Semantic Component Analysis

Calvin Murdock
Machine Learning Department
Carnegie Mellon University
cmurdock@cs.cmu.edu

Fernando De la Torre
The Robotics Institute
Carnegie Mellon University
ftorre@cs.cmu.edu

Abstract

Unsupervised and weakly-supervised visual learning in large image collections are critical in order to avoid the time-consuming and error-prone process of manual labeling. Standard approaches rely on methods like multiple-instance learning or graphical models, which can be computationally intensive and sensitive to initialization. On the other hand, simpler component analysis or clustering methods usually cannot achieve meaningful invariances or semantic interpretability. To address the issues of previous work, we present a simple but effective method called Semantic Component Analysis (SCA), which provides a decomposition of images into semantic components.

Unsupervised SCA decomposes additive image representations into spatially-meaningful visual components that naturally correspond to object categories. Using an overcomplete representation that allows for rich instance-level constraints and spatial priors, SCA gives improved results and more interpretable components in comparison to traditional matrix factorization techniques. If weakly-supervised information is available in the form of image-level tags, SCA factorizes a set of images into semantic groups of superpixels. We also provide qualitative connections to traditional methods for component analysis (e.g. Grassmann averages, PCA, and NMF). The effectiveness of our approach is validated through synthetic data and on the MSRC2 and Sift Flow datasets, demonstrating competitive results in unsupervised and weakly-supervised semantic segmentation.

1. Introduction

In the last decade, image classification has become an incredibly active research topic with widespread applications. Most methods for visual recognition are fully-supervised and make use of bounding boxes or pixel-wise segmentations to locate objects of interest. However, this type of manual labeling is time consuming, error-prone, and potentially suboptimal [29]. On the other hand, the increasing prevalence of large image collections emphasizes the need for fully- or partially-automated techniques for analyzing and archiving their content.

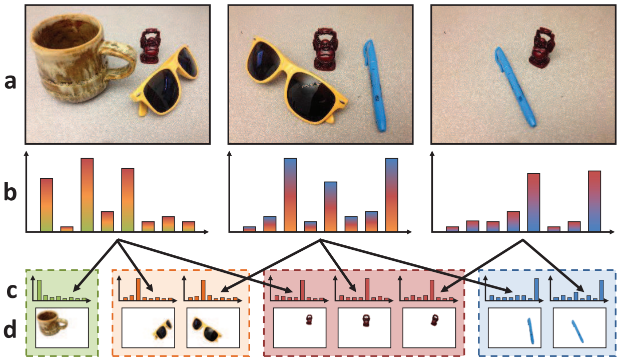


Figure 1. An overview of Semantic Component Analysis (SCA) applied to the task of unsupervised object discovery. (a) From a set of images containing multiple classes, (b) Bag-of-Words features are extracted pooling information from the entire image. (c) SCA decomposes these global representations into component histograms associated with meaningful component objects. The segments corresponding to these object histograms are shown in (d).

Real-world images are often composed of a number of distinct (but semantically-related) regions. A natural aim of visual learning is to find these meaningful regions in an unsupervised or weakly-supervised manner. For instance, consider Fig. 1(a): it is clear that there are four component objects that can explain the given images. The question is how to recover these semantic components with minimal supervision. Algorithms that approach this problem face many challenges, primarily in dealing with large intra-class variability in appearance, illumination, and pose.

A generative model for image formation can be considered as mixing a number of semantic components: one for each class present within an image. While the same local image features (e.g. quantized sift descriptors) may appear in instances from different classes, the *distributions* of features within semantic regions are often distinct across classes. If these global image features could be unmixed into their semantic components–each representing consistent segmentations belonging only to a single class–then recognition tasks could be simplified dramatically. This problem motivates a component analysis (CA) approach to image understanding in which an image is decomposed into *semantic* components.

