Location Constrained Pixel Classifiers for Image Parsing with Regular Spatial Layout

Kang Dang
kangdang@gmail.com
Junsong Yuan
jsyuan@ntu.edu.sg

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

Abstract

When parsing images with regular spatial layout, the location of a pixel (x,y) can provide important prior for its semantic label. This paper proposes a novel way to leverage both location and appearance information for pixel labeling. The proposed method utilizes the spatial layout of the image by building local pixel classifiers that are location constrained, i.e., trained with pixels from a local neighborhood region only. Albeit simple, our proposed local learning works surprisingly well in different challenging image parsing problems, such as pedestrian parsing and object segmentation, and outperforms state-of-the-art results using global classifiers. To better understand the behavior of our local classifier, we perform bias-variance analysis, and demonstrate that the proposed local classifier essentially performs spatial smoothness over the global classifier that uses appearance information and location, which explains why the local classifier is more discriminative but can still handle mis-alignment. Meanwhile, our theoretical and experimental studies suggest the importance of selecting an appropriate neighborhood size to perform location constrained learning, which can significantly influence the parsing results.

1 Introduction

Spatial layout of an image conveys significant information for labeling its pixels. For example, in street view images, the sky pixels are more likely to appear in the top of the image, and road pixels are more likely to appear at the bottom. As illustrated in Figure 1, absolute location is useful for a variety of image parsing problems, such as pedestrian parsing after detection [5], street view scene parsing [23, 37], and medical image segmentation [12, 36]. This paper focuses on utilizing absolute location of a pixel to improve image parsing.

Many approaches have been proposed to fuse absolute position (x,y) and feature vector $\mathbf{f}$ for image parsing [2, 3, 5, 32, 36, 37]. Some of them use early fusion, e.g., concatenating feature and position to form $(\mathbf{f},x,y)$, then train a global discriminative classifier. Others use late fusion, e.g., they firstly model $p(L \mid \mathbf{f})$ and $p(L \mid x,y)$ separately where L stands for pixel label, and combine them with weighted multiplication and normalization. Most of them attempt to solve the pixel classification problem with a single global model. In other words, they learn a single global pixel classifier for the entire image space, and all pixels of the image are used to train the classifier.

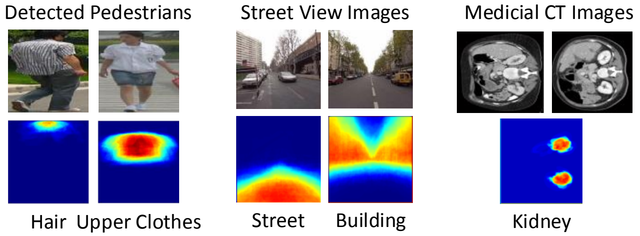

Figure 1: This figure shows the images and absolute location priors of certain semantic labels. The location priors are obtained from the corresponding dataset. (Best viewed in color)

In contrast, our method is based on local learning. Given location (x, y) and feature $\mathbf{f}$, instead of learning a universal pixel classifier, i.e., global classifier, for the entire image, at each image location we learn a location constrained classifier, i.e. local classifier. Each local classifier is trained only by pixel samples from the neighborhood region $\mathcal{N}(x, y, s)$ centered at (x, y) and of scale s. Since each local pixel classifier is learned by only using the pixels in a local neighborhood, it is expected to better fit the local pixel distribution and capture local discriminative information. Compared with building a global classifier, such a local learning task is usually easier, as it avoids dealing with confusing negative samples outside the local region, which have similar feature to positive samples and easily confuse the global learning. Meanwhile, as the number of classes that can be observed in a local neighborhood is usually not many, learning the local classifier is less challenging too. To prevent local classifier overly depending on the image location and to improve generalization, the neighborhood scale s is important. On one hand, if s is too large, each local classifier behaves more like a global one. On the other hand, if s is too small, local classifiers strongly depend on location and will be sensitive to the image misalignment. To better understand the trade-off, we perform bias-variance analysis on the local classifiers and establish the relationship between local learning and global learning. Our theoretical and experimental results both validate that a proper selection of neighborhood size s is critical to obtain good performance.

The main steps of our proposed algorithm are illustrated in Figure 2. At each location we learn a location constrained local classifier with training samples only from surrounding pixels in a neighborhood. Then the learned classifier will be used to classify pixels of the same region in a testing image. To ensure smoothed labeling results, we allow the local classifiers to overlap with each other, such that each pixel will be voted by multiple local classifiers. The final score is the average score of all engaged local classifiers. Therefore, after each local classifier outputs a local labeling map, we merge all local maps to build the overall labeling map. The final result is obtained after a proper discretization of the labeling map with a conditional random field (CRF). We verify the performance of our proposed algorithm on two pedestrian parsing datasets [5, 25] and Weizmann horse dataset [6]. The experiments demonstrate that the proposed algorithm achieves significant improvements compared with state of the arts.

