IoU for both experiments. Additionally, for hand segmentation, we report precision, recall, and $F_1$ score.

A. Hand-Object Interaction (HOI)

Dataset. We collected a new dataset using Microsoft Kinect v2 since we were not able to find a publicly available dataset for hand-object interaction with pixel-wise annotation. The collected dataset consists of more than 9,175 pairs of depth maps and color images from 6 people (3 males and 3 females) interacting with 21 different objects. In addition, the dataset includes the cases of one hand and both hands in a scene. Ground truth was labeled by wearing a color glove during data collection and by finding the color of the glove on the color images.

To increase the variation of the dataset further (e.g. the distance from the camera to hands), 18,350 pairs of images were augmented by moving the camera closer/further to/from the scene as shown in Fig. 8. In total, the augmented dataset has 27,525 pairs of depth maps and ground truth labels. Indeed, the standard deviation of the augmented data increases to 725 relative to that of the collected dataset is 225, as evidenced in Fig. 9(a). The distances of the augmented dataset is distributed at more diverse distances as demonstrated in Fig. 9(b).

Among 27,525 pairs, we used 19,470 pairs for training, 2,706 pairs for validation, and 5,349 pairs for testing.

Experiments. All the models were initialized using the
TABLE I
THE QUANTITATIVE RESULTS OF THE HOI DATASET. THE SCORES ARE SCALED BY A FACTOR OF 100. BOLD FACE AND BLUE COLOR EMPHASIZE THE BEST PERFORMANCE FOR EACH INPUT AND FOR ENTIRE CASES, RESPECTIVELY.

| Input          | Method     | Precision | Recall | F\textsubscript{1} score | Pixel acc. | Mean acc. | FW IoU | Mean IoU |
|----------------|------------|-----------|--------|--------------------------|------------|-----------|--------|----------|
| Depth map      | Frontend [24] | 72.4      | 70.2   | 71.3                     | 99.0       | 84.9      | 98.2   | 77.2     |
|                | Proposed A | 78.3      | 78.7   | 78.5                     | 99.2       | 89.2      | 98.6   | 81.9     |
|                | Proposed B | 78.6      | 78.3   | 78.5                     | 99.2       | 89.0      | 98.6   | 81.9     |
|                | Proposed C | 79.7      | 82.5   | 81.1                     | 99.3       | 91.1      | 98.7   | 83.8     |
| HHA [30]       | Frontend [24] | 76.3      | 85.8   | 80.8                     | 99.3       | 92.7      | 98.7   | 83.5     |
|                | Proposed A | 81.4      | 84.4   | 82.9                     | 99.4       | 92.0      | 98.8   | 85.1     |
|                | Proposed B | 83.6      | 84.1   | 83.9                     | 99.4       | 91.9      | 98.9   | 85.8     |

Fig. 10. The qualitative comparison of the result for the HOI dataset. (a) Ground truth labels. (b) Results of the frontend module [24] for the input of a depth map. (c) Results of the proposed architecture B for the input of a depth map. (d) Results of the frontend module [24] for the input of an HHA encoded image [30]. (e) Results of the proposed architecture B for the input of an HHA encoded image [30]. The results and the ground truth labels are visualized on the depth maps with different color channels for better visualization.

For Architectures A and B, \( s_r^1 \) was set to \( \{1, 2, 4\} \) and \( s_r \) for other layers was set to 1 for the half of the feature spaces (channels) and 2 for the other features. For Architecture C, \( s_r^1 \) was set to \( \{1, 1.5, 2\} \) and \( s_r \) for other layers was set to \( \{0.75, 1, 1.25, 1.5\} \) for each quarter of the feature spaces in each convolution layer.

**Results.** The performances of the proposed methods and the comparing methods are tabulated in Table I for the inputs of the depth maps and the HHA encoded images [30]. The visual segmentation results are displayed in Fig. 10. The proposed neural network improves about 14% (depth maps) and 3% (HHA) in \( F_1 \) score relative to the baseline frontend model [24]. Moreover, the proposed Architecture C with the input of depth map achieves higher \( F_1 \) score and mean IoU.
TABLE II
THE QUANTITATIVE RESULTS OF THE NYUDv2 DATASET. THE SCORES ARE SC ALED BY A FACTOR OF 100. BOLD FACE AND BLUE COLOR EMPHASIZE THE BEST PERFORMANCE FOR EACH INPUT AND FOR ENTIRE CASES, RESPECTIVELY.

