Hand segmentation with structured convolutional learning

[1,2]Natalia Neverova [1,2]Christian Wolf [3]Graham W. Taylor [4]Florian Nebout

[1]Université de Lyon, CNRS, France firstname.surname@liris.cnrs.fr
[2]INSA-Lyon, LIRIS, UMR5205, F-69621
[3]University of Guelph, Canada gwtaylor@uoguelph.ca
[4]Awabot, Lyon, France florian.nebout@awabot.com

Abstract. The availability of cheap and effective depth sensors has resulted in recent advances in human pose estimation and tracking. Detailed estimation of hand pose, however, remains a challenge since fingers are often occluded and may only represent just a few pixels. Moreover, labelled data is difficult to obtain. We propose a deep learning based-approach for hand pose estimation, targeting gesture recognition, that requires very little labelled data. It leverages both unlabeled data and synthetic data from renderings. The key to making it work is to integrate structural information not into the model architecture, which would slow down inference, but into the training objective. We show that adding unlabelled real-world samples significantly improves results compared to a purely supervised setting.

1 Introduction

We present a new method for hand pose estimation from depth images targeting gesture recognition. We focus on *intentional* gestures, bearing *communicative function*, with an emphasis on the recognition of fine-grained and smooth gestures. In particular, our aim is to go beyond classification and estimate hand motion with great accuracy, allowing for richer human-computer interactions. From an application perspective, this ensures a tight coupling between users and objects of interest, for instance a cursor, or a manipulated virtual object.

Most existing methods are based on the estimation of articulated pose of the body or the hands. Made possible by the introduction of cheap and reliable consumer depth sensors, these representations have revolutionized the field, effectively putting visual recognition into the hands of non-specialists in vision. While the robust estimation of body joints is now possible in real time, at least in controlled settings [1], most systems provide coarse body joints and do not give the positions of individual joints of the hand. This restricts applications to full body motion, whereas fine-grained interaction requires hand pose estimation.

Estimating hand pose is inherently more difficult than full body pose. Given the low resolution of current sensors and the strong noise they produce, fingers usually are composed of only a few pixels. To make matters worse, individual fingers frequently are not discernible. Existing work is mostly applicable in situations where the hands are close to the sensor, which is suited to applications

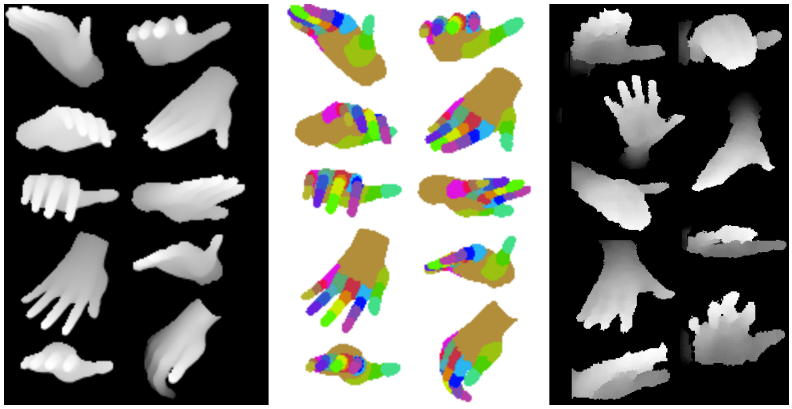

Fig. 1. Our model learns from labeled synthetic data and unlabeled real data. Left: Synthetic depth input images. Middle: Ground truth for synthetic input. Right: Real depth data.

where the user interacts with a computer he or she is close to or sitting in front of. However, applications in domotics, mobile robotics and games, to cite a few, do not fall into this category.

One way to address these issues is to add strong spatial and structural priors or hard constraints, for instance by fitting an articulated model to the data [2]. The computational complexity of the underlying optimization is a disadvantage of this solution. Machine learning has a preponderant role, where most solutions estimate joint positions through an intermediate representation based on hand part segmentation [3–5], or direct regression of finger joint positions [6], or both [5]. However, methods based on learning are hungry for labelled training data, which is difficult to come by. This is especially true for hand pose estimation, where manual annotation of both joint positions and finger parts is difficult.

Existing work deals with this issue by including priors, for instance by including structural information combining the learned predictions with graphical models [7]. Transductive learning is an alternative, where a few labelled samples are combined with a large amount of unlabelled samples, and a transfer function between both sets is learned [5].

In this work we tackle this problem in a structured machine learning setting by segmenting hands into parts. In a semi-supervised context, a deep convolutional network is trained on a large dataset of labelled synthetic hand gestures rendered as depth images, as well as unlabelled real depth images acquired with a consumer depth sensor (Fig. 1). The main contribution of this paper is the way in which structural information is treated in the learning process. Instead of combining a learned prediction model with a structured model, for instance a graphical model, we integrate structural information directly into the learning algorithm aiming to improve the prediction model. As as consequence, at test time, pixels are classified independently, keeping the advantages of low computational complexity and retaining the ability to parallelize.

The information integrated into the training procedure is related to prior information which can be assumed on a segmented image. Our method is based on two contributions. Firstly, contextual information is extracted from local context in unlabelled samples through a learned trained on synthetic labelled examples. Secondly, similar to body part maps, ground truth hand part segmentation maps are assumed to feature a single connected region per part, which commonly holds ignoring rare exceptions due to severe self occlusion. We show that this information can be formalized and leveraged to integrate unlabelled samples into the training set in a semi-supervised setting.

