Combining Image-level and Segment-level Models for Automatic Annotation

Daniel Kuettel, Matthieu Guillaumin, and Vittorio Ferrari

Computer Vision Laboratory, ETH Zurich, Switzerland

Abstract. For the task of assigning labels to an image to summarize its contents, many early attempts use segment-level information and try to determine which parts of the images correspond to which labels. Best performing methods use global image similarity and nearest neighbor techniques to transfer labels from training images to test images. However, global methods cannot localize the labels in the images, unlike segment-level methods. Also, they cannot take advantage of training images that are only locally similar to a test image. We propose several ways to combine recent image-level and segment-level techniques to predict both image and segment labels jointly. We cast our experimental study in an unified framework for both image-level and segment-level annotation tasks. On three challenging datasets, our joint prediction of image and segment labels outperforms either prediction alone on both tasks. This confirms that the two levels offer complementary information.

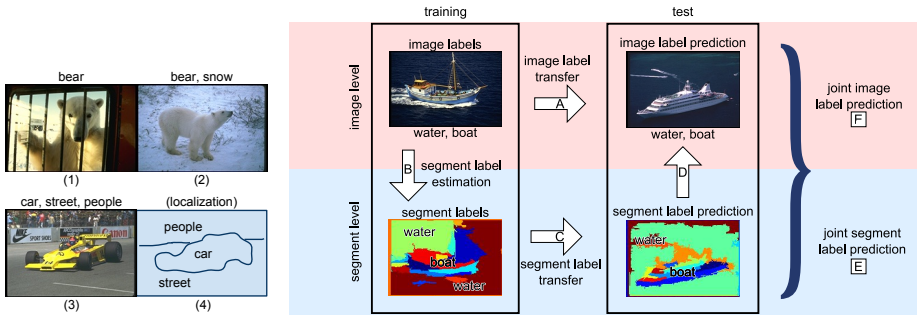
Keywords: image auto-annotation, image region labelling, keyword-based image retrieval

1 Introduction

In recent years, automatic image annotation has received increasing attention [11, 13, 17, 18]. In its basic version, which we call *image-level annotation*, the task is to assign a few semantic labels to a test image, roughly describing its contents (fig. 1(a)). In its elaborate version, which we call *segment-level annotation*, the semantic labels are assigned to every segment in the image (fig. 1(a)4). The union over the segment labels is then proposed as image labels [2, 4, 7].

Segment-level annotation poses additional challenges compared to image-level annotation. First, labels for the segments in the training images are not given, and must be estimated from the image labels. As a consequence, segment-levels methods need to be robust to errors in this estimation. Second, appearance features extracted from segments are less distinctive than global image features, which incorporate contextual layout information. Finally, even with perfect segment labels, their union does not always match user-provided image labels, since the latter focus on the salient objects in the image. Overall, segment-level annotation is a much more difficult task, which explains why recent global methods outperform local ones for image-level annotation.

On the other hand, global methods cannot *localize* labels in the test images, but merely indicate their presence (fig. 1(a)3). This limits the interpretability of the different methods and reduces the spectrum of possible applications of the output predictions: image labels are restricted to classification and indexing purposes. With localized labels instead, it is possible to visualize the learned concepts and identify their spatial extent in



(a) (b)

Fig. 1. Left (a): A test image (1) of a bear out of its typical context in the wild (2), highlighting the need for compositionality. On the other hand, context is a powerful force for recognizing cars in typical images such as (3). (4) shows a localization of the labels in (3). Right (b): **Summary of image annotation models.** For each arrow there are several applicable models. Alternatives are discussed in the respective sections. For E and F, we present novel methods to combine segment and image-level models.

the images. Therefore, segment labels can be used to train object detectors or compute class-specific features invariant to position and scale. Overall, they provide a deeper understanding of an image.

Our work builds on the observation that image-level and segment-level techniques have several complementary strengths. Segment-level methods explicitly attempt to determine which parts of the training images belong to each label. This is typically done by describing the local appearance of segments and then searching for recurrences over the training set with a probabilistic model [2, 3, 5, 9, 19]. Segment-level methods can recognize the presence of a class in a test image even if it appears in a context not observed during training (*e.g.* a bear in a cage while training images show bears in the wild, fig. 1(a)1+2). This *compositional* character is a strength of segment-level methods and endows them with great generalization potential. On the other hand, the global image layout is more characteristic than the appearance of individual segments, as it indicates certain combinations of labels (cars-roads in fig. 1(a)3). Recent image-level methods [17, 25] employ global image similarities and predict labels for a test image based on the labels of its most similar training images. Those methods perform better on the image-level annotation task [1, 11], as they better exploit the large number of available images annotated by keywords.

