



# Human body part estimation from depth images via spatially-constrained deep learning

Mingyuan Jiu, Christian Wolf, Graham W. Taylor, Atilla Baskurt

## ► To cite this version:

Mingyuan Jiu, Christian Wolf, Graham W. Taylor, Atilla Baskurt. Human body part estimation from depth images via spatially-constrained deep learning. Pattern Recognition Letters, 2014, 1, 50, pp.122-129. hal-01269994

**HAL Id: hal-01269994**

**<https://hal.science/hal-01269994>**

Submitted on 27 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Human body part estimation from depth images via spatially-constrained deep learning

Mingyuan Jiu<sup>a,b,\*</sup>, Christian Wolf<sup>a,b</sup>, Graham Taylor<sup>c</sup>, Atilla Baskurt<sup>a,b</sup>

<sup>a</sup>*Université de Lyon, CNRS*

<sup>b</sup>*INSA-Lyon, LIRIS, UMR5205, Villeurbanne, France*

<sup>c</sup>*School of Engineering, University of Guelph, Guelph, Ontario, Canada*

---

## Abstract

Object recognition, human pose estimation and scene recognition are applications which are frequently solved through a decomposition into a collection of parts. The resulting local representation has significant advantages, especially in the case of occlusions and when the subject is non-rigid. Detection and recognition require modelling the appearance of the different object parts as well as their spatial layout. This representation has been particularly successful in body part estimation from depth images.

Integrating the spatial layout of parts may require the minimization of complex energy functions. This is prohibitive in most real world applications and therefore often omitted. However, ignoring the spatial layout puts all the burden on the classifier, whose only available information is local appearance. We propose a new method to integrate spatial layout into parts classification without costly pairwise terms during testing. Spatial relationships are exploited in the training algorithm, but not during testing. As with competing methods, the proposed method classifies pixels independently, which makes real-time processing possible. We show that training a classifier with spatial relationships increases generalization performance when compared to classical training minimizing classification error on the training set. We present an application to human body part estimation from depth images.

*Keywords:* Segmentation, Spatial layout, Deep learning, Convolutional networks, Depth images

---

## 1. Introduction

Many computer vision problems can be solved in part by an initial step which segments an image, a video, or their constituent objects into regions, which are called parts in this context. The segmentation algorithm typically considers local appearance information, and frequently also models the spatial relationships between different parts. Unfortunately, considering these relationships within the segmentation process mostly amounts to solving constraint satisfaction problems or performing inference in a graphical model with cycles and a non sub-modular energy function, both of which are intractable in the

general case. In this paper we address the problem of efficiently modeling spatial relationships without the need for solving complex combinatorial problems.

This general class of problems corresponds to various applications in computer vision. For example, pose estimation methods are also often naturally solved through a decomposition into body parts. A preliminary pixel classification step segments the object into body parts, from which joint positions can be estimated in a second step. The well-known system described in (Shotton et al., 2011), installed on millions of gaming consoles and taking as input images from consumer depth cameras, completely ignores spatial relationships between the object parts and puts all of the classification burden on the pixel-wise random forest classifier. To achieve its state-of-the-art level of performance, it required training on an extremely large training set of  $2 \cdot 10^9$  examples.

A similar problem occurs in tasks where joint object recognition and segmentation is required. Layout

---

\*Corresponding author, Tel.: +33-4-72-43-63-72

Email addresses: mingyuan.jiu@liris.cnrs.fr (Mingyuan Jiu), christian.wolf@liris.cnrs.fr (Christian Wolf), gwtaylor@uoguelph.ca (Graham Taylor), atilla.baskurt@liris.cnrs.fr (Atilla Baskurt)

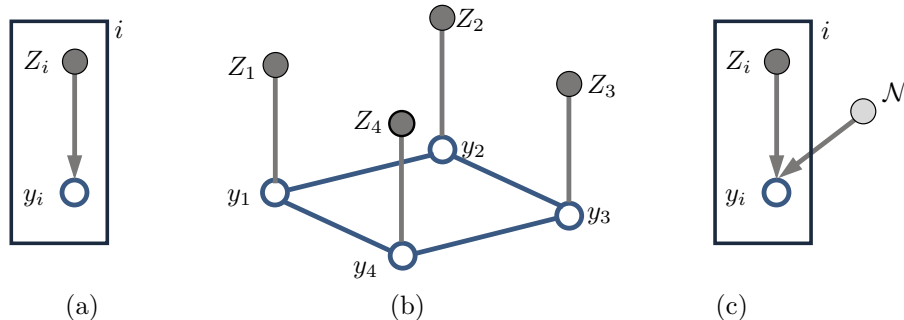


Figure 1: Different ways to include spatial layout, or not, into learning part labels  $y_i$  from features  $Z_i$  for pixels  $i$ : (a) pixelwise independent classification, where spatial layout information is not taken into account; (b) A Markov random field with pairwise terms coding spatial constraints; (c) our method: pixelwise independent classification including spatial constraints  $\mathcal{N}$ .

CRFs and their extensions model the object as a collection of local parts (patches or even individual pixels), which are related through an energy function (Winn and Shotton, 2006). However, unlike pictorial structures for object recognition (Felzenszwalb and Huttenlocher, 2005; Felzenszwalb et al., 2010), the energy function here contains cycles which makes minimization more complex, for instance through graph cuts techniques. Furthermore, the large number of labels makes the expansion move-type algorithms inefficient (Kolmogorov and Zabih, 2004).