Although we focus on the application of hand pose estimation, the proposed method is also applicable to other problems involving segmentation of entities, for example, objects, people, scenes into parts.

2 Related work

Hand pose estimation — the majority of approaches to pose estimation are conventionally assigned to one of two groups: 3D model or appearance-based methods. One of the most notable recent works in the spirit of 3D modeling and inverse rendering [8] is based on pixelwise comparison of rendered and observed depth maps. Liang et al. [2] apply the iterative closest point (ICP) algorithm to hand pose reconstruction and 3D fingertip localization under spatial and temporal constraints. Qian et al [9] proposed a hybrid method for realtime hand tracking using a simple hand model consisting of a number of spheres. Appearance-based methods typically include global matching of observed visual inputs with pose instances from training data. Athitsos et al. [10], for example, use a synthetic dataset featuring a great number of hand shape prototypes and viewpoints and perform matching by calculating approximate directed Chamfer distances between observed and synthetic edge images.

A seminal paper on pixel-based body segmentation with random decision forests [1] gave birth to a whole group of follow-up works including several adaptations for hand segmentation [3–5]. Deep learning of representations has been applied to body part or hand part segmentation [11, 7, 12]. In [13], hand part segmentation using random forests is combined with deep convolutional networks for gesture recognition.

A great amount of ad-hoc methods have been proposed specifically for hand-gesture recognition in narrow contexts. Most of them rely on hand detection, tracking, and gesture recognition based on global hand shape descriptors such as contours, silhouettes, fingertip positions, palm centers, number of visible fingers, etc. [14, 15]. Similar descriptors have been proposed for depth and RGBD data [16]. Sridhar et al [17] proposed a hybrid model for hand tracking using a multi-view RGB camera setup combined with a depth sensor.

Segmentation, structural information and context models — there has been renewed interest lately in semantic segmentation or semantic labelling methods. In these tasks, taking into account structural (contextual) information in addition to local appearance information is primordial. Contextual informa-

tion often allows the model to disambiguate decisions where local information is not discriminative enough. In principle, increasing the support region of a learning machine can increase the amount of context taken into account for the decision. In practice, this places all of the burden on the classifier, which needs to learn a highly complex prediction model from a limited amount of training data, most frequently leading to poor performance. An elegant alternative is to apply multi-scale approaches. Farabet et al. [18] propose a multi-scale convolutional net for scene parsing which naturally integrates local and global context.

Structural information can be directly modeled through spatial relationships, which are frequently formulated as probabilistic graphical models like Markov Random Fields, Conditional Random Fields [19] or Bayesian networks. Inference in these models amounts to solving combinatorial problems, which in the case of high-level contextual information are often intractable in the general case. In comparison, classical feed forward networks are causal models, which allow fast inference but which are inherently ill-suited to deal with cyclic dependencies. Architectures which permit feedback connections such as Deep Boltzmann Machines [20] are difficult to train and do not scale to high-resolution images.

An alternative way to approximate cyclic dependencies with causal models has recently gained attention in computer vision. In *auto-context models* [21], cascades of classifiers are trained, where each classifier takes as input the output of the preceding classifier and eventual intermediate representations. Follow-up work recast this task as a graphical model in which inference is performed through message passing [22, 23]. In [24], a sequential schema is proposed using randomized decision forests to incorporate semantic context to guide the classifier. In [25], auto-context is integrated into a single random forest, where pixels are classified breadth first, and each level can use decisions of previous levels of neighboring pixels. In [11], spatial neighborhood relationships between labels, as they are available in body part and part segmentation problems, are integrated into convolutional networks as prior knowledge.

Our proposed method is similar to auto-context in that the output of a first classifier is fed into a second classifier, which is learned to integrate the context of the pixel to predict. However, whereas auto-context models aim at repairing the errors of individual classifiers, our model uses contextual information to extract structural information from unlabelled images in a semi-supervised setting. The ability to seamlessly combine unlabeled examples and labeled examples is an important motivation behind the field of deep learning of representations [26, 27]. Also relevant to our work are paradigms in which the test task is similar but different than the training task – transfer learning and domain adaptation [28]. Although we do not explicitly treat domain adaptation, in our task we exploit synthetic data at training time but not at test time.

3 Semi-supervised structured learning

The pixelwise hand segmentation in this work is performed with a classifier which we call a *direct learner*. It operates frame-by-frame, ignoring inter-frame

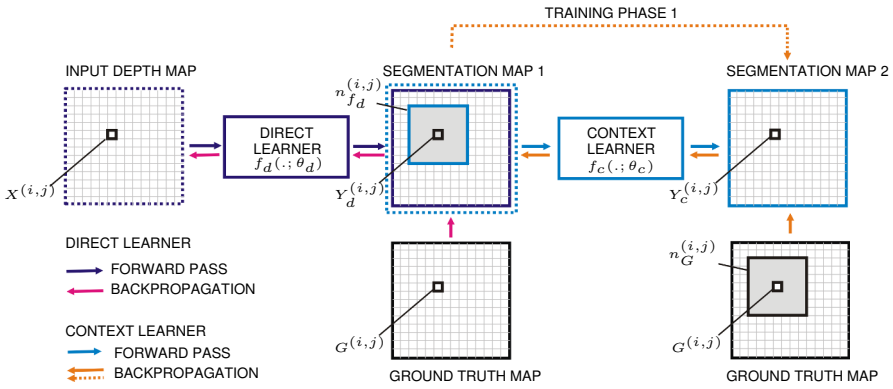

Fig. 2. The two learning pathways involving a direct learner f_d and a context learner f_d. The context learner operates on punctured neighborhood maps $n^{(i,j)}$, where the (to be predicted) middle pixel is missing.