2 Related Work

Layout Modeling. Spatial layout can be classified into two perspectives, relative [32, 35] and absolute [2, 3, 5, 21, 23]. While relative layout is more flexible, it does not account for

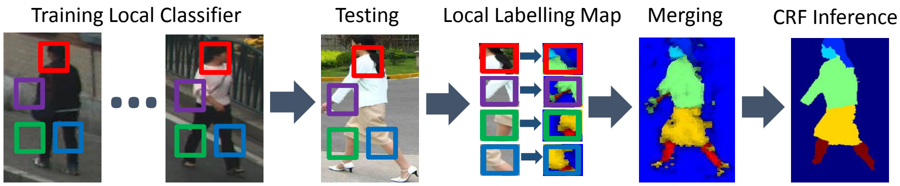

Figure 2: At each location we train a position dependent local pixel classifier with training samples from its neighborhood region represented by a patch. Assume the training and testing images have similar layout, the trained classifier is used for the same region in the testing images. Each classifier outputs a local labeling map, and these are merged through averaging to build the overall labeling map. (Best viewed in color)

the dependency on global image coordinates in an explicit manner. Such information is better modeled with absolute location. Because of the spatial misalignment, absolute location can be unreliable. For some methods, the principle is to correct the spatial misalignment. In [9, 23] the segmentation is done by label transfer with a matching processing to enhance the localization. In [12] image registration is performed prior to the graph-cut based segmentation to correct any misalignment. While these methods are effective, residue misalignment errors may still be present after the correction. As a result, an alternative approach is also useful based on tolerating the spatial misalignment. In spatial pyramid matching for image categorization [18], a multi-scale grid is used to model the absolute location at different scale levels. It is further extended in [16] where location is encoded with a Gaussian mixture model and variance of each Gaussian represents spatial uncertainty. In distribution field for tracking [30], an image descriptor is built to smooth the absolute location without affecting the intensity values. While [16, 18, 30] are not directly applicable to pixel labeling, the principle is related to our work, which is to tolerant rather than to correct the misalignment.
Local Learning. In [13, 29] it is discovered that for object detection, training a separate detector for each image location can significantly reduce the problem complexity. In [22] a multi-task local boosting is proposed for the object recognition application. Cuingnet et al. propose a two stage procedure for kidney segmentation [10]: kidney detection is firstly performed, before subsequently training a classifier specifically for labeling the detected region. Ren et al. propose to train local region regressors for the face alignment task [28]. While these applications demonstrate the effectiveness of the local learning, an important issue which has not been thoroughly studied is the influences of neighborhood scale of local learning. Our work specifically targets at this issue, and we have thoroughly discussed the connection between the global and local learning in the context of pixel labeling.

3 Our Approach

3.1 Feature Extraction

Given an image, we firstly use SLIC superpixel algorithm [1] as a preprocessing step to over-segment the image. All the subsequent operations are then performed on superpixel level instead of pixel level to reduce computational cost. We notice that in most cases an image of size 300×300 pixels can be represented with only 2000 superpixels without sacrificing important details such as object boundaries. Thus the label of each superpixel can be just set to the majority labels of its constitute pixels. However, in the rare event that a superpixel contains multiple objects of similar size, so the percentage of pixels with majority labeling is less than 70%, we consider such a superpixel label unreliable and remove it from the training set.

To represent each superpixel, multiple types of raw features are used such as RFS filter banks [19], dense SIFT [24] and LBP [26]. For each type, we firstly construct the dictionary via K-means, and then assign the raw feature of every pixel to its nearest cluster center. With every pixel assigned a cluster center index, we have obtained the texton map of the image and the cluster center index is called a texton index [38]. To represent the superpixel, we extract a patch around the superpixel center and then compute the histogram of texton indexes within the patch. After performing this calculation for all three types of raw features, the final representation is given by the concatenation of the three histograms.