In all cases, the underlying discrete optimization problem is very similar: an energy function encoding the spatial relationships in pairwise terms needs to be minimized. A typical dependency graph for this kind of problem is shown in Figure 1b: unary terms relate each label  $y_i$  to a feature vector  $Z_i$ , and pairwise terms encode prior knowledge on the possible configurations of neighboring labels  $y_i$  and  $y_j$ .

In this paper, we propose a method which segments an image or an object into parts through pixelwise classification, integrating the spatial layout of the part labels. Like methods which ignore the spatial layout, it is extremely fast as no additional step needs to be added to pixelwise classification and no energy minimization is necessary during testing. The (slight) additional computational load only concerns learning at an offline stage. The goal is not to compete with methods based on energy minimization, which is impossible through pixelwise classification only. Instead, we aim to improve the performance of pixelwise classification by using all of the available information during learning.

In each of the problems that we consider, the labels we aim to predict have spatial structure. Our proposed method uses an energy function to enforce a spatial con-

sistency in learned features which reflects the spatial layout of labels. Unlike combinatorial methods, our energy function is minimized during training (i.e. while learning features) but is unused at test time. It is based on two main assumptions. First, different high-dimensional features with the the same label are embedded to a lower-dimensional manifold which preserves the original semantic meaning. Second is our belief that greater loss should be incurred when misclassification occurs between features coming from non-neighbor labels than features coming from the same or neighboring labels. In other words, the geometry of learned features, to some extent, reflects the spatial layout of labels. We will show that this new loss function increases the classification performance of the learned prediction model.

Another way of looking at our contribution is to interpret it as a way of structuring the prediction model of a learning machine. Classical techniques working on data represented in a vector space, like neural networks, SVMs, randomized decision trees, boosted classifiers, etc., are, in principle, capable of learning arbitrary complex decision functions if the underlying prediction model (architecture) is complex enough. In reality, the available amount of training data and computational resources available limit the complexity which can be learned. In most cases, only a limited amount of data is available with respect to the complexity of the problem. It is therefore often useful to impose some structure on the model. We already mentioned structured models based on energy minimization and their computational disadvantages. Manifold learning is another technique which assumes that the data, although represented in a high dimensional space, is distributed according to a lower dimensional manifold in that space. Semi-supervised learning uses a large amount of addi-

tional training data, which is unlabeled, to help the learning machine better infer the structure of the decision function. In this work, we propose to use prior knowledge in the form of the spatial layout of the labels to add structure to the task of learning the decision function.

Another key aspect of our technique is end-to-end *feature learning*. The dominant methodology in computer vision, though changing in light of recent successes (Krizhevsky et al., 2012), is to extract engineered features such as SIFT (Lowe, 2004) or HOG (Dalal and Triggs, 2005), pool responses, and learn a classifier from this fixed representation. Our objective is to apply learning at all stages of the pipeline, from pixels to labels. However, compared to contemporary Deep Learning approaches, we learn representations that are informed by the spatial structure of the part labels instead of simply their identity.

This paper proposes several contributions:

- We propose spatial learning for unsupervised *pre-training* of deep convolutional networks (i.e. learning all but the topmost layer). The features learned by spatial pre-training are more informative than classical features, as experiments with a non-spatial LR classifier show.
- We propose a framework which integrates spatial part layout into supervised learning of deep neural networks. We show that the gain of spatial learning is indeed obtained at extremely small cost: it improves the performance of the classifier with absolute *zero* increase in computational complexity of testing.
- To the best of our knowledge, we are the first to apply Deep Learning to the problem of body parts segmentation from depth images, obtaining promising results.

## 2. Related work

Our framework proposes to learn a feature extractor from raw data combining the spatial layout of labels, in order to produce a better decision function for segmentation. It is equivalent to learning a mapping function from high-dimensional space to a low-dimensional manifold space, therefore there is some partial overlap with dimensionality reduction.

Unsupervised approaches for learning a mapping capturing global structure are well-known; most notably

Principal Component Analysis (PCA) (Jolliffe, 1986) and Multi-Dimensional Scaling (Cox and Cox., 1994). However, our aim is to embed based on the spatial layout of part labels, so we restrict our discussion to supervised methods. Neighborhood Components Analysis (NCA) (Goldberger et al., 2004) and its variants (Salakhutdinov and Hinton, 2007) implicitly learn a mapping function from high dimensional space to low dimensional space while preserving the neighbourhood relationship defined by class labels. However, NCA is optimized for nearest neighbor classification and does not take into account structure within the labels, only their identity. DrLIM (Hadsell et al., 2006) is an online, non-probabilistic method which explicitly learns a non-linear invariant embedding function. Similarly, we parameterize our embedding with a convolutional neural network (ConvNet) (LeCun et al., 1998), however, like NCA, DrLIM does not consider structure in the labels. Our method differs from NCA and DrLIM by incorporating the spatial layout of labels into an energy function, rather than a binary notion of neighbors defined by datapoints with the same label.

ConvNets are a successful deep architecture widely studied in various applications, such as object recognition (LeCun et al., 1998; Jarrett and Kavukcuoglu, 2009; Krizhevsky et al., 2012), scene parsing (Grangier et al., 2009; Farabet et al., 2012) and connectomics (Turaga et al., 2010). By parameter tying and feature pooling, ConvNets can automatically learn shift-invariant, discriminative low- and mid-level features from the raw pixels, avoiding the problem of generalization with hand-engineered features.