temporal dependencies. Given the constraint of real-time performance typical for this class of applications, we keep the test architecture as simple as possible and focus mainly on developing an effective training procedure to learn meaningful data representations that are robust to noise typical of real-world data.

The training data consists of input depth maps: $X = \{X^{(i)}\}$, $i = 1 \dots |X|$. From this whole set of maps, L maps are synthetic and annotated, denoted as $X_L = \{X^{(i)}\}$, where $i = 1 \dots L$, $L<|X|$. The subset of unlabeled real images is denoted as $X_U = \{X^{(i)}\}$, where $i > L$. The set of ground truth segmentation maps corresponding to the labelled set is denoted as $G = \{G^{(i)}\}$, where $i = 1 \dots L$, $L \leq |X|$. No ground truth is available for $X^{(i)}$, $i>L$. Pixels in the different maps are indexed using a linear index j: $X^{(i,j)}$ denotes the j^{th} pixel of the i^{th} depth map.

The synthetic frames are rendered using a deformable 3D hand model. A large variety of viewpoints and hand poses (typical for interactive interfaces) is obtained under manually defined physical and physiological constraints. For the sake of generalization, and also keeping in mind that manually labeling data is tedious and impractical, we do not assume that ground-truth segmentation of real data is available in any amount. Instead, in parallel with supervised learning on annotated synthetic images, we use unlabeled frames for global optimization during training time.

Optimization criteria are based on, first, consistency of each predicted pixel class with its local neighborhood on the output segmentation map and, second, global compactness and homogeneity of the predicted hand segments.

For the first task, at training time we introduce an additional classification path, called the *context learner* [29], which is trained to predict each pixel's class given labels of its local neighborhood. Both the direct and context learners are first pre-trained simultaneously in a purely supervised way on the synthetic images (see Fig. 2). The pre-training of the context learner is divided into two steps.

First, ground truth label maps are used as the training input. After convergence of the direct learner, its output is used instead for input to the context learner, and the context learner is fine-tuned to cope with realistic output segmentation maps.

Let us introduce notation that will be used to formalize the training algorithm: $f_d(\theta_d)\colon X^{(i,j)} \rightarrow Y_d^{(i,j)}$ denotes a *direct learner* with parameters θ_d mapping each pixel $j = 1 \dots M$ in a depth map i (having depth value $X^{(i,j)}$) into a corresponding pixel of an output segmentation map Y_d with elements $Y_d^{(i,j)}$, having one of possible $k = 1 \dots K$ values corresponding to hand segments.

$f_c(\theta_c)\colon N^{(i,j)} \rightarrow Y_c^{(i,j)}$ denotes a *context learner* predicting the pixel label $Y_c^{(i,j)}$ from its neighborhood $N^{(i,j)}$ on the same segmentation map. The neighborhood is *punctured*, i.e. the center pixel to be predicted, j, is missing. As we have already mentioned, this classifier is first pre-trained on the ground truth images $G^{(i)}$ followed by fine-tuning on the segmentation maps produced by the direct learner f_d.

The probabilistic setting of our training algorithm makes it convient to introduce a difference between a random variable and its realization. In the following and as usual, uppercase letters denote random variables or fields of random variables and lower case letters denote realizations of values of random variables or of fields of random values. Realizations of random fields X, Y_c, Y_d and G defined above are thus denoted as x, y_c, y_d and g. Furthermore, $P(X{=}x)$ will be abbreviated as $P(x)$ when it is convenient. Fig. 2 illustrates the configuration of the two learners f_d and f_c and the corresponding notation.

The loss function used for training the direct learner f_d in conjunction with the context learner f_c is composed of three terms whose activation depends on whether or not ground truth labels for the given training image are available:

$$Q = Q_{sd} + Q_{sc} + Q_u, \tag{1}$$

where Q_{sd} is responsible for training of the direct learner, Q_{sc} corresponds to the context learner (both) and Q_u is an unsupervised term serving as a natural regularizer.

During training, annotated and unannotated examples are considered interchangeably, starting with labeled data (supervised learning) followed by an increase in the amount of unlabeled samples (unsupervised domain adaptation).

3.1 Supervised terms

Supervised terms classically link the predicted class of each pixel to the ground truth hand part label. The first term Q_{sd} is formulated as vanilla negative log-likelihood (NLL) for pixel-wise classification with the direct learner f_d.

$$Q_{sd}(\theta_d \,|X_L\,) = -\sum_{i=0}^{L}\sum_{j=0}^{M} \log P\left(Y_d^{(i,j)} = g^{(i,j)} \,\middle|\, x^{(i,j)}; \theta_d\right) \tag{2}$$

Recall here, that $Y_d^{(i,j)}$ is the output of the direct learner and $g^{(i,j)}$ is a ground truth label.