3.2 Training Local Classifier

We have a collection of training images $\mathcal{I} \in \mathbb{R}^{\mathcal{W}\times\mathcal{H}}$, where $\mathcal{W}$ and $\mathcal{H}$ stand for width and height of the images respectively. Given a superpixel's central position (x,y) and its associated feature vector $\mathbf{f}$, our goal is to predict the class label L of that superpixel. Instead of simply concatenating $\mathbf{f}$ and (x,y) and training a global classifier $p(L \mid \mathbf{f},x,y)$ with samples from the whole image $\mathcal{I}$, we are interested in learning a number of local classifiers $p_{\mathcal{N}}(L \mid \mathbf{f})$ at different spatial locations. $\mathcal{N}(x,y,s)$ or its abbreviation $\mathcal{N}$ stands for a local image neighborhood, which is a patch centered at (x,y) and of width $s\times\mathcal{W}$ and height $s\times\mathcal{H}$, where s is the neighborhood scale. In other words, the training set for each local classifier is $\{(L_i,\mathbf{f}_i) \mid \forall(x_i,y_i) \in \mathcal{N}(x,y,s)\}$. We use linear SVM [11] to train the local classifier while other more advanced learning methods can be used as well. We discuss three issues relating to the training.

Firstly, the neighborhood scale s is an important parameter for the performance of our proposed local classifier. Its theoretical implication is explained detailedly in this section and section 4. We use a validation set to select the neighborhood scale.

Secondly, as it is computationally intensive to train classifiers at all locations to score the whole image, they are trained coarsely on a uniform grid. We set the grid spacing in proportional to the neighborhood scale s; so for classifiers of larger s their locations are sampled more coarsely. The reason is that classifiers with larger s have larger spatial support, so nearby classifiers become more similar to each other (see the discussion after Theorem 1) and the performance gain by using dense training location will become lesser. In our implementation the grid spacing is set to $0.2\times s\times(\mathcal{W},\mathcal{H})$, where s ranges from 0 to 1.

Thirdly, for problems such as pedestrian parsing, there is a significant class imbalancing issue. To better handle the minority labeling such as arms and legs, we do not subsample each class in proportional to their occurring frequency. Instead, we subsample majority labeling more aggressively and minority labeling less aggressively, in a similar manner proposed by Tighe and Lazebnik [33]. We find such a scheme is critical for the performance.

Probabilistic modeling. We give an exact definition of our proposed classifier in terms of conditional probability. By such an analysis we also suggest the important role of selecting a good neighborhood scale s. At each superpixel position (x_0,y_0), the following local distribution $p_{\mathcal{N}}(L,\mathbf{f},x,y)$ models the superpixels within a local neighborhood $\mathcal{N}(x_0,y_0,s)$:

$$p_{\mathcal{N}}(L,\mathbf{f},x,y) = \begin{cases} \frac{p(L,\mathbf{f},x,y)}{\sum_{(x',y')\in\mathcal{N}} p(x',y')}, & \forall(x,y)\in\mathcal{N} \\ 0, & \forall(x,y)\notin\mathcal{N} \end{cases}, \tag{1}$$

where $p(x',y') = \frac{1}{\mathcal{W}\times\mathcal{H}}$. Different from the global distribution $p(L,\mathbf{f},x,y)$ where $(x,y) \in$

$\mathcal{I}$, $p_{\mathcal{N}}(L,\mathbf{f},x,y)$ only models the superpixels within the local neighborhood thus is a local distribution. Correspondingly, the local classifier approximates the following conditional distribution:

$$\begin{aligned} p_{\mathcal{N}}(L\,|\,\mathbf{f}) &= \frac{\sum_{(x,y)\in\mathcal{I}} p_{\mathcal{N}}(L,\mathbf{f},x,y)}{\sum_{(x,y)\in\mathcal{I}} p_{\mathcal{N}}(\mathbf{f},x,y)} = \frac{\sum_{(x,y)\in\mathcal{N}} p(L,\mathbf{f},x,y)}{\sum_{(x,y)\in\mathcal{N}} p(\mathbf{f},x,y)} \\ &= \frac{\sum_{(x,y)\in\mathcal{N}} p(L\,|\,\mathbf{f},x,y)p(\mathbf{f}|x,y)}{\sum_{(x,y)\in\mathcal{N}} p(\mathbf{f}\,|\,x,y)} \propto \sum_{(x,y)\in\mathcal{N}} p(L\,|\,\mathbf{f},x,y)p(\mathbf{f}\,|\,x,y). \end{aligned} \tag{2}$$

Equation 2 explains the relationship between our local classifier $p_{\mathcal{N}}(L\,|\,\mathbf{f})$ and the global classifier $p(L\,|\,\mathbf{f},x,y)$ that uses both f and (x,y). The proposed local classifier $p_{\mathcal{N}}(L\,|\,\mathbf{f})$ is a spatially smoothed version of the global classifier $p(L\,|\,\mathbf{f},x,y)$ in a local neighborhood, where the weight $p(\mathbf{f}\,|\,x,y)$ characterizes the dependency of the observed feature $\mathbf{f}$ at the pixel location (x,y). In this sense, $p_{\mathcal{N}}(\mathbf{f}\,|\,x,y)$ actually serves as a smoothing kernel and s determines the smoothing bandwidth. The neighborhood scale s plays an important role in building the local classifier. On one hand, when the local neighborhood contains only a single superpixel, *i.e.*, $s=0$, our local classifier degenerates into:

$$p_{\mathcal{N}(x,y,0)}(L\,|\,\mathbf{f}) = p(L\,|\,\mathbf{f},x,y). \tag{3}$$

On the other hand, when the local neighborhood expands to the entire image, *i.e.*, $s=1$, it becomes $p_{\mathcal{N}(x,y,1)}(L\,|\,\mathbf{f}) = p(L\,|\,\mathbf{f})$, which indicates position information (x,y) is not utilized at all. In Section 4, we will further discuss the implication of choosing an appropriate neighborhood scale s for local classifiers from the perspective of bias-variance analysis.

3.3 Testing and CRF Inference

During the testing to score a superpixel at position (x,y), straightforwardly we can just perform prediction using the trained classifier at a nearest neighboring location. That is: $p_{nearest}(L\,|\,\mathbf{f},x,y,s) = p_{\mathcal{N}_{(x_k,y_k,s)}}(L\,|\,\mathbf{f})$, where (x_k,y_k) is the location of the nearest trained classifier to (x,y). However this is less effective as a testing location is related to multiple nearby classifiers. Instead we use an averaging merging procedure as shown in Figure 2.

For the average merging, after training the local pixel classifier $p_{\mathcal{N}}(L\,|\,\mathbf{f})$, it is used to label all superpixels belonging to the same patch $(x,y)\in\mathcal{N}$. So at each position (x,y) a superpixel will receive multiple votes from nearby local classifiers that cover that superpixel, *i.e.*, if $(x,y)\in\mathcal{N}$ the local classifier $p_{\mathcal{N}}(L\,|\,\mathbf{f})$ will vote for (x,y). We collect all local classifiers that can influence (x,y): $\Omega = \{k : (x,y)\in\mathcal{N}_{(x_k,y_k,s)}\}$, where k is the classifier's index. After that all probabilistic scores of nearby classifiers are averaged for the final score: $p_{merged}(L\,|\,\mathbf{f},x,y,s) = \frac{1}{|\Omega|}\sum_{k\in\Omega} p_{\mathcal{N}_{(x_k,y_k,s)}}(L\,|\,\mathbf{f})$. Experiments confirm that such an average merging procedure leads to an improved performance compared with scoring using a single classifier.

For transformation of the classifier probabilistic output to discrete labeling results, we use a simple CRF model with pairwise smoothing [23] and solve it using graph-cut with the GCO library[1] [7, 8, 15]. Similar as before, our CRF is constructed on superpixel level; so each node of the graph corresponds to a superpixel in the image.

[1] http://www.csd.uwo.ca/~olga/OldCode.html

4 Bias-Variance Trade-off

Bias-variance analysis is commonly used to understand data fitting and model selection. Using bias-variance analysis, we will further discuss the influence of neighborhood scale s for local classifiers.

To perform bias-variance analysis of the local classifiers, we assume to have a large number of training sets of images, following the global joint distribution $q(L,\mathbf{f},x,y)$. Each training set can build its own local classifier $p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})$, and its average is $\mathbb{E}(p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f}))$, where the expectation is taken over all the training sets. To evaluate the loss of a local classifier obtained from a specific training set, we measure the KL-divergence between the target $q(L\mid\mathbf{f},x,y)$ and $p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})$. The target equals to $q_{\mathcal{N}(x,y,0)}(L\mid\mathbf{f})$ according to Equation 3:

$$\begin{aligned} loss(p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})) &= \mathbb{E}(d_{KL}(\underbrace{q_{\mathcal{N}(x,y,0)}(L\mid\mathbf{f})}_{target} \mid\mid \underbrace{p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})}_{model}) \\ &= \mathbb{E}(\sum_{L,\mathbf{f}} q_{\mathcal{N}(x,y,0)}(L,\mathbf{f}) log \frac{q_{\mathcal{N}(x,y,0)}(L\mid\mathbf{f})}{p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})}), \end{aligned} \tag{4}$$

where the expectation is taken over all the training sets. Following the bias-variance analysis in [14], we can further decompose the loss into bias and variance terms:

$$\begin{aligned} & loss(p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})) \\ = \; & \underbrace{d_{KL}(\underbrace{q_{\mathcal{N}(x,y,0)}(L\mid\mathbf{f})}_{target} \mid\mid \underbrace{\mathbb{E}(p_{\mathcal{N}}(L\mid\mathbf{f}))}_{average})}_{bias(p_{\mathcal{N}(x,y,s)}(L|\mathbf{f}))} + \underbrace{\mathbb{E}(d_{KL}(\underbrace{\mathbb{E}(p_{\mathcal{N}}(L\mid\mathbf{f}))}_{average} \mid\mid \underbrace{p_{\mathcal{N}}(L\mid\mathbf{f})}_{model}))}_{var(p_{\mathcal{N}(x,y,s)}(L|\mathbf{f}))}, \end{aligned} \tag{5}$$