The observations above suggest that segment-level prediction is a task of its own, which should be evaluated on a per-pixel basis, and that combining segment-level and image-level predictions may help both tasks. The potential for interaction between the two levels is largely unexplored and very promising. Image labels help reduce the space of possible segment-level annotations. On the other hand, even imperfect segment labels carry valuable complementary information about image content.

In this paper we explore the combination of image and segment levels and make the following contributions: (i) we present a unified view of existing methods as processing stages in a generic scheme (sec. 2); (ii) we propose new alternative models to perform many of the stages (sec. 3 to 6); (iii) we propose novel joint models to combine the predictions from image and segment levels (sec. 7). In sec. 8 we present the datasets and features we used. Through extensive experiments, we demonstrate that our combined models perform better at both segment-level and image-level annotation than

either component alone (sec. 9). We conclude and draw directions for future research in sec. 10.

Related works. Our work relates to the numerous segment-level and image-level methods discussed above, as we seek to combine the two strands.

Some earlier works tried to incorporate context in segment-level methods, *e.g.* by modeling co-occurence of labels [6] or their spatial relationships [23]. However, these methods typically do not use global image predictions. Most importantly, their training scenarios are radically different from ours, where ground-truth segment labels are available at training time. Therefore, they address a different task, known as *semantic segmentation* in the literature [14, 20], which can be seen as the fully supervised version of segment-level annotation.

Note how several earlier methods proposed for image label prediction actually perform segment-level annotation. Early methods based on probabilistic models [2, 5, 19] describe the image as an orderless bag of segments. Non-parametric mixture models like multiple bernoulli relevance models [9] also rely on image regions.

2 Models Overview

Before investigating ways to combine segment-level and image-level information, we present a unified view which incorporates most previous works. Fig. 1(b) shows the two main existing ways to obtain predictions on a test image using image-level (arrow A) or segment-level methods (sequence of arrows B-C-D). Image-level methods [1, 11, 17, 25] directly transfer labels from training images to test images using global image similarities (A). Segment-level methods [2–5, 9, 19] first estimate labels for the segments in the training images (B), then transfer them to the segments in the test image (C). Finally, they derive a prediction of image labels from these predicted segment labels (D).

In the following sections, we first present various alternatives for the components in fig. 1(b) (arrows), including new ones that we propose. We then present novel methods to combine segment and image-level models in sec. 7 (stages E and F) .

3 Image Label Transfer (A)

Transferring labels from training images to test images is the most direct way to predict image labels. This strategy has recently been shown to be very successful [1, 11, 17].

Formally, let $\mathcal{I}$ be the set of N training images I_i. The *dictionary* $\mathcal{D}$ is the set of unique labels in the annotations of the training images. There are V labels in $\mathcal{D}$ and we refer to them by their id $l \in \{1..V\}$. Each training image is annotated with labels from $\mathcal{D}$. We summarize the annotation as L_l, which is an indicator function for label l. If image I_i is annotated with label l, then $L_l(I_i) = 1$, and 0 otherwise.

Here, we focus on the recent, state-of-the-art TagProp [11]. which transfers labels using a weighted nearest neighbor approach, but other works fall in this category (A) [17, 25].

3.1 TagProp

The label prediction $L_l(Y)$ for a test image Y is based on a weighted sum over the training images:

$$\text{tagprop}_l(Y) = p(L_l(Y)|\mathcal{I}) = \sum_{i=1}^{N} \pi_{yi} p(L_l(I_i)) \quad (1)$$

Where $p(L_l(I_i)) = 1 - \epsilon$ for $L_l(I_i) = 1$, ϵ otherwise. In [11] several variants for π_{yi} are presented. We summarize here the best performing variant, which produces state-of-the-art results. Specifically, the weights π_{yi} are

$$\pi_{yi} = \frac{\exp(-d_w(Y,I_i))}{\sum_j \exp(-d_w(Y,I_j))} \text{ with } d_w(Y,i) = \mathbf{w}^T \mathbf{d}_{yi} \quad (2)$$

where $\mathbf{d}_{yi}$ is a vector of base distances between Y and I_i. A separate base distance is computed for each type of image feature and $\mathbf{w}$ is a vector of positive coefficients for combining these distances. This variant is called *ML*, for metric learning, because $\mathbf{w}$ is learned so as to maximize the log-likelihood $\mathcal{L}$ of the leave-one-out predictions on the training set