Farabet et al. (2012) have recently introduced multi-scale ConvNets and applied them successfully to Full Scene Labeling. Our work is similar in its motivation and the fact that we adopt their multi-scale approach to learn scale-invariant features. However, the way in which we approach the problem is very different. They apply sophisticated machinery based on optimal purity cover to search the best spatial grouping of feature vectors from which to predict a label. Our model has no such notion of adaptive pooling. Instead, we use weakly-supervised learning to introduce spatial context into the features. We believe the two approaches are complementary, and although beyond the scope of this paper, could be applied together.

### 3. Integrating spatial constraints into deep learning

In this section, we propose a method which learns to segment an image into parts which may be articulated. We pose this task as a typical classification problem, where each pixel of the image is classified into one of a discrete number of part labels  $\mathcal{L} = \{1 \dots L\}$ . We aim to make decisions pixel-wise in order to minimize computational complexity.

We are given a set of  $M$  images  $\{X^1, \dots, X^M\}$  and their associated labeled groundtruths. In our notation, the pixels of an image are indexed similarly index:  $X^m = \{X^m(i)\}$ . We seek to learn a segmentation model consisting of two parts:

- A parametric mapping function  $Z = f(X|\theta_f)$  which embeds each image  $X$  to a feature representation. This representation consists of  $Q$  maps, each of which have the same dimensions as  $X$ , and therefore can be indexed linearly:  $Z(i) \in \mathbb{R}^Q$ . The parameters  $\theta_f$  are learned from training data, taking into account Euclidean distances of pairs of features in the embedding space:  $d(i, j) = \|Z(i) - Z(j)\|_2$  (see Section 3.1).
- A classifier  $\hat{y}(i) = g(Z(i)|\theta_g)$  which classifies the features  $Z(i)$  given trained parameters  $\theta_g$  giving an estimate  $\hat{y}(i)$  of the part label (see Section 3.2).

As is common in the Deep Learning literature, the embedding can be pre-trained in an unsupervised (Hinton and Salakhutdinov, 2006) or supervised (Salakhutdinov and Hinton, 2007) manner based on auto-encoding or some other kind of inductive principle. Then, the classifier and the embedding are jointly learned in a supervised way by minimizing some classification-based loss<sup>1</sup>. Our method proceeds in a similar way. We assume that the spatial part layout remains consistent across the different images of a corpus. In particular, adjacency information of parts is assumed not to vary across images. In body part estimation, for example, we suppose that the upper arm and the forearm are always adjacent. The next two subsections will describe how spatial constraints can be integrated into, respectively, the training procedure for the embedding  $f(\cdot)$ , as well as the training procedure for the classifier  $g(\cdot)$ . The two contributions can be applied independently, or combined. The result is a

method proceeding as illustrated in Figure 1c: the information on the neighborhood layout is injected into a classifier working independently for each pixel  $i$ .

#### 3.1. Spatial pre-training

Common ways of learning the embedding function  $f(\cdot)$  are to minimize reconstruction error in an unsupervised auto-encoding setting (Hinton and Salakhutdinov, 2006), or in a supervised setting, map pixels with the same labels to close-by points in feature space and pixels with different labels to distant points (Salakhutdinov and Hinton, 2007; Hadsell et al., 2006; Taylor et al., 2011).

Given the availability of ground-truth data (i.e. the part labels), we advocate its use in the pre-training step as supervised or semi-supervised training has been shown to produce a more discriminative embedding (Salakhutdinov and Hinton, 2007). However, instead of using classification loss as a training criteria, we aim to benefit from the knowledge of the neighborhood layout of the parts in the corpus. We introduce a new energy function which exploits the spatial layout of different parts:

$$E = \sum_{i,j} \delta_{y(i),y(j)} L_S(i,j) + \sum_{i,j} \nu_{y(i),y(j)} L_D(i,j) \quad (1)$$

where  $y(i)$  and  $y(j)$  are the ground truth labels for pixels indexed by  $i$  and  $j$ , respectively;  $\delta_{a,b}$  is the Kronecker delta defined as  $\delta_{a,b}=1$  if  $a=b$  and 0 otherwise;  $\nu_{a,b}$  is defined as  $\nu_{a,b}=1$  if parts  $a$  and  $b$  are *not* neighbors in the corpus, and 0 otherwise.  $L_S$  and  $L_D$  are measured between pairs of features indexed by  $i$  and  $j$  in the feature space, which is embedded from raw data  $X$  through the embedding function. Note that we simply ignore contributions from pairs which have different labels but whose respective parts are neighbors in the corpus. As the exact boundaries between neighbors (given a human annotator) are somewhat arbitrary and often noisy, we choose to remain agnostic toward the relationship between features across a single boundary. We have chosen this scheme for simplicity, though we admit there are many other ways to design an energy function based on preserving spatial relationships. For example, if we could quantify the distance between labels (e.g. mean distance between body parts), such a “soft” neighbour criterion could be incorporated into the energy function. This is reserved for future work.

$L_S$  is the loss component for a pair of pixels having the same label, which has the effect of pushing their associated embeddings together.  $L_D$  is a contrastive loss

<sup>1</sup>If pre-training is supervised, the common setting is to perform classification by  $k$ -nearest neighbor search in embedded space, requiring no additional learning step.