The second term Q_{sc} is also a negative log-likelihood loss for pixelwise classification, this time using the context learner f_c. Learning of θ_c proceeds in two steps. First, ground truth segmentation maps are fed to the learner, denoted as $n_G^{(i,j)}$, minimizing NLL:

$$Q_{sc}^{(1)}(\theta_c \,|G) = -\sum_{i=0}^{L}\sum_{j=0}^{M} \log P\left(Y_c^{(i,j)} = g^{(i,j)} \,\middle|\, N^{(i,j)} = n_G^{(i,j)}; \theta_c\right) \tag{3}$$

After convergence, in a second phase, segmentation maps produced by the direct learner are fed into the context learner, denoted as $n_{f_d}^{(i,j)}$.

$$Q_{sc}^{(2)}(\theta_c \,|f_d(X_L)) = -\sum_{i=0}^{L}\sum_{j=0}^{M} \log P\left(Y_c^{(i,j)} = g^{(i,j)} \,\middle|\, N^{(i,j)} = n_{f_d}^{(i,j)}; \theta_c\right) \tag{4}$$

Parameters θ_d are kept fixed during this step, and depth maps are not used during both steps.

3.2 Unsupervised terms

In the unsupervised case, ground truth labels are not available. Instead, the loss function measures structural properties of the predicted segmentation at two different scales, either on context (at a neighborhood level), or globally on the full image. The estimated properties are then related to individual pixelwise loss.

$$Q_u = f(Q_{loc}, Q_{glb}) \tag{5}$$

Local structure Q_{loc} is a term capturing local structure. It favors predictions which are consistent with other predictions in a local neighborhood. In particular, it favors predictions where the direct learner agrees with the context learner (recall that the context learner is not learned in this phase).

This term is formulated as a conditional negative likelihood loss. For each given pixel, if both classifiers f_d and f_c (the latter one operates on the output of the former one) agree on the same label, this pixel is used to update parameters θ_d of the direct learner and the error is minimized using the classical NLL scenario treating the predicted label as corresponding to the ground truth:

$$Q_{loc}(\theta_d \,|X_U\,) = -\sum_{j=0}^{M} \mathbb{I}_{y_d^{(i,j)} = y_c^{(i,j)}} \log P\left(Y_d = y_c^{(i,j)} \,\middle|\, x^{(i,j)}; \theta_d, \theta_c\right), \tag{6}$$

where $\mathbb{I}_\omega$=1 if ω holds and 0 else. In this case the parameters θ_c of the context learner remain unchanged. The indicator function is non-smooth with respect to the parameters. For backpropagation, we treat it as a constant once both segmentation maps are computed.

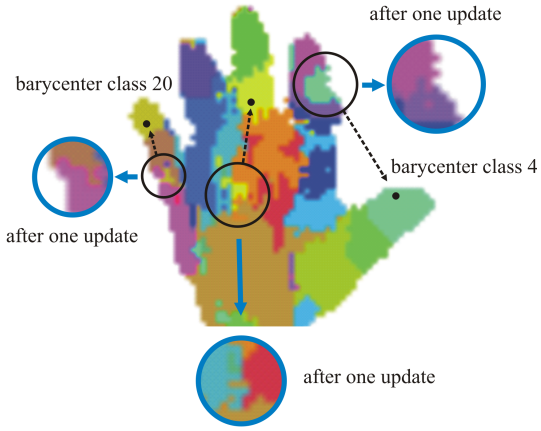

Fig. 3. Global structural information which can be extracted from a segmented image even if ground truth labels are **not** available. Small circles with thin black borders contain segmented pixels which are far away from the principal pixel mass of the given hand part, indicated by the barycenter of the hand part. The global unsupervised loss term Q_{glb} punishes these results. Large circles with thick blue borders show the same content after a single network parameter update using Q_{glb}.

Global structure Q_{glb} is a term capturing global structure. It favors predictions which fit into global image statistics and penalizes the ones which do not by changing parameters in the direction of a more probable class. Technically, this term aims on minimizing variance (in terms of pixel coordinates) of each hand segment. Fig. 3 illustrates the intuitive understanding of this terms. Ground truth labels are **not** available for the real images dealt with in this part, but there is intrinsic structural information which can be extracted from a segmented image, and which is related to strong priors we can impose on the segmentation map. In particular, unlike general segmentation problems, body and hand part segmentation maps contain a single connected region per hand part label (ignoring cases of strong partial self-occlusion, which are extremely rare). In Fig. 3, small circles with thin black borders contain segmented pixels which are not connected to the principal region of the given hand part, indicated by the barycenter of the pixels of this hand part. The global unsupervised loss term Q_{glb} punishes these results. Large circles with thick blue borders show the same content after a single network parameter update using Q_{glb}.

We formalize this concept as follows. For each class k present in the output map Y_d, barycentric coordinates of the corresponding segment are calculated:

$$\mathbf{R}_k = \frac{\sum\limits_{j:Y_d^{(i,j)}=k} P(y_d^{(i,j)}|x^{(i,j)})\mathbf{r}^{(i,j)}}{\sum\limits_{j:Y_d^{(i,j)}=k} P(y_d^{(i,j)}|x^{(i,j)})}, \tag{7}$$

where pixel coordinates, in vector form, are denoted as $\mathbf{r}^{(i,j)}$.