$$\mathcal{L} = \sum_{i,l} c_{il} \ln p(L_l(I_i)|\mathcal{I} \backslash I_i) \quad (3)$$

where $\mathcal{I} \backslash I_i$ is the set of training images without I_i, and c_{il} is a reweighting parameter for labels. It gives more weight to present labels than to absent ones since the absence of labels in the annotation is less reliable information [11]. As the log-likelihood (3) is concave, we maximize it using a projected-gradient algorithm. The first derivative of eq. (3) with respect to $\mathbf{w}$ is

$$\frac{\delta \mathcal{L}}{\delta \mathbf{w}} = \sum_{i,j} W_i(\pi_{ij} - \rho_{ij}) d_{ij} \text{ with } \rho_{ij} = \sum_l \frac{c_{iw}}{W_i} p(L_l(I_j)|L_l(I_i)) \quad (4)$$

This learning step was shown by [11] to outperform earlier, ad-hoc ways to transfer labels from image neighbors [17]. Note that, in order to keep learning efficient, the $\mathbf{d}_{yi}$ are only computed for the K nearest neighbors (typically 200) of Y in $\mathcal{I}$. We set $\pi_{yi} = 0$ for all others.

Weighted nearest neighbor models tend to have low recall, since rare labels are unlikely to appear in many neighbor images. Therefore, [11] further adds a word-specific logistic discriminant model to boost the probability for rare labels:

$$p(L_l(Y)|\mathcal{I}) = \sigma(\alpha_l x_{yl} + \beta_l) \text{ with } \sigma(z) = (1 + \exp(-z))^{-1} \quad (5)$$

$$x_{yl} = \sum_i^N \pi_{yi} p(L_l(I_i)) \quad (6)$$

The parameters (α_l, β_l) and $\mathbf{w}$ are learned in alternating fashion to maximize eq. (3). See [11] for details.

4 Segment Label Estimation (B)

We discuss here models to estimate segment labels from image labels during training (fig. 1(b), arrow B). This stage is necessary since only ground-truth image labels are available for training. Estimating segment labels from image labels can be seen under different points of view: as a Multiple Instance Learning problem [12] where an image forms a bag of instances (segments); as a constrained clustering problem [7]; or the missing segment labels can be recovered by MRFs [21]. The same task is also referred to as the *Label-to-Region* problem by a few authors [16].

Formally, the task is to estimate the labels of every segment $s \in \mathcal{S}_i$ in every training image I_i, guided by the given image labels $L_l(I_i)$. This involves estimating the probability $p(L_l(s)|\{\mathcal{S}_i\}, \mathcal{I})$ of $L_l(s) = 1$ for every label l and segment s in every image i. We present below three alternative approaches for this task (either one can be used).

4.1 Label Copy

As a straightforward approach, labels can be simply copied from an image to its segments. In this case, all segments in an image are assigned the same labels. We obtain the following expression for the segment labels

$$p(L_l(s)|\{\mathcal{S}_i\},\mathcal{I}) = L_l(I_i). \tag{7}$$

This is a conservative approach. It contains noise for the presence of a label, but almost none for the absence of a label. Some methods for segment label transfer (C) are very robust to label presence noise and perform surprisingly well with label copy.

4.2 Token Model

This model represents segments by visual words as in [7]. All N_s segments are collected in the set $\mathcal{S} = \cup_i \mathcal{S}_i$. We describe the appearance of each segment $s_j \in S$ with a feature vector f_j (sec. 9) and then apply k-means to all vectors to obtain Q cluster centers c_q. Each c_q is a *visual word* and $\mathcal{C} = \cup_q c_q$ is the *codebook* of visual words. We now assign each segment s_j to its closest cluster center c_q and denote the id q as the *token* $T(s_j)$ of s_j. The Token Model represents segments solely by their token. This turns the estimation of $p(L_l(s)|\{\mathcal{S}_i\},\mathcal{I})$ into

$$p(L_l(s)|\{\mathcal{S}_i\},\mathcal{I}) = p(L_l(T(s))|\{T(\mathcal{S}_j)\},\mathcal{I}). \tag{8}$$

Representing a segment as a token rather than a feature vector is beneficial because tokens are discrete and finite, whereas feature vectors live in a continuous and typically high-dimensional space. Therefore, estimating (8) is easier than estimating the distribution $p(L_l(s)|\{\mathcal{S}_i\},\mathcal{I})$ directly.