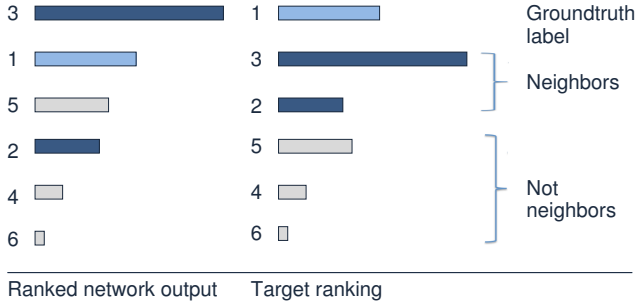


Figure 2: The proposed loss function based on differences in ranking.

component for a pair having different labels which pulls their embeddings apart. Similar to (Hadsell et al., 2006),  $L_S$  and  $L_D$  are set to

$$L_S(i, j) = \frac{1}{2}(d(i, j))^2, \quad (2)$$

$$L_D(i, j) = \frac{1}{2}[\max(0, \alpha - d(i, j))]^2, \quad (3)$$

where  $\alpha > 0$  is a margin. It controls the contrastive scope such that only dissimilar inputs which lie close-by in the embedded space contribute to the energy.

The gradient of Equation (1) with respect to the parameters can be computed via the chain rule (backpropagation):

$$\frac{\partial E}{\partial \theta_f} = \frac{\partial E}{\partial Z(i)} \frac{\partial Z(i)}{\partial \theta_f} + \frac{\partial E}{\partial Z(j)} \frac{\partial Z(j)}{\partial \theta_f} \quad (4)$$

where  $\frac{\partial E}{\partial Z(i)}$  and  $\frac{\partial E}{\partial Z(j)}$  can be obtained from (2) and (3).  $\frac{\partial Z(i)}{\partial \theta_f}$  and  $\frac{\partial Z(j)}{\partial \theta_f}$  depend on the particular architecture of the embedding function  $f(\cdot)$  which is described in Section 3.3.

We finally want to point out that, although the energy function is defined on pairwise pixel terms, testing is done pixelwise, so testing does not require solving a combinatorial problem.

### 3.2. Supervised spatial LR learning

In Section 3.1, we introduced spatial constraints into unsupervised pre-training of deep learning architectures, which are ConvNets in our setting (see Section 3.3). In this section we show that a similar principle can also be applied to supervised learning of classifiers. In our work, we choose a ConvNet with a single fully-connected classification layer (i.e. the top layer is equivalent to logistic regression) which are trained end-to-end. The same

principle, introducing spatial relationships into pixelwise classification, can also be applied to other classifiers, although the method may differ if the classifier is not learned to minimize classification error (or something akin to it, such as cross-entropy). In particular, in preliminary work we proposed a method to include spatial constraints into learning random forests by adapting the training algorithm based on maximizing gain in entropy (Jiu et al., 2013).

Classification-based neural nets are typically trained to minimize cross-entropy. When the normalized outputs of the net are viewed as probabilities, this is equivalent to maximizing the log probability the net assigns to the true class. In the multi-class setting, this involves normalizing the outputs of the network via a softmax function and comparing them to the groundtruth label. However, minimizing cross-entropy does not take into account the layout of the part labels.

We propose the following new loss function, which is based on the ranking of class labels according to network output. For each input vector, a forward pass gives a network response for each class label, which can be used to rank the class labels in decreasing order. A loss can be defined based on the difference between this ranking and a desired target ranking, which is defined by the following properties:

- The highest ranked class label should be the target groundtruth label. This constraint is related to the entropy loss in traditional neural network learning;
- The next highest-ranked class labels should be neighbors of the groundtruth label in the class neighborhood definition of the corpus. We advocate that better generalization to unseen data can be achieved by forcing the net to learn these constraints.

An example for this is given in Figure 2, where the groundtruth label for the pixel is 1. The actual output ranks the groundtruth label at second place. The target ranking ranks groundtruth label 1 at first place, followed by labels 3 and 2 which, in this example, are neighbors of label 1.

Learning to rank is a classical problem in machine learning which has been addressed in the literature (Burges et al., 2005; Freund et al., 2003; Dekel et al., 2004). We adopt a loss function similar in spirit to *RankNet* (Burges et al., 2005), defined on pairwise constraints. Given a pair of labels  $(u, v)$ , we denote by  $g_u(Z(i))$  and  $g_v(Z(i))$  the respective topmost (i.e. classification) layer outputs for location  $i$  and by  $o_{uv}(i) =$

$g_u(Z(i)) - g_v(Z(i))$  their difference. The probability of label  $u$  being ranked higher than  $v$  is mapped through a logistic function:

$$P_{uv}(i) = \frac{e^{o_{uv}(i)}}{1 + e^{o_{uv}(i)}} \quad (5)$$

Given a target probability  $\bar{P}_{uv}(i)$ , the loss function  $C_{uv}(i)$  is the cross entropy loss (Burges et al., 2005):

$$C_{uv}(i) = -\bar{P}_{uv}(i) \log P_{uv}(i) - (1 - \bar{P}_{uv}(i)) \log (1 - P_{uv}(i)) \quad (6)$$

The target probability  $\bar{P}_{uv}(i)$  is set to  $\lambda > 0.5$  if class  $u$  is ranked higher in the desired ranking, and to  $1 - \lambda$  otherwise.

Given the properties of the desired ranking described above, the following two sets of pairwise constraints have been derived:

1. A set of  $L - 1$  constraints, where each constraint specifies that the groundtruth label is to be ranked higher than one of the other labels;
2. A set of constraints each one specifying that a label  $u$ , which is a neighbor of the groundtruth label, should be ranked higher than another label  $v$ , which is not a neighbor of the groundtruth label.

The loss function  $\mathcal{E}$  for a single pixel is the sum over the pairwise constraints of the pairwise loss  $C_{uv}$ :

$$\mathcal{E}(i; l) = \sum_{u \neq l} C_{lu}(i) + \sum_{(u,v): \nu_{u,l}=0, \nu_{v,l}=1} C_{uv}(i) \quad (7)$$

where  $i$  is the index of the pixel, and we use the shortform  $l = y(i)$  as its ground truth label.