If $\left|\mathbf{r}^{(i,j)} - \mathbf{R}_k\right| > \tau$, i.e. the pixel (i,j) is close enough to its barycenter (τ is estimated from the labelled synthetic data), then the pixel is considered as correctly classified and used to update parameters of the direct learner θ_d. The loss function term for one pixel (i,j) is given as follows:

$$Q^{+}_{glb}(\theta_d \left| y_d^{(i,j)} \right.) = -F^{(i)}_{y_d} \log P\left(Y_d = y_d^{(i,j)} \left| x^{(i,j)}, \theta_d, \theta_c \right.\right), \tag{8}$$

where $F_k^{(i)}$ is a weight related to the size of class components:

$$F_k^{(i)} = |\{j : Y_d^{(i,j)} = k\}|^{-\alpha} \tag{9}$$

and $\alpha > 0$ is a gain parameter. In the opposite case, when $\left|\mathbf{r}^{(i,j)} - \mathbf{R}_k\right| \leq \tau$, the current prediction is penalized and the class γ corresponding to the closest segment in the given distance τ is promoted:

$$Q^{-}_{glb}(\theta_d \left| y_d^{(i,j)} \right.) = -F^{(i)}_{\gamma} \log P\left(Y_d = \gamma \left| x^{(i,j)}, \theta_d, \theta_c \right.\right), \tag{10}$$

where

$$\gamma = \operatorname{argmin}\left(|\mathbf{r}^{(i,j)} - \mathbf{R_k}|\right). \tag{11}$$

This formulation is related to the k-means algorithm. However, data points in our setting are embedded in two spaces: the space spanned by the network outputs (or, alternatively, feature space), and the 2D geometric space of the part positions. Assigning cluster centers requires therefore optimizing multiple criteria and distances in heterogeneous spaces. Other clustering costs could be also adopted.

Integrating local and global structure Local structure and global structure are fused emphasizing agreement between both terms. In particular, activation of the penalizing global term (which favors parameters pushing a pixel away from currently predicted class) is confirmed by a similar structural information captured by the local term ($Q_{loc} = 0$):

$$Q_u = \beta_{loc} Q_{loc} + \beta_{glb} \begin{cases} Q^{+}_{glb} & \text{if } \left|\mathbf{r}^{(i,j)} - \mathbf{R}_k\right| \leq \tau, \\ Q^{-}_{glb} & \text{if } \left|\mathbf{r}^{(i,j)} - \mathbf{R}_k\right| > \tau \text{ and } Q_{loc} = 0, \\ 0 & \text{else} \end{cases} \tag{12}$$

where β_{loc} and β_{glb} are weights.

Combining the two terms, Q_{loc} and Q_{glb}, is essential as they are acting in an adversarial way. The local term alone leads to convergence to a trivial solution when all pixels in the image are assigned to the same class by both classifiers. The global term favors multi-segment structure composed of homogeneous regions, while exact shapes of the segments may be distorted as to not satisfy the desirability of of compactness. The two terms acting together, as well as mixing the labeled and unlabeled data, allow the classifier to find a balanced solution.

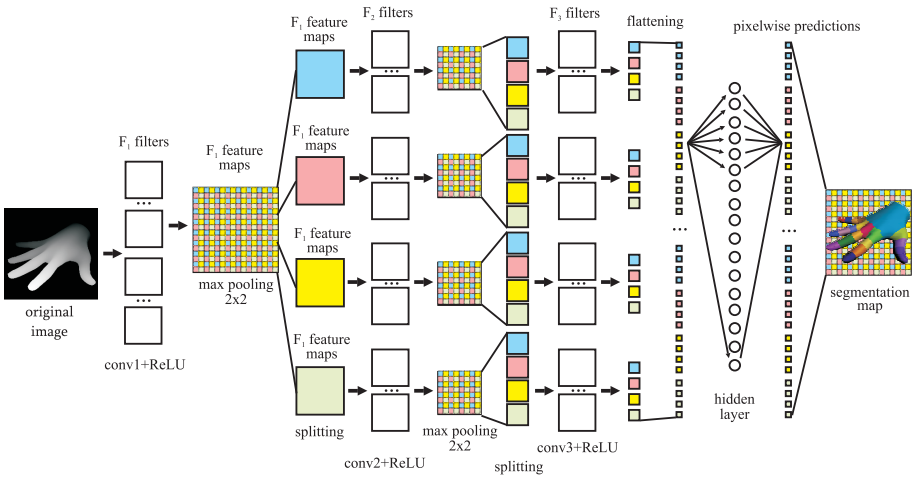

Fig. 4. The proposed deep convolutional architecture of a single learner.

4 Architecture

The direct and context learners are based on a convolutional network architecture and have the same general structure (see Fig. 4) including three consecutive convolutional layers F_1, F_2 and F_3 with rectified linear activation units (ReLU). Layers F_1 and F_2 are followed by 2×2 max pooling and reduction.

As opposed to most existing methods for scene labeling, instead of randomly sampling pixels (or patches), training is performed image-wise, i.e. all pixels from the given image are provided to the classifier at once and each pixel gets assigned with an output class label based on information extracted from its neighborhood.