In the spirit of [7], we adopt a simple clustering approach, which assigns exactly one label z_{ij} to each segment s_{ij} of image I_i

$$L_l(s_{ij}) = \begin{cases} 1 \text{ if } l = z_{ij} \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

From a given segment-label assignment z we derive the empirical label-token distribution

$$p(L_l(t)|t,z) = Z \sum_{ij}^{T(s_{ij})=t} L_l(s_{ij}), \tag{10}$$

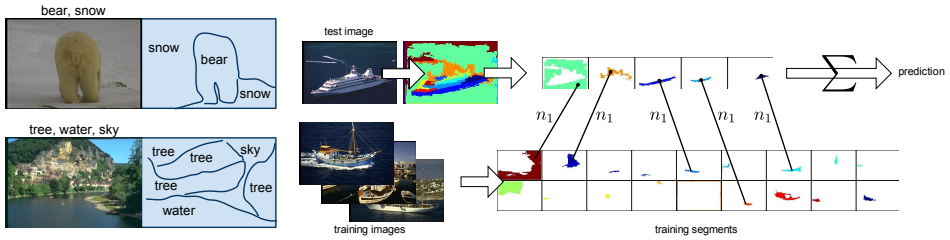
where Z is the normalization factor and t is a token.

To learn the labeling we use an EM-like scheme. We initialize z_{ij} with a random label of image I_i. In the first step, the probability in eq. (10) is estimated using the last assignments z_{ij}. In the second step, z_{ij} are estimated using eq. (10) (keeping them restricted to the labels $L_l(I_i)$ of the ground-truth image labels). The steps are repeated until convergence.

4.3 Label-to-Region (LTR)

This is the approach described in the recent work of [16]. It consists of two stages. First, corresponding segments between image with common labels are found. Second, labels are assigned to segments based on these correspondences.

In the first stage, a segment s in an image I_i is approximated in the feature space as a sparse linear combination of segments $s' \in \mathcal{S}'$ in other images $\mathcal{I} \backslash I_i$ sharing at least one label. Then, labels are transferred to s from $\mathcal{S}'$ according to the sparse linear



(a) (b)

Fig. 2. Left (a): Example segment label estimations on two training images (ground-truth annotated only at the image level). Right (b): **The Global Segprop model.** The prediction for a test image (top) is a mixture over the nearest neighbors of the image's segments (center, shown with lines) in the training set (bottom). For clarity, only the first nearest neighbor n_1 of each segment is shown.

combination. This scheme is repeated for all segments until convergence. The initial labels for the segments are copied from the image, as in Label Copy (sec. 4.1). For each segment, this stage returns a probability vector over labels (multinomial distribution).

In the second stage, labels are assigned to segments. For each image, the probability vectors of the segments are clustered into as many clusters as there are labels for the image. The resulting clusters are then labeled with the most likely label according to the centroid. Finally, each segment is given the label of the its cluster.

5 Segment Label Transfer (C)

We present here two alternatives for transferring labels from training segments to segments in a test image Y. While this is not as direct as image-level predictions (A), it is more flexible as it can explain the test image as a combination of segments not observed during training. At this stage, segment labels on the training set have already been derived from ground-truth image labels (B). Throughout this section, $\mathcal{S}$ is the set of segments s_i in the training set.

5.1 Token Model

The Token Model trained in (B) is directly applicable to test images. We apply to each test image segment y the quantization procedure described in sec. 4.2 and obtain its token $t = T(y)$. Then, the multinomial distribution $p(L_l(t)|t)$ in (10) is used to predict the label of y

$$\text{tokenmodel}_l(y) = p(L_l(t)|t) \propto \sum_{s \in S}^{T(s)=t} L_l(s). \tag{11}$$

For any given token, this is the vector of frequencies of estimated segment labels in the training set.

5.2 SegProp

As a novel alternative to the Token Model, we propose here an approach analog to TagProp (sec. 3) to transfer labels from training segments to test segments. We refer to it as SegProp, for Segment-level Propagation. The output of SegProp for label l for a test image segment y is

$$\text{segprop}_l(y) = p(L_l(y)|\mathcal{S}) = \sum_{i=1}^{N_s} \pi_{yi} p(L_l(s_i)), \tag{12}$$

where $p_k(L_l(s)) = 1 - \epsilon$ for $L_l(s) = 1$, ϵ otherwise. Therefore, the label prediction of a segment is a weighted sum over the training segments s_i. As in sec. 3, we restrict ourselves to the K nearest neighbors, set $\pi_{yi} = 0$ for all others, and use the same projected-gradient method to learn this model. Note that, for a test segment y, SegProp outputs a vector of probabilities with one entry per label (*e.g.* $[p(L_1(y)) \dots p(L_V(y))]$).