Let:

$$var(s) = \sum_{(x,y)\in\mathcal{I}} var(p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f})), \tag{6}$$

where $var(p_{\mathcal{N}(x,y,s)}(L\mid\mathbf{f}))$ is defined by Equation 5. Based on the bias-variance analysis, the following theorem shows that averaging over all the positions, the testing error variance satisfies the following monotonically decreasing property.

Theorem 1. *Assume s is much smaller than the image size and $s_1 = k\times s_2$ for k being an integer, and:*

$$\sum_{(x',y')\in\mathcal{N}(x,y,s)} p_{i,\mathcal{N}}(\mathbf{f},x',y') = \sum_{(x',y')\in\mathcal{N}(x,y,s)} p_{j,\mathcal{N}}(\mathbf{f},x',y'), \tag{7}$$

for any position $(x,y)\in\mathcal{I}$ and training set indexes i and j, we have:

$$var(s_1) \le var(s_2),$$

where $var(s)$ is defined by Equation 6.

From Theorem 1, using a larger neighborhood scale s for local learning will result in a larger overlap among the nearby local classifiers, such that nearby local classifiers behave more similarly to each other. Such a smoothness makes the classification less sensitive to the alignment variations, thus can better tolerate spatial misalignment.

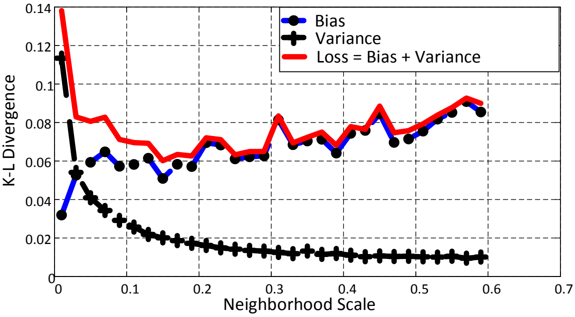

Figure 3: Bias-variance trade-off on Weizmann horses dataset.

For the *bias* term defined by Equation 5, by assuming that the ensemble is unbiased, the following approximation holds: $\mathbb{E}(p_{\mathcal{N}(x,y,s)}(L \mid \mathbf{f})) \approx q_{\mathcal{N}(x,y,s)}(L \mid \mathbf{f})$. The bias term of our model can then be approximated as: $d_{KL}(q_{\mathcal{N}(x,y,0)}(L \mid \mathbf{f}) \mid\mid q_{\mathcal{N}(x,y,s)}(L \mid \mathbf{f}))$. From the above approximation, we see that the bias will equal zero when neighborhood scale $s = 0$. As KL divergence is non-negative, the bias terms will increase with a larger neighborhood scale.

To evaluate our result, we simulate the bias-variance trade-off with Weizmann horses dataset, and the result is shown in Figure 3. We use the whole dataset for calculation of the target, while randomly selecting 20% of all images for 10 times as training sets for calculation of an ensemble of 10 models. Feature vector **f** of each pixel is simply its intensity value, and we calculate the bias, variance and loss by Equation 4 and 5. The simulation confirms that bias increases with the neighborhood scale and variance decreases. In addition, we can also find an appropriate neighborhood scale to reduce the testing loss significantly. From the simulation, we understand that an appropriate neighborhood scale to balance the bias and variance is essential for minimizing the testing error. In the implementation, we use a validation set to select the neighborhood scale.

5 Experiments

5.1 Pedestrian Parsing

Our first set of experiments are performed on Penn-Fudan [5] and PPSS dataset [25]. The datasets contain 7 semantic labels of body parts, such as hair, face, upper-clothes, etc. In these two datasets, location prior becomes useful as parsing is performed within the detected bounding box. The dataset splitting is identical to the previous works [5, 25], and we reserve 20% from the training set to be used as validation images. In this experiment, the accuracy metric for each semantic label is intersection over union (IOU) score defined between ground-truth A and output B by $\frac{|A \cap B|}{|A \cup B|}$ ([5]).