Image decomposition is often accomplished through matrix factorization techniques, such as Principal Component Analysis (PCA) [38], Non-negative Matrix Factorization (NMF) [21], or Probabilistic Latent Semantic Analysis (pLSA) [36]. These methods approximate data as linear combinations of latent factors by minimizing total reconstruction error. While some variations of these approaches can result in localized, semantically-meaningful, or parts-based image decompositions, they are generally unable to adhere to a key property of image formation: objects are *occlusive*, i.e. image formation is nonlinear in pixel space because an object occludes everything behind it. Thus, images tend to consist of contiguous groups of pixels that belong only to a single object class. On the other hand, matrix decompositions represent each pixel as a superposition of multiple components. Since they rely on a *shared* basis that only *approximates* the original data, modifying these methods to enforce semantically-meaningful components by incorporating such nonlinear pixel-level constraints with real-word, unaligned images is nontrivial.

This paper introduces Semantic Component Analysis (SCA), a novel method for visual data decomposition that finds semantic factorizations of visual data. Fig. 1 illustrates SCA applied to Bag-of-Words (BoW) histograms extracted from input images. Our algorithm decomposes these global image features into class-specific histograms (Fig. 1c) constructed from partitions of semantically-related image segments (Fig. 1d). While existing factorization methods use a global basis common to all images, the key idea of SCA is the introduction of instance-specific sets of components allowing for more complex image constraints and priors. Specifically, we enforce that object partitions be spatially-consistent. This type of coherence would not be not possible with a global basis because instances of the same class vary in appearance and location across images.

For completeness, we analyze the relationship between SCA and existing techniques for traditional component analysis, empirically showing qualitative and quantitative similarities with PCA, NMF, and the Grassmann average [12]. These relations suggest that SCA be considered as a spatially-invariant extension to CA that adheres to pixel-level assumptions about image formation without explicitly requiring a parametric model of image transformation.

The effectiveness of SCA is validated through synthetic data and on the MSRC2 and Sift Flow datasets, demonstrating the qualitatively-meaningful unsupervised clustering of image regions and competitive results in weakly-supervised semantic segmentation.

2. Related Work

Component Analysis (CA) and Matrix Factorization: CA methods play a key role in many computer vision applications due to its ability for linear and non-linear dimensionality reduction, denoising, feature extraction and exploratory data analysis. See [6] for a review of CA methods. Though successful, early CA applications such as Eigenfaces [38] were unable to produce interprettable components. This was partially resolved through NMF, which demonstrated the ability to decompose images into more natural components corresponding to localized parts [21]. Numerous extensions have since been proposed to improve interpretability through localization constraints [23] or sparsity-inducing regularization [15]. Other approaches have explicitly modeled the physical process of occlusion by introducing additional latent variables that encode the ordering of objects in the scene [13]. However, all of these methods still require that all objects in different images be aligned, which is impractical for real images.

Transformation-Invariant Representations: Other methods have attempted to explicitly address this need for representations that are invariant to uninformative image variations. This is usually accomplished by simultaneously aligning and decomposing the images in an alternating manner. For example, [8] introduced discrete latent variables that select from predefined linear image transformations. Similarly, [18] learned translation-invariant appearance and occlusion models for videos. To be able to scale to higher parametric models, [7] proposed parameterized CA. However, these types of methods are typically restricted to small parametric classes of image transformations (e.g. translation or rotation) and cannot account for multiple objects or strong changes in pose.

Object Localization and Segmentation: Identifying and localizing the semantic classes within an image is an example of a task for which invariances cannot be easily parametrized. In addition to accounting for non-rigid transformations, large intra-class appearance variations must also be considered. Thus, none of the CA techniques described above would be able to give a semantically-meaningful separation into classes.

Instead, most approaches to this problem incorporate prior knowledge about class appearance and image composition to guide image segmentations or bounding box localizations. If fully-supervised training data is available, the most effective method is to train discriminative models that can be used to directly classify individual image regions. These local predictions are typically guided towards global consistency using prior knowledge such as local similarity [4, 9, 19], contextual geometric constraints [37], or agreement between multiple independent segmentations [1, 16]. More recently, convolutional neural networks (CNNs) have been applied to region classification with great success [10, 27].

Without pixel-wise labeling of training images, simple discriminative models are no longer viable. Some weakly-supervised approaches attempt to simultaneously learn discriminative classifiers alongside object locations through alternating methods like multiple-instance learning [14, 5] or matrix completion [3]. Others use graphical models that enforce consistency both within and across images to ensure class similarity [39, 41, 42]. However, exact inference in these models is typically intractable, so approximate methods must be used instead. Furthermore, all of these methods require large, non-convex optimization problems that are sensitive to initialization and do not scale well to large data sets. Leveraging the recent work in the optimization of deep networks, approaches based on CNNs have resulted in high-quality segmentations even without full supervision [33, 30, 34, 31, 32]. However, none of these approaches can be used like SCA for the unsupervised clustering of images into semantically-meaningful regions.