6 Image Labels from Segment Predictions (D)

The last stage of predicting image labels using segments is to transfer labels to the image from the predicted labels of its segments. When each segment label is predicted as a multinomial or multiple Bernoulli distributions, it is natural to combine them, for instance using a mixture model. We detail two alternatives below. Let Y denote a test image and $\{y_r\}$ the set of its segments.

6.1 Maximum Prediction

In this approach, we combine segment-level predictions into an image-level one by keeping, for each label, the largest prediction over the segments. This procedure takes advantage of the compositionality of segments. If two regions are predicted to have different labels, it indeed transfers both labels to the image. Formally, we define:

$$p(L_l(Y)|\{y_r\}) = \max_r p(L_l(y_r)). \tag{13}$$

6.2 Global SegProp

Instead of considering each segment to have the same importance in the final prediction, an alternative is to use a mixture over the segments. This is the base of our new Global SegProp model. Specifically, Global SegProp outputs an image-level prediction as a mixture of the labels of the training neighbors of its R largest segments $\{y_r\}$ (largest area relative to image):

$$p(L_l(Y)|\{y_r\}) = \sum_{i=1}^{N_s} \pi_{yi} p(L_l(s_i)) \tag{14}$$

Where $p(L_l(s)) = 1 - \epsilon$ for $L_l(s) = 1$, ϵ otherwise. The components for $\mathbf{d}_{yi}$ (see eq. (2)) are the feature space distances for segment s_i to the R largest segments $\{y_r\}$. As before, we compute the K nearest neighbors for every of the R largest segments, take the union set, and set $\pi_{yi} = 0$ for segments not in this set.

Importantly, the weights are now optimized for image-label prediction during training, whereas SegProp optimizes them for segment-label prediction. Hence, this model perform stages (C) and (D) jointly (fig. 2(b)).

7 Joint Label Prediction

In this section we propose several models for combining the image and segment levels for predicting labels of a test image Y. This is desirable as the information that the two levels offer is orthogonal. The global, image-level models are more distinctive because they capture context. The local, segment-level models are more flexible thanks to compositionality. Moreover, they can annotate the test image at the segment level. By doing the prediction jointly, we can hope to bring some contextual information into the segment-level predictions as well as improving image annotation by exploiting compositionality.

We devise three alternatives to combine TagProp (A) with segment-level predictions (C), for achieving both segment-level prediction (E) and image-level prediction (F). The

Table 1. Summary of pixel annotation results on the MSRC-21 dataset.

Name (Parameters)	A	B	C	E	Overall acc.
Token Model ($Q=2300$)	-	Token	Token	-	24.4%
SegProp ($Q=2300, K=50$)	-	Token	SegProp	-	25.6%
SegProp ($K=50$)	-	LTR	SegProp	-	29.6%
SegProp ($K=50$)	-	Copy	SegProp	-	31.4%
TagProp+Token	TagProp	Token	Token	Prod.	27.8%
TagProp+SegProp	TagProp	Copy	SegProp	Prod.	**33.8%**

first two are rather simple and based on multiplying the output probabilities (sec. 7.1 and 7.2). Last, we propose a more complex one, based on combining neighborhoods of image-level and segment-level models (sec. 7.3).

7.1 Joint Segment-level Prediction by Product (E)

In this joint model, the image-level prediction acts as a prior to guide the segment-level prediction. To include the prediction for image Y to predict its segment y_i, we compute $p(L_l(y_i)|Y)$ as:

$$p(L_l(y_i)|Y) = p(L_l(Y))p(L_l(y_i)), \tag{15}$$

where $p(L_l(Y))$ is the output of any image-level method (A), and $p(L_l(y_i))$ of any segment-level prediction (C).

For (A), we have only considered TagProp, so $p(L_l(Y)) = \text{tagprop}_l(Y)$. For (C), $p(L_l(y_i))$ can be set to either $\text{tokenmodel}_l(y_i)$ or $\text{segprop}_l(y_i)$ (sec. 5), leading to combinations that we refer to as "TagProp×Token" and "TagProp×SegProp".