Applying of the convolutional classifier with pooling/reduction layers to an image in the traditional way would lead to loss in resolution by a factor of 4 (in the given configuration). On the other hand, simply not reducing the image resolution will prevent higher layers from learning higher level features, as the size of the filter support does not grow with respect to the image content. To avoid this dilemma, we employ specifically designed splitting functions originally proposed for image scanning in [30] and further exploited in *OverFeat* networks [31]. Intuitively speaking, each map at a given resolution is reduced to four different maps of lower resolution using max pooling. The amount of elements is preserved, but the resolution of each map is lower compared to the maps of previous layers.

In more detail, let us consider the output of the first convolutional layer F_1 of the network. Once the output feature maps are obtained, 4 virtual extended copies of them are created by zero padding with 1) one column on the left, 2) one column on the right, 3) one row on top, 4) one row in the bottom. Therefore, each copy will contain the original feature map but shifted in 4 different directions. On the next step, we apply max pooling (2×2 with stride 2×2) to each of the extended maps producing 4 low-resolution maps. By introducing the shifts,

pixels from all extended maps combined together can reconstruct the original feature map as if max pooling with stride 1×1 had been applied. This operation allows the network to preserve results of all computations for each pixel on each step and, at the same time, perform the necessary reduction, resulting in a significantly speed up during training and testing.

After pooling, convolutions of the following layer F_2 are applied to all 4 low-resolution maps separately (but in parallel). The same procedure is repeated after the second convolutional layer F_2, where each of 4 branches is split again into 4, producing 16 parallel pathways overall. If necessary, the algorithm can be extended to an arbitrary number of layers and employed each time when reduction is required.

All outputs of the third convolutional layer F_3 are flattened and classified with an MLP, producing a label for each pixel. Finally, the labels are rearranged to form the output segmentation map corresponding to the original image.

The direct and the context learners have the same architecture with the only difference that the middle parts of the first layer filters of the context learner are removed. It helps to prevent the network from converging to a trivial solution where a pixel label is produced by directly reproducing its input. This is especially important on the initial training stage, when the context learner is trained on ground truth segmentation maps.

5 Experiments

For this project, we have created a vast collection of about 60000 synthetic training samples, including both normalized 8 bit depth maps and ground truth segmentations with resolution 640×640 pixels. Hand shapes, proportions, poses and orientations are generated with a random set of parameters. Each pose is captured from 5 different camera view points sampled randomly for each frame. An additional set of 6000 images is used for validation and testing.

The unlabeled part of the training set consists of 3000 images captured with a depth sensor. To evaluate the algorithm performance on the real-world data, we have manually annotated 50 test samples. An example of a ground truth segmentation map is shown in Fig. 5, where the hand is divided into 20 segment classes. Background pixels are set to 0 and assigned a class label of 0.

For training, synthetic depth maps are downsampled by a factor of 4 (to imitate real world conditions), and cropped. As a result, the network input is of size 80×80 pixels.

Both direct and context learners have 3 convolutional layers F_1, F_2 and F_3 with 16, 32, 48 filters respectively, where each filter is of size 7×7. Max pooling 2×2 is performed after the first two convolutional layers. The hidden layer is composed of 150 units. Thus, each pixel is classified based on its receptive field of size 46×46. In the context learner, the middle parts of size 3×3 of the first layer filters are removed. The learning rate is initially set to 0.1. Unsupervised learning parameters are set to $\beta_{loc} = 0.1$, $\beta_{glb} = 1.2$, and $\alpha = 0.5$.

Table 1. Performance of networks trained with different objective functions.

Loss function	Training data	Test data	Accuracy	Average per class
Q_{sd} (supervised baseline)	synth.	synth.	85.90%	78.50%
		real	47.15%	34.98%
$Q_{sd} + Q_{loc} + Q_{glb}$ (semi-supervised, ours)	all	synth.	85.49%	78.31%
		real	**50.50**%	**43.25**%

Table 2. Perf. improvement on a real image after updating parameters using different supervised and unsupervised terms, estimated as an average over 50 real images.

Terms	Q_{loc}	${Q_{glb}}^+$	${Q_{glb}}^+ + {Q_{glb}}^-$	$Q_{loc} + {Q_{glb}}^+ + {Q_{glb}}^-$	Q_{sd}
Requires labels	no	no	no	no	yes
Gain in % points	+0.60	+0.36	+0.41	**+0.82**	+16.05

The current pure CPU implementation of the entire pipeline runs at 436 ms per frame (with a potential speed-up by a factor of 20-30 [13] on GPU).

The training procedure is started with purely supervised learning by back-propagation which proceeds until 50% of the synthetic training data is seen by the network. From this moment on, we replace 10% of the training set with unlabeled real world samples. A single training image consisting of $80 \times 80 = 6400$ pixel samples is used for each step of gradient descent.

Comparative performance of classifiers trained by including and excluding different unsupervised terms of the loss function is summarized in Table 1. Exploiting unlabeled real data for unsupervised training and network regularization has proven to be generally beneficial, especially for reconstruction of small segments (such as finger parts), which leads to a significant increase of average per-class accuracy. The bar plot on the Fig. 5 demonstrates significant improvement of recognition rates for almost all classes except for the first, base "palm" class which can be seen as a background for a hand image against which finger segments are usually detected. Therefore, this reflects the fact that more confident detection in the case of semi-supervised training comes together with a certain increase in the amount of false positives.