3. Semantic Component Analysis

Data decomposition techniques that rely on matrix factorization approximate a matrix $\mathbf{X}$ (with data instances $\boldsymbol{x}_i$ as its columns) as the product of two lower-rank matrices $\mathbf{W}$ and $\mathbf{B}$, i.e. $\mathbf{X} \approx \mathbf{B}\mathbf{W}^{\intercal}$ (see notation [1]). In other words, data points are represented as linear combinations of a shared set of basis components, i.e. $\boldsymbol{x}_i \approx \mathbf{B}\boldsymbol{w}_i = \sum_j w_{ij}\boldsymbol{b}_j$ where $\boldsymbol{b}_j$ are the columns of $\mathbf{B}$ and $\boldsymbol{w}_i$ are the columns of $\mathbf{W}$. While modifications can be made depending on the application of interest through constraints on the factors (e.g. NMF), different loss functions for measuring reconstruction error (e.g. robust PCA), or regularization terms (e.g. sparse coding), matrix factorization approaches are limited in their ability to incorporate more complicated priors. It is also unclear how they could be effectively applied to structured tasks like image segmentation in which semantic regions are known to be spatially localized in distinct, non-overlapping regions. SCA addresses these issues by allowing for rich, instance-level constraints that can depend on image content.

3.1. Semantic Constraints for Segmentation

Ideally, we seek a semantically-interpretable technique for CA that represents each class as a single component. In order to encourage that this be the case in the absence of pixel-level annotations, we must rely on priors and constraints that summarize assumptions about how classes are represented in images. Specifically, we note that images tend to be separated into spatially-consistent partitions of object classes. However, because of intra-class variability and differing spatial layouts across images, these constraints would be inconsistent and impossible to enforce in traditional matrix factorization approaches.

Instead, we propose an exact data decomposition of each image feature $\boldsymbol{x}_i$ into it's own distinct set of instance components $\mathbf{H}_i$ (with columns $\boldsymbol{h}_{ij}$) in lieu of a shared basis:

$$\boldsymbol{x}_i = \mathbf{H}_i\boldsymbol{w}_i = \sum_{j=1}^{m} w_{ij}\boldsymbol{h}_{ij} \quad \forall i = 1, \ldots, n. \qquad (1)$$

Here, n represents the size of the dataset and m represents the total number of semantically-related groups of components (i.e. object classes) that we consider. Observe that having a separate set of components for each image–where the basis $\mathbf{H}_i$ depends on the image index i–differs from traditional CA methods which use a global basis common to every image. While learning $m \times n$ components from only n training examples may seem intractable, we show in Section 3.2 how this can be accomplished by enforcing similarity between instance components with same index j, i.e those belonging to the same object class. Exposing these latent components allows for easily incorporating instance-level semantic constraints related to *a priori* knowledge about individual data points, such as the layout and composition of objects within images. Specific examples of these constraints are given in Section 4 for the application of semantic segmentation, which allow $\boldsymbol{h}_{ij}$ to contain the image features corresponding to the pixels in the i^{th} image assigned to the j^{th} class.

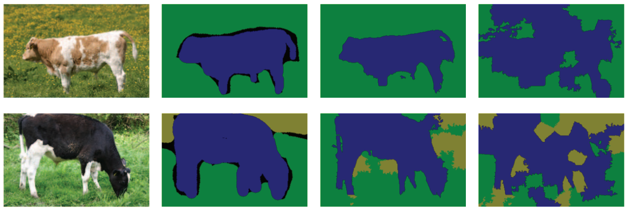
This formulation assumes *additive* image representations, meaning that an image's global feature vector can be expressed as the sum of its segment feature vectors. Note that many shallow representations share this property, including all average-pooled local features. However, the recently popularized deep features (extracted from intermediate activations in a convolutional neural network [17]) do *not* have this property due to the complicated nonlinear interactions within the network. In this paper, we use as image representations simple ℓ_1-normalized Bag-of-Words histograms over dense SIFT descriptors [28] quantized to $d = 1024$ dictionary elements.