7.2 Joint Image-level Prediction by Product (F)

In order to achieve the effect of improving image-level prediction using segment-level prediction, we propose to combine the output of any image-level method (A) with the image-level prediction (D) corresponding to a segment-level method (C):

$$p(L_l(Y)|\{y_i\}, Y) = p(L_l(Y)|Y)p(L_l(Y)|\{y_r\}). \tag{16}$$

Again, TagProp will be used for $p(L_l(Y)|Y)$, while $p(L_l(Y)|\{y_r\})$ can be obtained by Maximum Prediction (D) from any segment-level method, or by using Global SegProp (sec. 6.2). As in the previous section, we refer to these as "TagProp×Token" and "TagProp×SegProp".

7.3 Tagprop + Global SegProp (F)

We propose a novel and more elaborate technique to predict image labels by combining image-level and segment-level information. We include both segment neighbors (as in Global Segprop) and image neighbors (as in Tagprop)

$$p(L_l(Y)|\{y_r\}, \mathcal{I}) = \sum_{i}^{N_s} \pi^s_{yi} p(L_l(s_i)) + \sum_{i}^{N} \pi^I_{yi} p(L_l(I_i)) \tag{17}$$

Note that there are two sets of weights, π^S for segment neighbors, and π^I for image neighbors. By fixing one set of weights, we can maximize the log-likelihood over the other set as done for eq. (3). So, we learn both sets in alternation. As done in sec. 3.1, for efficient learning we only consider the K nearest neighbors of Y for image neighbors. For segment neighbors, we include the T nearest neighbors for each of the R top largest segments in Y. In total, there are $K + RT$ neighbors. We set to 0 the π weights for training images/segments not in this set.

8 Data Sets and Features

In this section, we describe the datasets we experiment on, and the image/segment features we use. Note that to properly evaluate our approaches on segment-level annotation from image labels, datasets with ground-truth pixel annotation are required (MSRC-21, SIFT-Flow).

The MSRC-21[1] dataset contains 591 images of 23 object classes, annotated at the pixel level. We adopt the evaluation protocol of [21] and keep the 21 most frequent classes and *void*, leaving *horses* and *mountain* out. As in [21, 24], we use a random selection of 531 images for training and the other 60 for testing.

The SIFT-Flow[2] dataset [15] contains 2688 images with a total of 33 objects and background classes annotated at the pixel level (*sky*, *sea*, etc.). We use the training and test subsets defined in [15], with 200 images for testing and the rest for training.

The Corel 5k[3] dataset [7] is commonly used for image auto-annotation. It comes with pre-defined training and test images that have been manually labeled with at most 5 keywords out of a vocabulary of 260. The training set consists of 4500 images while the test set has 499 images, which we use to evaluate image-level prediction. There is no pixel-level annotation for this dataset.

To describe images globally, we adopt the features of [11]. They consist of GIST, color histograms (RGB, LAB, HSV) with 16 bins per channel, and bag-of-features histograms. For the latter, SIFT and Hue [22] descriptors are computed on a multiscale grid of points and at Harris interest points. These descriptors are quantized using K-means with 1000 centroids for SIFT and 100 for Hue. Additionally, histograms over three horizontal regions are also computed for all descriptors except for GIST. This results in 15 different descriptors. For the base distances, we use $L2$ for GIST, $L1$ for color, χ^2 for bag-of-features.

For segments, we adapt the descriptors described above. First, color histograms are computed with only 12 bins per channel to reduce the dimensionality. Quantized local descriptors are accumulated in individual histograms of segments based on the location of the interest points. In total, there are 7 descriptors. The base distances are computed analog to the image-level case. For the Token Model, we have reimplemented the segment features of [2]: relative size and position in the image, average and standard deviation of pixel RGB and LAB, and shape features such as ratio of area to perimeter, eccentricity and ratio of area to convex hull. Here, $L2$ is used as a distance measure. Our segments are computed using [8].

9 Experimental Evaluation

We present here the experimental protocols and our results for both segment and image label prediction tasks.

Segment-level prediction. Segment-level prediction is evaluated using a standard measure for semantic segmentation [15, 20, 21]: the percentage of correctly predicted pixels over all pixels (*overall pixel accuracy*).

[1] `http://research.microsoft.com/en-us/projects/objectclassrecognition/`

[2] `http://people.csail.mit.edu/celiu/CVPR2009/`

[3] `http://kobus.ca/research/data/eccv_2002/`

Table 2. Pixel annotation results on the SIFT-Flow dataset.