Table 2 illustrates the impact of one update of the network parameters for different loss functions on the performance on a given image which was used for computing the gradients. We note that a combination of two competitive unsupervised terms (local and global) produces a more balanced solution than the same terms separately.

The local term alone forces the network to favor the most statistically probable class (i.e. the "palm" in our settings), while the global one on its own tends to shift boundaries between regions producing segmentation maps similar to a Voronoi diagram. In the latter case, the number of cells is typically defined by

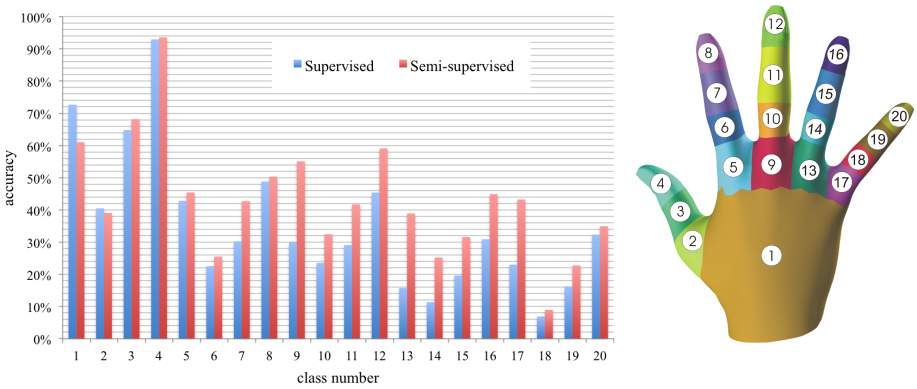

Fig. 5. Left: average accuracy per class obtained with the supervised method (in blue) and with semi-supervised structured learning (in red); Right: labeling of hand segments.

an initial guess of the network on the given image and is unlikely to be changed by global unsupervised learning alone.

Therefore we stress the importance of pre-training the direct and context learners on the synthetic data in order to be capable of producing structurally representative initial predictions for the unlabeled data. Furthermore, the frequency of supervised gradient updates during the final training stage should remain significant to prevent training from diverging.

Output segmentation maps produced by the proposed method are shown in Fig. 6. Fig. 7 shows several "problematic" images where the baseline supervised network performs poorly. Our algorithm is capable of finding regions which would not have otherwise been reconstructed and often leads to more consistent predictions and a reduction in the amount of noise in the segmentation maps.

6 Conclusion

We have proposed a novel method for part segmentation based on convolutional learning of representations. Unlike most deep learning methods which require large amounts of labeled data, we do not assume that ground-truth segmentation of real data is available. Our main contribution is a training method which exploits i) context learning; and ii) unsupervised learning of local and global structure, balancing a prior for large homogenous regions with pixel-wise accuracy. By integrating structural information into learning rather than the model architecture, we retain the advantages of very fast test-time processing and the ability to parallelize. The use of synthetic data is an important part of our training strategy. A potential area of further improvement is domain adaptation from synthetic to real images.

Acknowledgments. This work was partially funded by French grants **Interabot**, call *Investissements d'Avenir*, and **SoLStiCe** (ANR-13-BS02-0002-01), call *ANR blanc*.

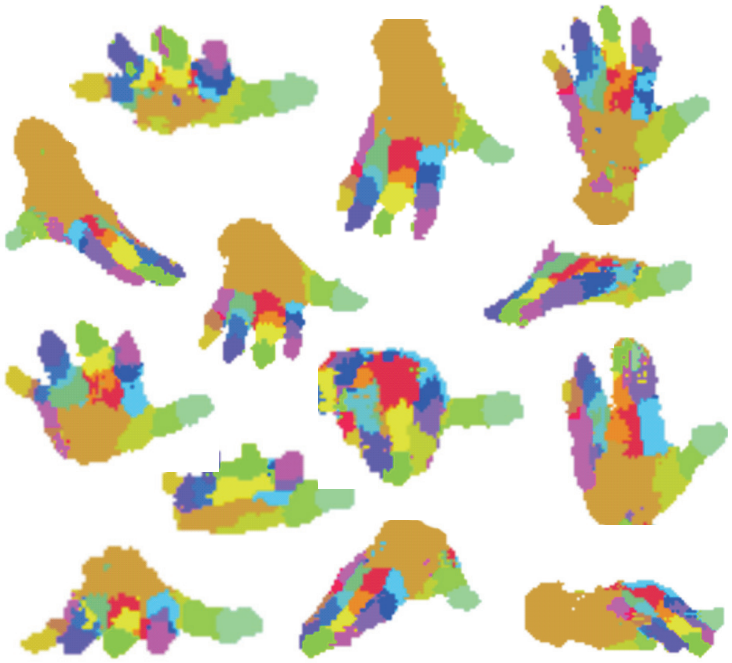

Fig. 6. Output segmentation maps produced by the semi-supervised network for real-world images.