The rest of this section describes the intuitive instance constraints that we enforce in order to encourage the semantic interpretability of components.

3.1.1 Superpixel Oversegmentation

To introduce local consistency and reduce computational requirements, we begin with an over-segmentation of each image into p_i locally-consistent superpixel feature vectors of dimensionality d. Let $\mathbf{S}_i \in \mathbb{R}^{d \times p_i}$ be a matrix with the i^{th} image's normalized superpixel features $\boldsymbol{s}_{ik}$ as its columns. Let q_{ik} represent the proportion of the image taken up by the

[1] Bold capital letters $\mathbf{X}$ denote a matrix; $\mathbf{X_i}$ represents the i^{th} column of the matrix $\mathbf{X}$. Bold lower-case letters a column vector $\mathbf{x}$; x_j denotes the scalar in the j^{th} element of $\mathbf{x}$. All non-bold letters represent scalars.



(a) Image (b) Ground Truth (c) $\lambda = 0.05$ (d) $\lambda = 0$

Figure 2. A comparison of segmentation results both with (c) and without (d) spatial consistency regularization.

k^{th} superpixel and denote by $\boldsymbol{q}_i$ the vector with these values as its elements. Thus, due to its additivity, $\boldsymbol{x}_i = \mathbf{S}_i \boldsymbol{q}_i$. That is, the image histogram $\boldsymbol{x}_i$ is a convex combination of its superpixel histograms $\boldsymbol{s}_{ik}$.

To account for object class occlusion in the image, we enforce that the instance components $\boldsymbol{h}_{ij}$ come from non-overlapping partitions of superpixels by defining indicator variables $z_{ijk} \in \{0, 1\}$ that are 1 if the k^{th} superpixel belongs only to the j^{th} class and 0 otherwise. Let $\boldsymbol{z}_{ij}$ be the column vector formed by stacking the z_{ijk} for all k. Then, the weighted component histograms can be written as $w_{ij}\boldsymbol{h}_{ij} = \mathbf{S}_i \text{diag}(\boldsymbol{q}_i)\boldsymbol{z}_{ij}$, where w_{ij} represents the proportion of the i^{th} image belonging to the j^{th} class. This also constrains the component by $w_{ij} = \boldsymbol{q}_i^{\mathsf{T}} \boldsymbol{z}_{ij}$ so that $0 \leq w_{ij} \leq 1$ and $\sum_{j=1}^{m} w_{ij} = 1$.

3.1.2 Spatial Consistency via Spectral Clustering

While the over-segmentation of images into superpixels provides some local spatial consistency, many superpixels could still make up a single object. Thus, we incorporate an additional regularization term borrowed from the spectral clustering and co-segmentation literature [19] that promotes smoothness between superpixels. Specifically, we define a similarity matrix $\mathbf{W}_i$ that assigns each pair of superpixels in an image a weight determined by their spatial proximity and color similarity. Denote by $\mathbf{L}_i$ the normalized graph Laplacian constructed from $\mathbf{W}_i$. Enforcing that the quantity $\boldsymbol{z}_{ij}^{\mathsf{T}} \mathbf{L}_i \boldsymbol{z}_{ij}$ be small (less than a threshold parameter ρ) encourages nearby superpixels with similar color to take on the same label. Fig. 2 shows an example of this.

3.1.3 Constraint Relaxation

Note that this set of constraints is non-convex since we enforce z_{ijk} to be binary, which would make optimization difficult. Thus, we first relax this constraint by allowing z_{ijk} to take on values within the continuous interval $[0, 1]$. Since $\sum_{j=1}^{m} z_{ijk} = 1$, z_{ijk} can be interpreted as the degree to which the k^{th} superpixel in the i^{th} image belongs to the j^{th} class. The solution can then be rounded by selecting the class with the highest value in order to produce a discrete segmentation.