Name (Parameters)	Overall acc.
Token Model ($Q=2300$)	18.5%
SegProp ($K=50$)	34.2%
TagProp+Token	31.1%
TagProp+SegProp	**35.9%**

Table 3. Image annotation results on the Corel5k dataset. TagProp is abbreviated as TP and SegProp as SP.

Name (Parameters)	A	B	C	D	E	BEP
Token Model ($Q=2300$)	-	Token	Token	Max.	-	8.2%
SP ($Q=2300, K=50$)	-	Token	SP	Max.	-	11.2%
SP ($K=50$)	-	Copy	SP	Max.	-	14.9%
Global SP ($R=10, K=5$)	-	Copy	Global SP		-	19.8%
TP ($K=200$)	TP	-	-	-	-	36.2%
TP+Token	TP	Token	Token	Max.	Prod.	22.2%
TP+SP	TP	Copy	SP	Max.	Prod.	27.9%
TP+Global SP ($K=200, R=10, T=5$)	-	Copy	-	-	TP+G SP	**37.0%**

In tab. 1, we summarize the different methods that we compare for segment-level annotation on the MSRC-21 dataset. The Token Model achieves an overall accuracy of 24.4%. Our proposed SegProp model performs considerably better, reaching 29.6% in conjunction with LTR for stage B, and 31.4% with the simple label copy mechanism for stage B. As SegProp is very robust to the presence of label noise, it performs well in conjuction with label copy.

More importantly, when combining the segment-level predictions with image-level predictions from TagProp, we obtain significant improvements: +2.2% for SegProp and +3.4% for the Token Model. The larger improvement for the Token Model can be explained by the higher complementarity of the methods and features, compared to SegProp. Our TagProp+SegProp combination achieves the best overall accuracy of 33.8%.

In tab. 2, we give the accuracy on the SIFT-Flow dataset. The same conclusions can be drawn: SegProp is superior to the Token Model for segment-level annotation, and the combination with TagProp improves both models. In fig. 3, we illustrate the benefit of using image-level prediction to guide segment-level prediction.

Note that several works [15, 20] report higher scores than ours for both datasets. However, they operate in the *fully supervised* scenario, i.e. using ground-truth pixel labels for training, whereas we use *only image labels*. Those methods are able to train strong appearance classifiers, and can leverage position and smoothness priors.

Image-level prediction. Following previous works [10, 11], we measure the Break-Even Point score (BEP). To compute the BEP, first the images are ordered by the predicted probability for a label l. This list is truncated to the length of the true number of relevant images (using ground-truth). The BEP measures the percentage of relevant images in this truncated list, averaged over all labels $l = 1 \dots V$. Some works [7, 11, 17] additionally measure precision/recall after assigning the 5 highest-scoring labels to each test image. However, as many test images have fewer than 5 ground-truth labels, the algorithm performance is incorrectly penalized. As a result, the maximum achievable

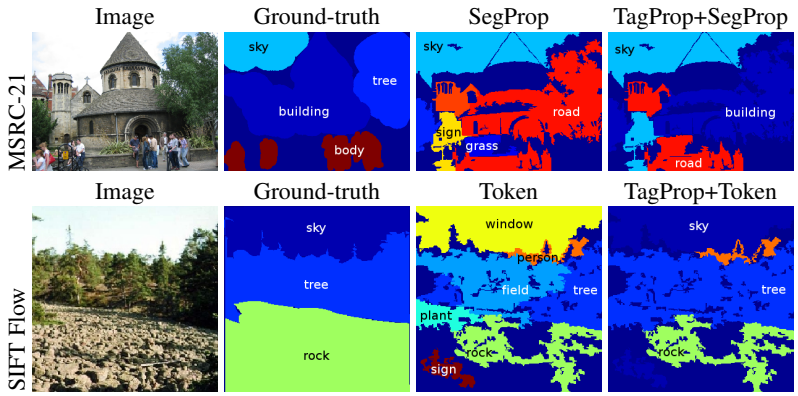


Fig. 3. Example images from the MSRC-21 (top row) and SIFT Flow (bottom row) data set. The first column shows a test image for each. The ground-truth segmentations with their labels are shown in the second column. The last two columns highlight the benefits of using image-level predictions to help segment level prediction. Label predictions using SegProp and TagProp+SegProp (top row), Token and TagProp+Token respectively (bottom row), are shown. In both cases, the combined method improves over the segment-level one.

precision is not 100%. We report BEP scores and agree with [10, 11] that they are more meaningful.