Comparison with Alternative Approaches of Fusion f and (x, y)**.** We firstly compare the proposed approach with three commonly used methods of combining feature vector **f** and location coordinate (x, y). (1) $(\mathbf{f}, x, y)$ + SVM (as used in [3]): we concatenate feature and position information together to form $(\mathbf{f}, x, y)$, and put it into a SVM classifier. (2) $(\mathbf{f}, x, y)$ + Boosting (as used in [36, 37]): we put the concatenated feature vector $(\mathbf{f}, x, y)$ into a joint boosting classifier [34]. (3) Product of Experts (as used in [23]): the merge is done by multiplying the two posterior probability map with weighting: $\frac{p(L|x,y)^k p(L|\mathbf{f})^{(1-k)}}{Z}$, where k is

	Penn-Fudan	PPSS
Feature Only	45.1	31.8
$(\mathbf{f},x,y)$ + SVM	54.3	39.7
$(\mathbf{f},x,y)$ + Boosting	60.3	45.1
Product of Experts	52.6	45.1
Ours(nearest)	62.1	52.7
Ours(merged)	**63.1**	**53.5**

Penn-Fudan	
SBP[5]	57.3
P&S[27]	55.0
DL[25]	59.9
Ours	**63.1**
PPSS	
DDN[25]	47.2
Ours	**53.5**

Table 1: Benchmark results for Penn-Fudan and PPSS dataset. The performance metric is the average intersection over union (IOU) score over all labels.

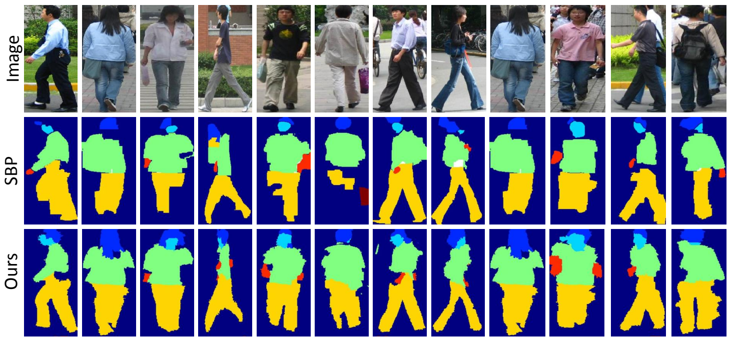

Figure 4: Image results from Penn-Fudan dataset. Visual quality is generally better than SBP[5]. (Best viewed in color)

between 0 and 1, and Z is a normalization constant. To ensure a fair comparison, all the parameters have been throughly adjusted by validation.

Table 1 shows that our method's performance is superior than the alternative ways of feature fusion. It confirms the advantages of our local classifiers that they are better adapted to the local image characteristics than a global classifier. Table 1 also demonstrates that the average merging discussed in section 3.3 (denoted as "merged") performs better than scoring with a single classifier at the nearest neighbor location (denoted as "nearest"). Thus in all the comparisons with state of the arts, average merging scheme is used.

State of the Arts Comparison. Table 1 shows that the performance is greatly improved compared with state of the arts, although it has not been designed specially for the purpose of pedestrian parsing. From Figure 4 and 5 we see that except for the occlusion cases, parsing

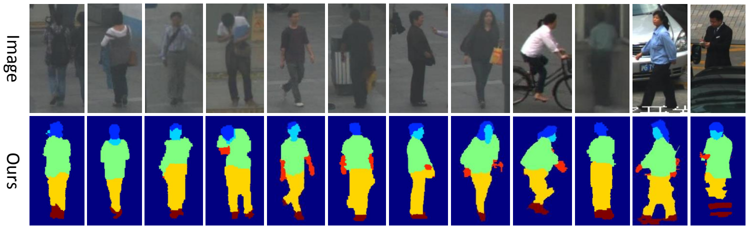

Figure 5: Image results from PPSS dataset. In the last image, we fail to parse the legs correctly because of the occlusion. (Best viewed in color)

Penn-Fudan	Hair	Face	UC	Arms	LC	Legs	BG
SBP[5]	44.9	60.8	74.8	26.2	71.2	42.0	81.0
DL[25]	43.2	57.1	**77.5**	27.4	**75.3**	**52.3**	**86.3**
Ours	**66.5**	**61.4**	74.8	**36.9**	74.0	43.8	84.6

PPSS	Hair	Face	UC	Arms	LC	Legs	BG
DDN[25]	35.5	44.1	68.4	17.0	**61.7**	23.8	80.0
Ours	**55.6**	**46.6**	**71.9**	**30.9**	58.8	**24.6**	**86.2**

Table 2: Accuracy of each semantic label on Penn-Fudan and PPSS pedestrian parsing dataset. "UC" stands for upper-clothes and "LC" for lower-clothes.