Combining this with the constraints in the previous section allows us to write our semantic instance constraints as follows in Eq. 2:

$$\mathcal{C}_i = \Big\{ w_{ij}, \boldsymbol{h}_{ij} \ : \ w_{ij}\boldsymbol{h}_{ij} = \mathbf{S}_i \text{diag}(\boldsymbol{q}_i)\boldsymbol{z}_{ij}, \ \boldsymbol{z}_{ij}^{\mathsf{T}} \mathbf{L}_i \boldsymbol{z}_{ij} \leq \rho, \\ w_{ij} = \boldsymbol{q}_i^{\mathsf{T}} \boldsymbol{z}_{ij}, \ \sum_{j=1}^{m} z_{ijk} = 1, \ 0 \leq z_{ijk} \leq 1 \Big\} \tag{2}$$

This constraint set is very general and can be easily adapted to include additional image priors or modified to be applicable to tasks even beyond image segmentation. Even so, these simple, intuitive constraints surprisingly still result in semantically-meaningful decompositions. Furthermore, because this set is convex, it allows for convenient optimization, as discussed later in Section 3.3.

3.2. Problem Formulation

While these instance-level constraints limit the segmentations possible within a single image, there is no information shared between images to aid in the consistent assignment of classes. However, we can assume that regions belonging to the same class should be more similar than those belonging to different classes, and so too should their corresponding instance components. We formalize this intuition with the optimization problem in Eq. 3, which constrains the global image feature vector $\boldsymbol{x}_i$ to equal a linear combination of its instance components $\boldsymbol{h}_{ij}$ while minimizing the sum of weighted distances to exemplar components $\boldsymbol{b}_j$ that are representative of the semantic classes.

$$\begin{aligned} &\underset{w_{ij}, \boldsymbol{h}_{ij}, \boldsymbol{b}_j}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij}^2 \left\| \boldsymbol{h}_{ij} - \boldsymbol{b}_j \right\|_2^2 \\ &\text{s.t.} \sum_{j=1}^{m} w_{ij}\boldsymbol{h}_{ij} = \boldsymbol{x}_i, \ \{w_{ij}, \boldsymbol{h}_{ij}\} \in \mathcal{C}_i \end{aligned} \tag{3}$$

This formulation attempts to regularize the solution for $\boldsymbol{h}_{ij}$ by shrinking them towards other instance components of the same class while adhering to the instance-level constraints in $\mathcal{C}_i$. Effectively, instead of minimizing the total reconstruction error of each image, we minimize the variation within classes. Inference in this paradigm for CA amounts to finding instance components that adhere to image constraints and exactly reconstruct the data while being as close as possible to shared exemplar components, in contrast to traditional matrix factorization approaches which simply project the data onto a shared basis.

3.3. Optimization

Because Eq. 3 is not jointly convex, we take an alternating minimization approach for finding its solution that is similar in spirit to Lloyd's algorithm for k-means [26].

After a random initialization, we alternate between solving first for the exemplar components and then for the individual instance components and coefficients. This training procedure is easily implementable with off-the-shelf solvers and often requires very few iterations to converge.

With the coefficients w_{ij} and components $\boldsymbol{h}_{ij}$ fixed, the exemplar components $\boldsymbol{b}_j$ are given simply as the weighted averages of all of the individual components that share the same class (with weights given by w_{ij}^2).

$$\arg\min_{\boldsymbol{b}_j} \sum_{i=1}^{n} w_{ij}^2 \left\|\boldsymbol{h}_{ij} - \boldsymbol{b}_j\right\|_2^2 = \frac{\sum_{i=1}^{n} w_{ij}^2 \boldsymbol{h}_{ij}}{\sum_{i=1}^{n} w_{ij}^2} \tag{4}$$

Then, we fix the exemplar components $\boldsymbol{b}_j$ allowing for the separation of our problem into n smaller, independent subproblems–one per training image. This is accomplished by considering augmented variables $\tilde{\boldsymbol{h}}_{ij}$ formed by concatenating the unnormalized components $w_{ij}\boldsymbol{h}_{ij}$ with their corresponding coefficients w_{ij}, as shown in Eq. 5.