This loss function based on rankings provides a principled way of combining classification loss and spatial layout. Two types of constraints can be set through the  $\lambda$  parameter controlling the target probability  $\bar{P}_{uv}$ . Varying this parameter, different priorities could be given to different constraints.

It is important to note that this formulation allows one to include unlabeled data into the learning procedure in a semi-supervised setting. In this case, labeled images will be presented to the network with the loss function described above, whereas the loss function for unlabeled data does not contain the pairwise constraints including the groundtruth label - we consider *neighbours of the network's strongest output* instead of the ground truth neighbours.

As in Section 3.1, we would like to avoid confusion created by the term “pairwise”, which here involves pairs of labels selected during training, which typically are of

low cardinality compared to pixels. Testing proceeds independently for each pixel, conforming to the objectives of our work.

### 3.3. The deep learning architecture

While our mapping  $f(\cdot)$  can be any differentiable architecture, we adopt a two-stage ConvNet (LeCun et al., 1998). Each stage consists of a convolutional layer with a bank of filters, an element-wise non-linear squashing mapping (i.e. tanh function), followed by a spatial pooling and subsampling operator. A bank of feature maps are produced after convolving with the filters. The output of the first stage is fed as input to the convolutional layer at the second stage, where each feature map is connected to several maps in the previous stage. Connectivity is chosen uniform randomly before learning begins. The gradient of the energy function is computed with respect to the network output, and backpropagation is applied to update the parameters  $\theta_f$ . The parameters  $\theta_f$  consist of the filters and element-wise additive and multiplicative biases at the feature maps.

Many visual recognition problems require a large context due to complex interactions of different parts. However, the contextual size of ConvNets is limited by the filter size and sampling rate. Simply selecting larger filters does not address the fact that important cues may be observed at different scales. To overcome this dilemma, multiscale ConvNets (Farabet et al., 2012) are employed in our architecture, as shown in Figure 3. We use the notation  $f^s(\theta_f^s), \forall s \in \{1, \dots, N\}$  to denote the output produced by each scale ConvNet where  $s$  indexes the scales and  $N$  is the total number of scales. By employing the weight sharing across scales, ConvNets can learn scale-invariant features.

A multiscale Laplacian pyramid  $X^s, \forall s \in \{1, \dots, N\}$  is constructed for each image  $X$  in the pre-processing stage, where  $X^1$  has the same size as the original image. Local contrast normalization is applied to the Laplacian images to ensure that the local neighborhood has zero mean and standard deviation. A batch consisting of pairs of patches<sup>2</sup> is randomly extracted from each scale image  $X^s$ . The patches are processed by the corresponding ConvNet  $f^s$ , where the learned parameters  $\theta_f^s$  are shared across the scales.

<sup>2</sup>Since our loss function operates on pairs of features, i.e. particular locations  $i$  and  $j$  in the feature maps, we do not perform a full convolution while training. Instead, we extract patches corresponding to the receptive fields at locations  $i$  and  $j$  and map these patches to their respective feature vectors.

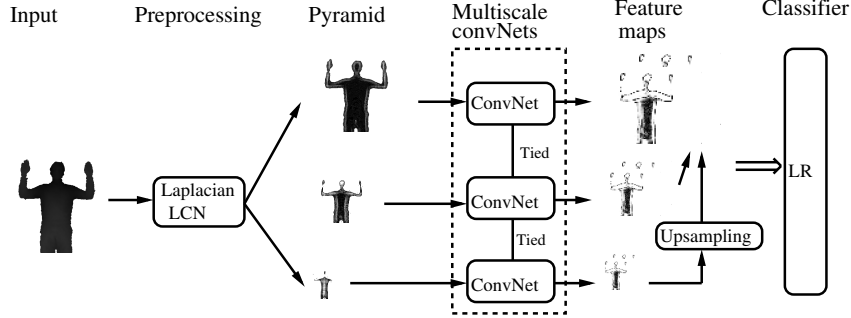


Figure 3: Multiscale ConvNets framework (Farabet et al., 2012). LCN means local contrast normalization. Each ConvNet contains several layers of convolutions and pooling described in 3.3.

The training procedure of the above architecture contains two steps. The first step is the spatial deep learning described in section 3.1, where the labels are only used to define spatial layout. We therefore call this stage weakly supervised pre-training. At this stage, the ConvNet parameters are initialized such that the features are consistent with the spatial arrangement of the labels. The second step is supervised spatial learning in section 3.2. A logistic regression layer parameterized with  $\theta_g$  is connected to the topmost feature maps of the ConvNet to predict the labels. We also apply a fine-tuning scheme in which the gradients from the LR are back-propagated to update the the ConvNet parameters  $\theta_f$ .

#### 4. Experimental results

We experimented with our framework to validate spatial learning with multiscale ConvNets in different situations, with respect to the application of specific human body part estimation from depth images.

Unless otherwise specified, we use a 2-stage ConvNet in all of the experiments below. The first convolutional layer consists of 8 filters of size  $9 \times 9$ , followed by an element-wise tanh function and a  $2 \times 2$  non-overlapping average pooling operator. The second convolutional layer consists of 32 filters of size  $9 \times 9$ , each of which combines 4 feature maps of the previous layer, with the same activation and pooling operator. The output is 32 feature maps. The receptive field for a single ConvNet on a single scale is  $28 \times 28$ . In a multiscale setting with a pyramid of 3 scales, with each scale consisting of 2 convolutional layers (Farabet et al., 2012), the receptive field is of size  $112 \times 112$ , thus a large spatial context is used to learn better features, comparable to best probe distance of  $129 \times 129$  of the randomized forest in (Shotton et al., 2011).