| Input | Method | Pixel acc. | Mean acc. | FW IoU | Mean IoU |
|-------|--------|------------|-----------|--------|---------|
| RGB   | Gupta et al. [30] | 60.3 | - | 47.0 | 28.6 |
|       | FCN-32s [20] | 60.0 | 42.2 | 43.9 | 29.2 |
|       | FCN-32s [21] | 61.8 | 44.7 | 46.0 | 31.6 |
|       | FCN-16s [21] | 62.3 | 45.1 | 46.8 | 32.0 |
|       | FCN-8s [21] | 62.1 | 46.1 | 47.2 | 32.4 |
|       | Frontend [24] | 62.1 | 45.8 | 46.6 | 32.3 |
|       | Proposed A | 63.4 | 46.7 | 48.0 | 32.9 |
|       | Proposed B | 63.5 | 47.0 | 48.2 | 32.9 |
|       | Proposed C | 63.7 | 47.2 | 48.3 | 33.3 |
| RGB-D | FCN-32s [20] | 61.5 | 42.4 | 45.5 | 30.5 |
|       | FCN-32s [21] | 62.1 | 44.8 | 46.3 | 31.7 |
|       | FCN-16s [21] | 62.3 | 45.4 | 46.8 | 32.2 |
|       | FCN-8s [21] | 62.7 | 46.0 | 47.4 | 32.5 |
|       | Frontend [24] | 62.1 | 46.2 | 46.8 | 32.5 |
|       | Proposed A | 63.5 | 46.8 | 48.1 | 33.0 |
|       | Proposed B | 63.2 | 47.0 | 48.1 | 32.9 |
|       | Proposed C | 63.8 | 47.1 | 48.3 | 33.3 |
| HHA [30] | FCN-32s [20] | 57.1 | 35.2 | 40.4 | 24.2 |
|       | FCN-32s [21] | 58.3 | 35.7 | 41.7 | 25.2 |
|       | FCN-16s [21] | 57.5 | 36.0 | 41.7 | 25.3 |
|       | FCN-8s [21] | 56.8 | 36.7 | 41.9 | 25.6 |
|       | Frontend [24] | 56.7 | 38.5 | 41.8 | 25.9 |
|       | Proposed A | 58.2 | 37.9 | 42.7 | 26.3 |
|       | Proposed B | 58.5 | 37.9 | 43.0 | 26.2 |
|       | Proposed C | 58.2 | 38.4 | 42.6 | 26.4 |
| RGB-HHA | FCN-32s [20] | 64.3 | 44.9 | 48.0 | 32.8 |
|       | FCN-32s [21] | 65.3 | 44.0 | 48.6 | 33.3 |
|       | FCN-16s [20] | 65.4 | 46.1 | 49.5 | 34.0 |
|       | FCN-16s [21] | 67.0 | 47.2 | 51.1 | 35.8 |
|       | FCN-8s [21] | 66.8 | 47.8 | 51.4 | 36.1 |
|       | Frontend [24] | 66.6 | 48.1 | 51.0 | 36.0 |
|       | Proposed A | 67.0 | 49.5 | 51.7 | 36.5 |
|       | Proposed B | 67.2 | 49.3 | 51.8 | 36.6 |
|       | Proposed C | 67.5 | 48.9 | 51.9 | 36.8 |

TABLE III
ABLATION STUDY OF SELECTING MULTISCALE PARAMETER \( s_r \). THE SCORES ARE SC ALED BY A FACTOR OF 100. BOLD FACE AND BLUE COLOR EMPHASIZE THE BEST PERFORMANCE.

| Multiscale parameter \( s_r \) | First conv. | Other conv. | Pixel acc. | Mean acc. | FW IoU | Mean IoU |
|-----------------------------|-------------|-------------|------------|-----------|--------|---------|
| \{1, 1.25, 1.5\} | 63.6 | 46.9 | 48.1 | 33.1 |
| \{1, 1.5, 2\} | 63.7 | 46.2 | 48.2 | 32.9 |
| \{1, 1.5, 2\} | 63.7 | 47.2 | 48.3 | 33.3 |
| \{1, 1.5, 2\} | 63.6 | 46.6 | 48.3 | 33.0 |
| \{1, 1.5, 2\} | 63.5 | 46.4 | 48.0 | 32.8 |
| \{1, 1.75, 2.5\} | 63.4 | 46.4 | 48.0 | 32.9 |

than the frontend module with the input of HHA encoded image. These results verify that the proposed networks improve segmentation performance without any additional layer or pre/post-processing.

B. Indoor Semantic Segmentation (NYUDv2)

**Dataset.** The NYUDv2 dataset consists of 1,449 pairs of RGB-D images including various indoor scenes with pixel-wise annotations [9]. The pixel-wise annotations were coalesced into 40 dominant object categories by Gupta et al. [47]. We experimented on this 40 classes problem with the standard separation [9], [47] of 795 training images and 654 testing images.