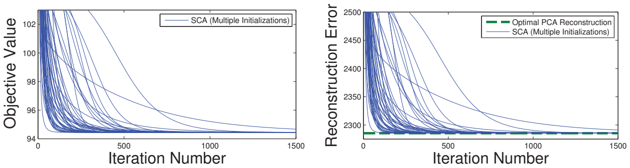
$$\begin{aligned} &\arg\min_{w_{ij},\tilde{\boldsymbol{h}}_{ij}} \sum_{j=1}^{m} \tilde{\boldsymbol{h}}_{ij}^{\mathsf{T}} \begin{bmatrix} \mathbf{I}_d & -\boldsymbol{b}_j \\ -\boldsymbol{b}_j^{\mathsf{T}} & \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{b}_j \end{bmatrix} \tilde{\boldsymbol{h}}_{ij} \quad \text{s.t. } \tilde{\boldsymbol{h}}_{ij} = \begin{bmatrix} w_{ij}\boldsymbol{h}_{ij} \\ w_{ij} \end{bmatrix} \\ &\sum_{j=1}^{m} \begin{bmatrix} \mathbf{I}_d & \mathbf{0} \end{bmatrix} \tilde{\boldsymbol{h}}_{ij} = \boldsymbol{x}_i,\ \{w_{ij}, \boldsymbol{h}_{ij}\} \in \mathcal{C}_i \end{aligned} \tag{5}$$

These subproblems are sparse, strictly convex quadratic programs in both w_{ij} and $\boldsymbol{h}_{ij}$ as long as the constraint set $\mathcal{C}_i$ is convex, resulting in a unique solution. (Note that while the inner matrices are not positive definite in general, they are when restricted to the affine subspace defined by the linear equality constraint.)

This optimization procedure is similar to the alternating projection algorithm for finding the approximate intersection between two sets. Specifically, we have two competing goals. On one hand, we want all instance components $\boldsymbol{h}_{ij}$ to be (approximately) equal. Projection onto this set yields the exemplar components $\boldsymbol{b}_j$ given simply as the weighted average in Eq. 4. But on the other hand, all instance components need to exactly reconstruct their corresponding data points and satisfy any additional constraints. This projection is accomplished through Eq. 5. This type of algorithm is known to converge to a solution when the two sets are convex [11] or more generally when they are smooth manifolds that intersect transversally [22]. In our case, the exact geometric characterization of this problem is not obvious. However, in Figures 3 and 4 we demonstrate empirically that our algorithm converges quickly and is robust to initialization in both unsupervised and weakly-supervised environments.

3.4. Introducing Supervision

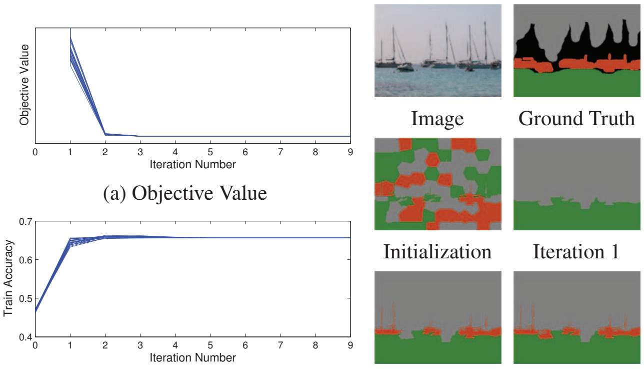
While the formulation described thus far does not require any training labels, various levels of supervision can be easily incorporated by simply fixing certain known elements during training. In particular, weak supervision can be included by forcing the coefficients w_{ij} for all absent classes to be zero, which effectively requires summing only over those classes present in an image.



(a) Objective Value

(b) Reconstruction Error

Figure 3. Our algorithm's convergence without constraints on synthetic data and 50 random initializations. Despite its alternating nature, our approach is robust to initialization and typically converges very quickly in both objective value (a) and reconstruction error from projection onto the exemplar components (b).



(a) Objective Value

(b) Train Pixel Accuracy

Image

Ground Truth

Initialization

Iteration 1

Iteration 2

Iteration 3

Figure 4. Left: The convergence of our algorithm with the constraints in Eq. 2 on the MSRC2 data set with weak labels. Showing 20 random initializations, both the objective value (a) and the training accuracy (b) consistently converge to the same values after only around 3 iterations. Right: Example segmentations at different points in the training process. After iteration 1, the large water and sky regions are successfully found, while iterations 2 and 3 segment the smaller boats.

In the case of full supervision, the true class assignments for each superpixel are known, so training reduces simply to taking weighted averages over instance components belonging to the same class, as shown in Eq. 4.