Local contrast normalization is applied to the inputs of the ConvNet, and the pixels from the background or the context are set to an arbitrary high value (in our case, 4) to distinguish from the zero-distributed pixels of the object. The margin  $\alpha$  for pre-training was set to 1.25 (Hadsell et al., 2006), and  $\lambda$  was set constant to 1 for all constraints in all experiments, i.e.  $\bar{P}_{uv} = 1$  if  $u$  is ranked higher than  $v$ , otherwise 0. Weakly-supervised feature learning and supervised learning each used 30 epochs through the entire dataset, where mini-batch gradient descent is adopted. End-to-end fine-tuning, if applied, used another 30 epochs. Different learning hyperparameters (such as learning rates for different layers) were chosen empirically, e.g.  $\epsilon_1$  is set to  $10^{-6}$  for the first convolutional layer and  $\epsilon_2$  to  $10^{-5}$  for the second convolutional layer. When performing classification, we consider each pixel in isolation, applying its corresponding 32-dimensional feature vector as input to an LR. The parameters of the LR are shared at each pixel location. In the following experiments, we report mean pixel-wise accuracy.

**The CDC4CV Poselets dataset** (Holt et al., 2011) has been designed for human upper body part segmentation from depth images converted and rescaled to the grayscale domain (0-255). It contains 345 training and 347 test depth images of upper body poses shot with a Kinect module. All images have been pre-processed by background subtraction and annotated. The body parts are: head, neck, left shoulder, right shoulder, left upper arm, left forearm, right upper arm, right forearm, left hip, and right hip. Additionally, the part below the waist is given the label of type “other” (see the black area in Figure 5, column 2). We define a spatial layout of 28 neighborhood relations among the 11 parts calculated from a canonical pose where a subject stretches his or her arms, for example, head–neck, neck–left shoulder,



Table 1: Evaluation of different baselines on the CDC4CV dataset.

Methods	Accuracy
Randomized forest (Shotton et al., 2011)	60.30%
Spatial Randomized forest (Jiu et al., 2013)	61.05%
Single-scale (vanilla) ConvNet (LeCun et al., 1998)	47.17%
Multi-scale ConvNet (Farabet et al., 2012)	62.54%

neck–right shoulder, left forearm–left upper arm, and so on. 900 pairs of patches were randomly chosen at three different scales for spatial pre-training.

For the application of part estimation from depth images, our baselines are the algorithms based on randomized decision forests described in (Shotton et al., 2011), as well as single- and multi-scale ConvNets with supervised end-to-end training without any spatial context. We also compare our framework with the recent spatial learning algorithm in randomized decision forests (Jiu et al., 2013). In our baseline experiments, a randomized forest of three trees with a depth of 9 is learned. For each node, the best offsets and threshold are chosen from 4000 offset candidates and 22 threshold candidates. More details can be found in (Jiu et al., 2013). From Table 1, it is clear that the large receptive field of the multi-scale architecture is necessary to reliably estimate body part labels, and also that the Deep Learning-based architecture outperforms random forests on the CDC4CV dataset.

We investigate the performance of different combinations of our framework based on spatial pre-training of the purely convolutional layers (which we call ConvNet) and the classification layer (which we call LR). As a non-spatial baseline, we also implemented a “pixel-wise” version of DrLIM (Hadsell et al., 2006), a similar pre-training strategy in which spatial layout is not taken into consideration, i.e.  $\delta_{a,b}=1$  if  $a=b$  and 0 otherwise;  $\nu_{a,b}=1$  if  $a \neq b$  and 0 otherwise in equation 1. The results are shown in Table 2. For each setting (fine-tuning or no fine-tuning), spatial training outperforms non spatial training, and in many cases, the gains are high.

Examples of segmentation results are shown in Figure 5. From visual inspection we can see that the segmentation results produced by spatial learning are better than the ones by the non spatial methods. In particular, the segmentation produced by spatial learning is more consistent (less noisy), especially for the arms.

**Spatial distribution of the error** — The proposed method injects neighborhood relationships into the training algorithm. The question arises, whether the benefit of the proposed method only applies to pix-

Table 2: Results of different combinations of classical and spatial learning on the CDC4CV dataset. Fine-tuning means end-to-end training of the LR (top two layers) and ConvNet (remaining layers) with the same objective used to train the LR.

Convolutional layers	LR	Fine-tuning	Accuracy
DrLIM	classical	no	35.10%
DrLIM	spatial	no	41.05%
spatial	classical	no	38.60%
spatial	spatial	no	<b>41.65%</b>
DrLIM	classical	yes	64.39%
DrLIM	spatial	yes	65.12%
spatial	classical	yes	65.18%
spatial	spatial	yes	<b>66.92%</b>

els at the boundaries, which would imply a rather trivial improvement. To rule this out, an in-depth analysis has been performed to verify that the proposed methods generally improve classification error independent of the pixel positions.

For each pixel in the test images the distance to the nearest part boundary was calculated using a distance transform, which allowed us to calculate histograms over these distances, where each bin corresponds to a range of distances. Histogram  $H_g$  shows the number of pixels of the test images over distances, shown in Figure 4(a). This histogram largely depends on the sizes of the different body parts, and we can see that the pixel count decreases with distance. Distributions  $H_s$  and  $H_b$  are over pixels which are better classified by the baseline or by the proposed method, respectively:

- Distribution  $H_s$ , shown in Figure 4(b), is the distribution of pixels which have been correctly classified by the proposed method but wrongly classified by the baseline method (DrLIM and classical LR with fine-tuning);
- Distribution  $H_b$ , shown in Figure 4(c), is the distribution of pixels which have been correctly classified by the baseline method (DrLIM and classical LR with fine-tuning) but wrongly classified by the proposed method.