Tab. 3 summarizes the performance of the methods we compare on the Corel5k dataset. The Token Model achieves a low performance of 8.2%, in line with the published results of a similar model [2]. As in the segment-level evaluation, our SegProp model improves over the Token Model for stage C and reaches 11.2%. Moreover, the gain is higher when using label copy in stage B: 14.9%. Further improvement is obtained by fusing the C and D stages in our newly proposed Global SegProp model: 19.8%.

As the 'TagProp' row shows, consistent with previous observations [11, 17], directly predicting image labels using a global similarity outperforms segment-level methods on this task. Note that our result of 36.2% using TagProp with $K = 200$ closely matches the best variant of TagProp reported in [11] (36.3%).

Our integrated TagProp+Global SegProp method brings a large improvement over Global SegProp (+17.2%). Importantly, it also improves over state-of-the-art TagProp alone. Therefore, our method also improves over other works such as [9, 13], which were outperformed by TagProp (see scores for MBRM or TGLM within [11]).

10 Conclusion

We have presented a unified view on image-level and segment-level methods, where existing works can be casted in a common framework. We have proposed new models for some of the stages and, importantly, novel models to perform joint prediction on both levels.

We have conducted extensive experiments on two challeging data sets for pixel-level annotation and on a third one for image-level annotation. Our evaluation confirms that combining image-level and segment-level models brings better results than either model alone, on both tasks. The improvement is particularly strong for the segment labeling task. This shows that both levels have complementary strengths. Finally, note

that our combined method TagProp+SegProp performs *both tasks at the same time*. It labels both the pixels and the whole image, unlike TagProp and image-level methods in general, which only deliver image labels.

References

1. Babenko, B., Branso, S., Belongie, S.: Similarity metrics for categorization: from monolithic to category specific. In: ICCV (2009)
2. Barnard, K., Duygulu, P., de Freitas, N., Forsyth, D., Blei, D., Jordan, M.: Matching words and pictures. JMLR (2003)
3. Barnard, K., Fa, Q., Swaminatha, R., Hoog, A., Collin, R., Rondo, P., Kaufhold, J.: Evaluation of localized semantics: data, methodology, and experiments. IJCV (2007)
4. Barnard, K., Forsyth, D.A.: Learning the semantics of words and pictures. In: ICCV (2001)
5. Blei, D., Jordan, M.: Modeling annotated data. In: Proceedings of the ACM SIGIR conference (2003)
6. Choi, M., Lim, J., Torralba, A., Willsky, A.: Exploiting hierarchical context on a large database of object categories. In: CVPR (2010)
7. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: ECCV (2002)
8. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV 59(2) (2004)
9. Feng, S., Manmatha, R., Lavrenko, V.: Multiple Bernoulli relevance models for image and video annotation. In: CVPR (2004)
10. Grangier, D., Bengio, S.: A discriminative kernel-based model to rank images from text queries. PAMI 30(8), 1371–1384 (2008)
11. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation. In: ICCV (2009)
12. Jin, R., Wang, S., Zhou, Z.H.: Learning a distance metric from multi-instance multi-label data. In: CVPR (2009)
13. Li, J., Li, M., Liu, Q., Lu, H., Ma, S.: Image annotation via graph learning. Pattern Recognition 42(2), 218–228 (2009)
14. Lim, Y., Jung, K., Kohli, P.: Energy Minimization Under Constraints on Label Counts. In: ECCV (2010)
15. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR (2009)
16. Liu, X., Cheng, B., Yan, S., Tang, J., Chua, T., Jin, H.: Label to region by bi-layer sparsity priors. In: ACM Multimedia (2009)
17. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: ECCV (2008)
18. Me, T., Wan, Y., Hu, X., Gon, S., Li, S.: Coherent image annotation by learning semantic distance. In: CVPR (2008)
19. Monay, F., Gatica-Perez, D.: PLSA-based image auto-annotation: constraining the latent space. In: ACM Multimedia. pp. 348–351. ACM (2004)
20. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV (2006)
21. Verbeek, J., Triggs, B.: Region classification with Markov field aspect models. In: CVPR (2007)
22. van de Weijer, J., Schmid, C.: Coloring local feature extraction. In: ECCV (2006)
23. Yuan, J., Li, J., Zhang, B.: Exploiting spatial context constraints for automatic image region annotation. In: ACM Multimedia (2007)
24. Zha, Z., Hua, X., Mei, T., Wang, J., Qi, G., Wang, Z.: Joint multi-label multi-instance learning for image classification. In: CVPR (2008)
25. Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006)