**Experiments.** In this experiment, we used the multinomial logistic loss without normalization during training. So, the normalization term \(1/\lambda \) was removed from (3). All the models were initialized using the VGG-16 model [13] trained using the ImageNet ILSVRC-2014 dataset [46] except for the input of RGB-HHA. Then, the models were fine-tuned using the NYUDv2 training dataset [9]. For the input of RGB-HHA, we initialized the model using the two fine-tuned models using NYUDv2 dataset (one model using RGB images and the other using HHA images). Then, we fine-tuned the model using the pair of RGB images and HHA images similar to [20], [21]. The initial base learning rate was selected by trying several learning rates \( \gamma \) with a factor of 10 such as \([10^{-5}, 10^{-10}, 10^{-11}, ...]\). The initial base learning rate was selected as \(10^{-12}\) for the input of RGB-HHA and \(10^{-10}\) for the other inputs. The decay factor \(\lambda\) of the weight matrix is chosen as 0.0005.

The models used in the experiments were selected based on the mean IoU score. During training, we computed the mean IoU score at every 1,000 iterations for the input of RGB-HHA and at every 2,000 iterations for the other inputs. The training was performed based on the same criteria in the HOI dataset (see Section IV-A. Experiments).

The scaling parameter \( s_t \) was set to \{1, 1.5, 2\} for all the architectures. For Architecture B, \( s_r \) for other layers was set to 1 for the half of the feature spaces (channels) and 2 for the other features. For Architecture C, \( s_r \) for other layers was set to \{0.5, 0.75, 1.0, 1.25\} for color images and depth maps and \{0.75, 1.0, 1.25, 1.5\} for HHA images.

**Results.** We adopted the experimental settings in [20], [21]. We considered the inputs of an RGB image, the concatenated image of an RGB image and a depth map (early fusion), and an HHA encoded image [30]. We also experimented combining the scores from an RGB image and from an HHA encoded image [30] at the last layer (late fusion). Table II and Fig. 11 show the quantitative results and the qualitative results. The proposed method achieves the improvements without any additional layers or pre/post-processing.

**Analysis.** We experimentally analyze the effects of multiscale parameters \( s_t \) in Table III. The table shows that the proposed method outperforms other methods using the parameters in the reasonable ranges. Moreover, we show the convergence curve for the frontend module [24] and the proposed Architecture C in Fig. 12. The average loss is computed using the losses from 100 iterations. The graph shows that the proposed method converges slightly faster than the frontend module. Lastly, to compare efficiency, we measure the inference time for the input of RGB in Table IV using a
Fig. 11. The qualitative comparison of the result for the NYUDv2 dataset. The odd rows show the results of the frontend module [24], and the even rows show the results of the proposed architecture B. The results and the ground truth labels are visualized on the color images for better visualization.

Fig. 12. The comparison of convergence curves between frontend module [24] and the proposed Architecture C for the input of RGB.

TABLE IV

| Method      | Processing time (ms) |
|-------------|----------------------|
| FCN-32s [21] | 206                  |
| FCN-16s [21] | 201                  |
| FCN-8s [21]  | 201                  |
| Frontend [24] | 417                  |
| Proposed A   | 467                  |
| Proposed B   | 470                  |
| Proposed C   | 481                  |

machine with Intel i7-4790K CPU and Nvidia Tesla K40c.
V. CONCLUSION

In this paper, we presented the novel fully convolutional neural networks that adjust the receptive field using depth information to learn/extract depth-invariant feature representations. In the proposed neural networks, we introduced the adaptive perception neuron and the in-layer multiscale neuron. The proposed neural networks were applied to hand-segmentation for hand-object interaction and indoor semantic segmentation. The experimental results demonstrate that the proposed neural networks improve the accuracy of segmentation without any additional layers or pre/post-processing.