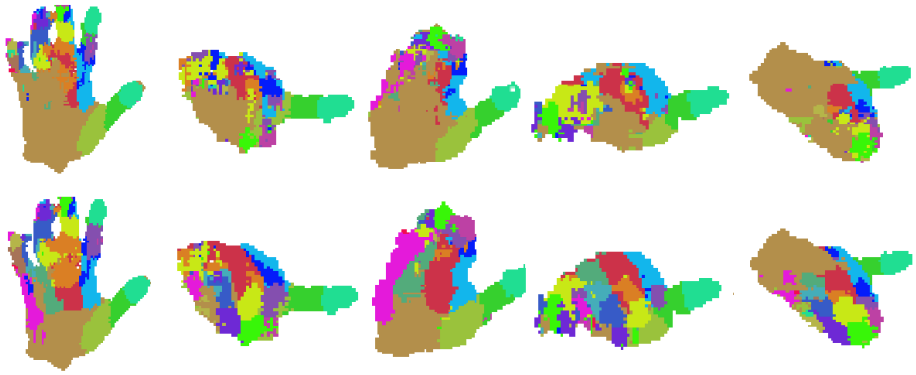

Fig. 7. Challenging examples. Top row: examples where the baseline method has difficulty in segmentation. Bottom row: the results of our proposed algorithm on the same examples.

References

1. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. CVPR (2011) 1297–1304
2. Liang, H., , Yuan, J., Thalmann, D., Zhang, Z.: Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization. In: The Visual Computer. Volume 29. (2013) 837–848
3. Keskin, C., Kiraç, F., Kara, Y., Akarun, L.: Real time hand pose estimation using depth sensors. In: ICCV Workshop on Consumer Depth Cameras, IEEE (2011)
4. Półrola, M.a., Wojciechowski, A.: Real-time hand pose estimation using classifiers. In: Computer Vision and Graphics. Volume 7594., Springer (2012) 573–580
5. Tang, D., Yu, T., Kim, T.K.: Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV. (2013)
6. Shotton, J.: Conditional regression forests for human pose estimation. In: CVPR). (2012) 3394–3401
7. Jain, A., Tompson, J., Andriluka, M., Taylor, G., Bregler, C.: Learning human pose estimation features with convolutional networks. In: ICLR. (2014)
8. Oikonomidis, I., Kyriazis, N., Argyros, A.: Efficient model-based 3D tracking of hand articulations using Kinect. In: BMVC. (2011) 101.1–101.11
9. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and Robust Hand Tracking from Depth. In: CVPR. (2014)
10. Athitsos, V., Liu, Z., Wu, Y., Yuan, J.: Estimating 3D hand pose from a cluttered image. In: CVPR, IEEE (2003)
11. Jiu, M., Wolf, C., Taylor, G., Baskurt, A.: Human body part estimation from depth images via spatially-constrained deep learning. Pattern Recognition Letters (2014)
12. Toshev, A., Szegedy, C.: DeepPose: Human pose estimation via deep neural networks. In: CVPR. (2014)
13. Tompson, J., Stein, M., LeCun, Y., Perlin, K.: Real time continuous pose recovery of human hands using convolutional networks. In: SIGGRAPH/ACM-ToG. (2014)
14. Stergiopoulou, E., Papamarkos, N.: Hand gesture recognition using a neural network shape fitting technique. Engineering Applications of Artificial Intelligence **22** (2009) 1141–1158
15. Malima, A., Özgür, E., Çetin, M.: A fast algorithm for vision-based hand gesture recognition for robot control. IEEE 14th Conference on Signal Processing and Communications Applications (2006)
16. Mateo, C.M., Gil, P., Corrales, J.A., Puente, S.T., Torres, F.: RGBD Human-Hand recognition for the Interaction with Robot-Hand. In: IROS. (2012)
17. Sridhar, S., Oulasvirta, A., Theobalt, C.: Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. In: ICCV. (2013)
18. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Scene parsing with multiscale feature learning, purity trees, and optimal covers. In: ICML. (2012)
19. Tighe, J., Lazebnik, S.: Superparsing: Scalable nonparametric image parsing with superpixels. In: ECCV. (2010) 352–365
20. Salakhutdinov, R., Hinton, G.E.: Deep boltzmann machines. In: International Conference on Artificial Intelligence and Statistics. (2009) 448–455
21. Tu, Z.: Auto-context and its application to high-level vision tasks. In: CVPR. (2008)

22. S.Ross, D.Munoz, M.Hebert, J.A.Bagnell: Learningmessage-passing inference machines for structured prediction. In: CVPR. (2011) 27372744
23. Shapovalov, R., Vetrov, D., Kohli, P.: Spatial inference machines. In: CVPR. (2013) 29852992
24. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR. (2008) 1–8
25. Montillo, A., Shotton, J., Winn, J., Iglesias, J., Metaxas, D., Criminisi, A.: Entangled decision forests and their application for semantic segmentation of CT images. In: Information Processing in Medical Imaging. (2011) 184196
26. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013) 1798–1828
27. Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade. Springer (2012) 639–655
28. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. Unsupervised and Transfer Learning Challenges in Machine Learning, Volume 7 (2012) 19
29. Fromont, E., Emonet, R., Kekeç, T., Trémeau, A., Wolf, C.: Contextually Constrained Deep Networks for Scene Labeling. In: BMVC. (2014)
30. Giusti, A., Ciresan, D.C., Masci, J., Gambardella, L.M., Schmidhuber, J.: Fast image scanning with deep max-pooling convolutional neural networks. In: ICIP. (2013)
31. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014)