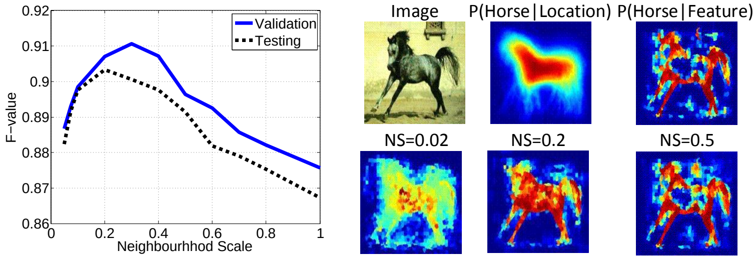

Figure 6: Illustration of influences of neighborhood scale (denoted by "NS" in the right figure) on Weizmann horse dataset. (Best viewed in color)

quality is quite good, even for some images' contrast being quite poor as in PPSS dataset. Table 2 shows that the parsing is reasonably good for upper-cloths, lower-clothes and background and worse for arms and legs. Arms and legs are subjected to more spatial variations and are much smaller in size. These make the parsing more challenging. Compared with [5] and [25], our method performs significantly better on hair and arms, with an improvement of more than 0.1 in IOU score.

5.2 Horse Segmentation

Our second experiment is to evaluate the performance on Weizmann horses dataset [6]. Although the horses are mostly located at the central portion of the image and facing left, they contain a large amount of appearance and pose variations especially for legs. This makes the dataset challenging if we want to correctly segment the details.

Neighborhood Scale. Figure 6 illustrates the influences of the neighborhood scale s as a percentage of the image size. We observe that as the s increases, the map transforms from

F-value	
Feature Only	83.9
$(\mathbf{f}, x, y)$ + SVM	83.9
$(\mathbf{f}, x, y)$ + Boosting	87.6
Product of Experts	86.5
Ours(nearest)	**90.1**
Ours(merged)	**90.1**

Method	F-value	Accuracy
[20]	-	95.5
[4]	-	94.6
[17]	-	94.7
[35]	84.0	-
[31]	89.9	95.4
Ours	**90.1**	**95.7**

Table 3: Benchmark results for Weizmann horse dataset.

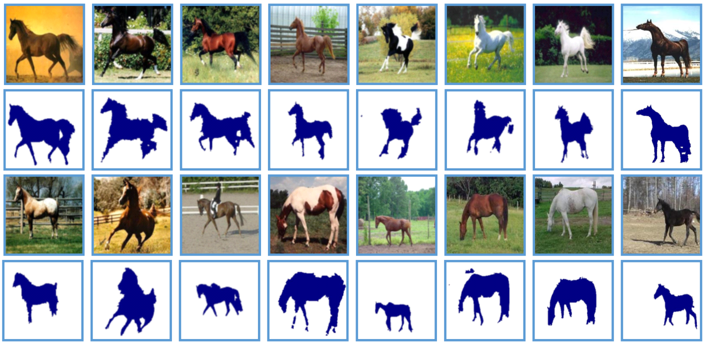

Figure 7: Image results from Weizmann horse. In the third and fourth row we show some examples that position, size or pose of the horse not following position prior.

$p(L \mid x,y)$ to $p(L \mid f)$ gradually. When the s is very small such as 0.02, location information is emphasized. However, because of the increased variance and limited training samples, the map also becomes quite noisy. On the other hand, if the neighborhood scale is too large such as 0.5, location information is "smoothed out" and consequently errors in the posterior map cannot be corrected. As a result, selection of a good neighborhood scale is important for good performance of our classifiers.

State of the Arts Comparison. Table 3 demonstrates that, on the Weizmann dataset our method performs much better than three alternative ways of fusion **f** and (x,y). In addition, our method has beaten the performance of five existing algorithms [4, 17, 20, 31, 35]. We also find it performs reasonably well even for cases that position, size, or pose of the horses does not follow position prior (see Figure 7). This demonstrates that our method is flexible for varied situations, and effective in handling misalignment of the location prior.

6 Conclusion

This paper proposes a novel way of fusing location and feature information for pixel labeling by local learning. Each local classifier is trained with pixels from its neighborhood region thus better fits the local distribution and is more discriminative. We analyze the bias-variance trade-off of our proposed local classifier, and indicate the importance of selecting an appropriate neighborhood size to train the local classifier such that it can tolerant the spatial misalignment. Our local learning scheme can accommodate any pixel classifiers with arbitrary features. In experiments we compare our method with alternative ways of fusion of feature **f** and location (x,y). The results validate that when properly trained, our proposed local classifier can be more effective than alternative ways that rely on global classifier. On both pedestrian parsing and Weizmann horse segmentation, our local learning can significantly improve the performance when compared with existing methods.

7 Acknowledgements

This work is supported in part by Nanyang Assistant Professorship SUG M58040015. We thank Dr. Gangqiang Zhao for extensive discussion and comments on the manuscript.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, 149300, 2010.