The normalized histogram  $H_d$ , shown in Figure 4(d), illustrates the contribution of the proposed spatial learning method as a function of the pixels’ distances from the part boundaries:

$$H_d(i) = (H_s(i) - H_b(i))/H_g(i). \quad (8)$$

Our method gains over the very large majority of distances except for a few outliers. We see a strong bimodality, i.e. our method wins close-to and away from

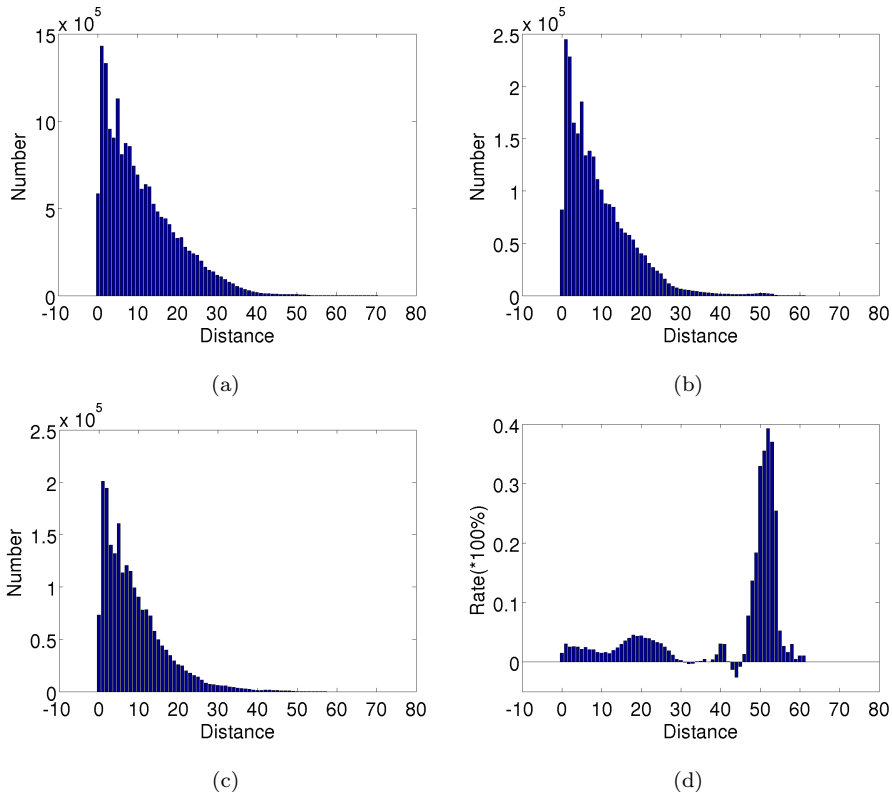


Figure 4: Histograms over different distances to part boundaries: (a)  $H_g$  — the histogram of all pixels; (b)  $H_s$  — the histogram of pixels for which the proposed method outperforms the baseline; (c)  $H_b$  — the histogram of pixels for which the baseline outperforms the proposed method; (d)  $H_d$  — is the normalized difference histogram.

the borders. Moreover, the mode corresponding to pixels near the center of parts shows an even higher proportion of improvement than the mode closer to the border. In particular, the classification rates are significantly improved between distance 50 and 60, which is more or less half of the receptive size. At the same time, this analysis verifies that our system indeed does benefit from the multi-scale framework.

## 5. Discussion

The experiments above demonstrate that our deep learning framework indeed benefits from the *a priori* additional information contained in the spatial part layout. Our framework can easily extend to semi-supervised learning, taking advantage of additional unlabeled training data. On this unlabeled data, the loss function would not include any terms based on classification loss (which require the ground truth part label), but only terms based on the spatial layout of the parts, as mentioned in Section 3.2.

It is worth noting that the two proposed contributions can be applied independently. In particular, the supervised spatial learning algorithm can be combined with any supervised or unsupervised training algorithms for ConvNets, for example auto-encoding or other algorithms reviewed in Section 2.

The quantitative evaluation measures presented in the different tables report pixelwise classification accuracy. This proves that the proposed learning algorithm improves pixelwise classification, although the injected additional information does not specifically state this. An additional advantage of our method is *not* captured by this evaluation: if a pixel is wrongly classified, then spatial learning will increase the probability that the wrong part label is a neighbor of the correct part label. In some applications this can be an advantage, for instance if the segmentation is post-processed to estimate joint positions, as in (Shotton et al., 2011).

Currently, we demonstrate one application of the proposed method to human part estimation in depth images. The approach is, however, not specific to depth

Table 3: Running times on the CDC4CV dataset of our proposed framework, compared with Randomized forest (RF). Training on 344 images includes pre-training, LR learning and fine-tuning, and testing time on one image is given.

Task	Method	Machine	Architecture	Time
Testing	Ours	Xeon E56620, 8 cores, 2.4GHz	MATLAB+IPP	00.41 sec
Testing	Ours	Xeon E56620, 8 cores, 2.4GHz	MATLAB+GPU	00.31 sec
Testing	RF	Xeon E56620, 8 cores, 2.4GHz	C++	00.73 sec
Training	Ours	Xeon E56620, 8 cores, 2.4GHz	MATLAB+GPU	36.00 h
Training	RF	Xeon E56620, 8 cores, 2.4GHz	C++	15.8 h

images, and can be applied to other problems, as for instance object segmentation from RGB images, or semantic full scene labeling. In these cases, the context may be dynamic and the spatial layout might be more difficult to extract from the corpus, requiring learning techniques. This is reserved for future work.