REFERENCES

[1] Z. Zhang, “Microsoft kinect sensor and its effect,” IEEE MultiMedia, vol. 19, no. 2, pp. 4–10, Feb 2012.
[2] M. Adams and P. Probert, “The interpretation of phase and intensity data from amcg light detection sensors for reliable ranging,” The International Journal of Robotics Research, vol. 15, no. 5, pp. 441–458, 1996.
[3] B. Schwarz, “Lidar: Mapping the world in 3d,” Nature Photonics, vol. 4, pp. 429–430, 2010.
[4] Z. Lee and T. Q. Nguyen, “Multi-resolution disparity processing and fusion for large high-resolution stereo image,” IEEE Transactions on Multimedia, vol. 17, no. 6, pp. 792–805, June 2015.
[5] Z. Lee and T. Nguyen, “Multi-array camera disparity enhancement,” IEEE Transactions on Multimedia, vol. 16, no. 8, pp. 2168–2177, Dec 2014.
[6] R. Ranftl, V. Vincet, Q. Chen, and V. Koltun, “Dense monocular depth estimation in complex dynamic scenes,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
[7] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, Jun. 2011, pp. 1297–1304.
[8] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, “Efficient human pose estimation from single depth images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 12, pp. 2821–2840, Dec 2013.
[9] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in ECCV, 2012.
[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” International Journal of Robotics Research (IJRR), 2013.
[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, 2012, pp. 1097–1105.
[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR, 2015.
[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun 2016.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[16] R. Girshick, “Fast r-cnn,” in 2015 IEEE International Conference on Computer Vision (ICCV), 2015.
[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Advances in Neural Information Processing Systems (NIPS), 2015.
[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 779–788.
[19] S. Tripathi, Z. Lipton, S. Belongie, and T. Nguyen, “Context matters: Refining object detection in video with recurrent neural networks,” in Proceedings of the British Machine Vision Conference (BMVC), 2016.
[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.
[21] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, pp. 1–1, 2016.
[22] L. C. Shen, Q. Pan, A. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected cnns,” in ICLR, 2015.
[23] L. C. Shen, Q. Pan, A. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected cnns,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PP, pp. 1–1, 2017.
[24] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in ICLR, 2016.
[25] J. Tompson, M. Stein, and K. Perlin, “Real-time continuous pose recovery of human hands using convolutional networks,” ACM Transactions on Graphics, 2014.
[26] 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 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
[27] A. Sinha, C. Choi, and K. Ramani, “Deephand: Robust hand pose estimation by completing a matrix imputed with deep features,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
[28] K. Wang, S. Zhai, H. Cheng, X. Liang, and L. Lin, “Human pose estimation from depth images via inference embedded multi-task learning,” in Proceedings of the 2016 ACM on Multimedia Conference, ser. MM ’16. New York, NY, USA: ACM, 2016, pp. 1227–1236.
[29] B. Kang, S. Tripathi, and T. Nguyen, “Real-time sign language fingerspelling recognition using convolutional neural networks from depth map,” in Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on, 2015.
[30] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from rgb-d images for object detection and segmentation,” in Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, Proceedings, Part VII, 2014, pp. 345–360.
[31] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vincet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, “Conditional random fields as recurrent neural networks,” in 2015 IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 1529–1537.
[32] I. Oikonomidis, N. Kyriazis, and A. Argyros, “Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints,” in Computer Vision – ICCV, 2011 IEEE International Conference on, Nov. 2011, pp. 2088–2095.
[33] J. Romero, H. Kjellstrom, and D. Kragic, “Hands in action: real-time 3d reconstruction of hands in interaction with objects,” in Robotics and Automation (ICRA), 2010 IEEE International Conference on, May 2010, pp. 458–463.
[34] J. Romero, H. KjellstroMc, C. H. Ek, and D. Kragic, “Non-parametric hand pose estimation with object context,” Image Vision Comput., vol. 31, no. 8, pp. 555–564, Aug. 2013.
[35] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai, “Video-based hand manipulation capture through compositive motion control,” ACM Trans. Graph., vol. 32, no. 4, pp. 43:1–43:14, Jul. 2013.
[36] J. A. Palmer, K. Kreutz-Delgado, and S. Makeig, “Super-gaussian mixture source model for ica,” in Independent Component Analysis and Blind Signal Separation, Mar. 2006, pp. 854–861.
[37] M. J. Jones and J. M. Rehg, “Statistical color models with application to skin detection,” International Journal of Computer Vision, vol. 46, no. 1, pp. 59–76, Nov. 2000.
[38] D. Zhou, Y. Shi, and T. Kanade, “View-based hand manipulation capture for compositive motion control,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[39] Y. Jeon and J. Kim, “Active convolution: Learning the shape of convolution for image classification,” CoRR, vol. abs/1703.09076, 2017. [Online]. Available: http://arxiv.org/abs/1703.09076
[40] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” CoRR, vol. abs/1703.06211, 2017. [Online]. Available: http://arxiv.org/abs/1703.06211
[41] G. Strang and T. Nguyen, Wavelets and filter banks. SIAM, 1996.
[42] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
[43] B. Kang, K.-H. Tan, N. Jiang, H.-S. Tai, D. Tretter, and T. Nguyen, “Hand segmentation for hand-object interaction from depth map,” in 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Nov 2017.

[44] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[45] T. Apostol, Mathematical Analysis, ser. Addison-Wesley series in mathematics. Addison-Wesley, 1974.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[47] S. Gupta, P. Arbelaz, and J. Malik, “Perceptual organization and recognition of indoor scenes from rgb-d images,” in 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013, pp. 564–571.