[2] Hamed Akbari and Baowei Fei. Automatic 3d segmentation of the kidney in mr images using wavelet feature extraction and probability shape model. In *SPIE Medical Imaging*. International Society for Optics and Photonics, 2012.

[3] Ayelet Akselrod-Ballin, Meirav Galun, Moshe John Gomori, Ronen Basri, and Achi Brandt. Atlas guided identification of brain structures by combining 3d segmentation and svm classification. In *MICCAI*. Springer, 2006.

[4] Luca Bertelli, Tianli Yu, Diem Vu, and Burak Gokturk. Kernelized structural svm learning for supervised object segmentation. In *CVPR*. IEEE, 2011.

[5] Yihang Bo and Charless C Fowlkes. Shape-based pedestrian parsing. In *CVPR*. IEEE, 2011.

[6] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV*. Springer, 2002.

[7] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI*, 26(9):1124–1137, 2004.

[8] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001.

[9] Pierrick Coupé, José V Manjón, Vladimir Fonov, Jens Pruessner, Montserrat Robles, and D Louis Collins. Patch-based segmentation using expert priors: Application to hippocampus and ventricle segmentation. *NeuroImage*, 54(2):940–954, 2011.

[10] Rémi Cuingnet, Raphael Prevost, David Lesage, Laurent D Cohen, Benoît Mory, and Roberto Ardon. Automatic detection and segmentation of kidneys in 3d ct images using random forests. In *MICCAI*. Springer, 2012.

[11] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[12] M Freiman, A Kronman, SJ Esses, L Joskowicz, and J Sosna. Non-parametric iterative model constraint graph min-cut for automatic kidney segmentation. In *MICCAI*. Springer, 2010.

[13] Helmut Grabner, Peter M Roth, and Horst Bischof. Is pedestrian detection really a hard task. In *IEEE Intern. Workshop on Performance Evaluation of Tracking and Surveillance*. IEEE, 2007.

[14] Tom Heskes. Bias/variance decompositions for likelihood-based estimators. *Neural Computation*, 10(6):1425–1433, 1998.

[15] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *TPAMI*, 26(2):147–159, 2004.

[16] Josip Krapac, Jakob Verbeek, and Frédéric Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*. IEEE, 2011.

[17] Daniel Kuettel and Vittorio Ferrari. Figure-ground segmentation by transferring window masks. In *CVPR*. IEEE, 2012.

[18] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*. IEEE, 2006.

[19] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001.

[20] Anat Levin and Yair Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV*. Springer, 2006.

[21] Dahua Lin and Jianxiong Xiao. Characterizing layouts of outdoor scenes using spatial topic processes. In *ICCV*. IEEE, 2013.

[22] Yen-Yu Lin, Jyun-Fan Tsai, and Tyng-Luh Liu. Efficient discriminative local learning for object recognition. In *ICCV*. IEEE, 2009.

[23] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric scene parsing via label transfer. *TPAMI*, 33(12):2368–2382, 2011.

[24] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60 (2):91–110, 2004.

[25] Ping Luo, Xiaogang Wang, and Xiaoou Tang. Pedestrian parsing via deep decompositional network. In *ICCV*. IEEE, 2013.

[26] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[27] Ingmar Rauschert and Robert T Collins. A generative model for simultaneous estimation of human body shape and pixel-level segmentation. In *ECCV*. Springer, 2012.

[28] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*. IEEE, 2014.

[29] Peter M Roth, Sabine Sternig, Helmut Grabner, and Horst Bischof. Classifier grids for robust adaptive object detection. In *CVPR*. IEEE, 2009.

[30] Laura Sevilla-Lara and Erik Learned-Miller. Distribution fields for tracking. In *CVPR*. IEEE, 2012.

[31] Mojtaba Seyedhosseini and Tolga Tasdizen. Scene labeling with contextual hierarchical models. *arXiv preprint arXiv:1402.0595*, 2014.

[32] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*. Springer, 2006.

[33] Joseph Tighe and Svetlana Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*. IEEE, 2013.

[34] Antonio Torralba, Kevin P Murphy, and William T Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*. IEEE, 2004.

[35] Zhuowen Tu and Xiang Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *TPAMI*, 32(10):1744–1757, 2010.

[36] Zhuowen Tu and Arthur W Toga. Towards whole brain segmentation by a hybrid model. In *MICCAI*. Springer, 2007.

[37] Jianxiong Xiao and Long Quan. Multiple view semantic segmentation for street view images. In *ICCV*. IEEE, 2009.

[38] Song-Chun Zhu, Cheng-En Guo, Yizhou Wang, and Zijian Xu. What are textons? *IJCV*, 62(1-2):121–143, 2005.