The main computational complexity in our framework comes from the convolution operations, which can be significantly accelerated by parallel computation, e.g. on multi-core CPUs or GPUs. We implemented the system in MATLAB with multi-threaded convolutions provided by the GPU (CUDA) library (for training and testing), and the testing time by the IPP framework is also given as a comparison. We also give the running time of the randomized forest method implemented in C++. Although our training stage needs more computation time, testing per image is more efficient, since our method profits from a parallel implementation. Table 3 gives running times for testing and for training.

## 6. Conclusion

In this paper, we proposed a way to significantly improve classification performance in segmentation problems by integrating prior information on the spatial layout of image or object parts into a learning architecture. Compared to other spatial relationship learning algorithms, the energy function in our algorithm does not contain pairwise pixel terms, which makes it extremely fast. In particular, there is a *zero* increase in complexity compared to a standard pixelwise classifier of the same type in the test stage. We demonstrate applications of our approach to human body part labeling from depth images. Our algorithm outperforms the well known method based on randomized decision forests and other supervised feature mapping methods.

## References

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to

rank using gradient descent. In: ICML.

Cox, T., Cox, M., 1994. Multidimensional scaling. Chapman and Hill.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: CVPR.

Dekel, O., Manning, C., Singer, Y., 2004. Log-linear models for label-ranking. In: NIPS.

Farabet, C., Couprie, C., Najman, L., LeCun, Y., 2012. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In: ICML.

Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part based models. IEEE Tr on PAMI 32 (9), 1627–1645.

Felzenszwalb, P., Huttenlocher, D., 2005. Pictorial Structures for Object Recognition. IJCV 61 (1), 55–79.

Freund, Y., Iyer, R., Schapire, R., Singer, Y., 2003. An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research 4, 933–969.

Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R., 2004. Neighbourhood component analysis. In: NIPS.

Grangier, G., Bottou, L., R.Collobert, 2009. Deep convolutional networks for scene parsing. In: ICML Deep Learning Workshop.

Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In: CVPR. pp. 1735–1742.

Hinton, G., Salakhutdinov, R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

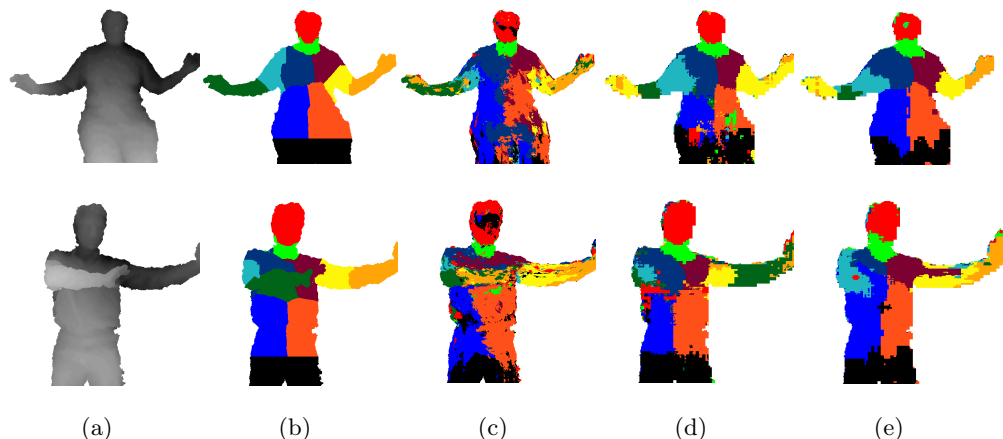


Figure 5: Classification examples from the CDC4CV dataset. (a) input depth image; (b) groundtruth segmentation; (c) appropriate baseline: randomized forest for CDC4CV; (d) DrLIM+LR without spatial learning; (e) our method (spatial pre-training and spatial LR learning).

- Holt, B., Ong, E.-J., Cooper, H., Bowden, R., 2011. Putting the pieces together: Connected poselets for human pose estimation. In: ICCV Workshop on Consumer Depth Cameras for Computer Vision.
- Jarrett, K., Kavukcuoglu, K., 2009. What is the best multi-stage architecture for object recognition? In: ICCV.
- Jiu, M., Wolf, C., Baskurt, A., 2013. Integrating spatial layout of object parts into classification without pairwise terms - application to fast body parts estimation from depth images. In: Proceeding of the conference on Computer Vision Theory and Applications (VISAPP).
- Jolliffe, T. I., 1986. Principal component analysis. In: New York: Springer-Verlag.
- Kolmogorov, V., Zabih, R., 2004. What energy functions can be minimized via graph cuts? IEEE Tr. on PAMI 26 (2), 147–159.
- Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks. In: NIPS.
- LeCun, Y., Botto, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of IEEE 86 (11), 2278–2324.
- Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.
- Salakhutdinov, R., Hinton, G., 2007. Learning a non-linear embedding by preserving class neighbourhood structure. In: AISTATS. Vol. 11.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A., 2011. Real-time human pose recognition in parts from single depth images. In: CVPR.
- Taylor, G., Spiro, I., Bregler, C., Fergus, R., 2011. Learning invariance through imitation. In: CVPR. pp. 2729–2736.
- Turaga, S. C., Murray, J. F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., Seung, H. S., 2010. Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Computation 22, 511–538.
- Winn, J., Shotton, J., 2006. The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR. Vol. 1. pp. 37–44.