3.5. Relation to Matrix Factorization

While SCA learns a separate set of instance components for each image, the exemplar components $\boldsymbol{b}_j$ can be represented simply as the weighted average of all instance components $\boldsymbol{h}_{ij}$ sharing the same index j. Importantly, unlike other methods employing high-dimensional and overcomplete bases, the many instance components of SCA are *not* estimated independently; they are related through the smaller set of exemplar components, which can be interpreted as a shared basis representative of the training data.

Despite their seemingly unfamiliar construction, we empirically found that the exemplar components of SCA share close connections between the bases learned through traditional matrix factorization techniques. For comparison purposes, we use the shared exemplar components as a basis that can approximately reconstruct data in the same manner as PCA or NMF. Without any additional semantic constraints $\mathcal{C}_i$, this basis consistently achieves reconstruction performance comparable to that of PCA despite the different objective function. This is shown empirically in Fig. 3. In addition, by introducing nonnegativity constraints on both w_{ij} and $\boldsymbol{h}_{ij}$, the resulting exemplar components are qualitatively similar to the basis vectors found through NMF. This is shown in Fig. 5, which gives a visual comparison between our method and both PCA and NMF.

The key contribution of SCA is that, unlike matrix factorization approaches, explicitly decomposing a shared basis into separate instance components allows for richer constraints that would otherwise not be possible.

3.6. Incorporating Robustness

The Grassmann Average [12] (GA) is a recent method for scalable dimensionality reduction that represents data points as one-dimensional subspaces and constructs a leading component as their spherical average. This is very similar to our method which also represents components as weighted averages.

Specifically, GA can be considered a special case of our problem for a single component ($m = 1$) with the additional constraints $w_i = \pm 1$ and $\|\boldsymbol{b}\|_2 = 1$. After incorporating these constraints, Eq. 3 can be written as:

$$\operatorname*{arg\,max}_{w_i,\boldsymbol{b}} \sum_{i=1}^{n} w_i \boldsymbol{x}_i^\intercal \boldsymbol{b} \quad \text{s.t. } w_i = \pm 1,\ \|\boldsymbol{b}\|_2 = 1 \qquad (6)$$

Note that $w_i = 1$ if and only if $\boldsymbol{x}_i^\intercal \boldsymbol{b}$ is positive. Thus, the objective can be equivalently represented by replacing the multiplication of $\boldsymbol{x}_i^\intercal \boldsymbol{b}$ by w_i with an absolute value, resulting in exactly the same problem solved by GA.

One of the main benefits of GA is that robustness can be easily incorporated simply by using the robust feature-wise trimmed average (in which the smallest and largest $P\%$ of values are ignored) in place of the ordinary average, which is highly sensitive to outliers. We apply this same idea to introduce robustness to our algorithm as well, which was found to be particularly effective in cases when supervision is minimal or altogether unavailable. However, while GA must rely on greedy methods for acquiring more than just the leading component (which could affect what is considered to be an outlier), our algorithm is able to estimate multiple components simultaneously.

4. Application to Semantic Segmentation

In this section, we describe how our method can be trivially applied to the task of semantic segmentation. Our intention is not to compete with highly-engineered, state-of-the-art systems, but instead to demonstrate that introducing intuitive instance-level constraints to traditional CA techniques results in interpretable semantic components. Furthermore, our method easily accommodates any level of supervision, allowing for both unsupervised clustering of image regions and weakly-supervised semantic segmentation.

The full semantic segmentation objective can be written by combining Eq. 3 with the semantic constraints in Eq. 2 as follows. The segment belonging to the j^{th} class is then represented as the collection of superpixels whose indices correspond to the non-zero elements in $\boldsymbol{z}_{ij}$.

$$\operatorname*{arg\,min}_{w_{ij},\boldsymbol{z}_{ij},\boldsymbol{b}_j} \sum_{i=1}^{n}\sum_{j=1}^{m} \left\{ \|\mathbf{S}_i \operatorname{diag}(\boldsymbol{q}_i)\boldsymbol{z}_{ij} - w_{ij}\boldsymbol{b}_j\|_2^2 + \lambda \boldsymbol{z}_{ij}^\intercal \mathbf{L}_i \boldsymbol{z}_{ij} \right\}$$

$$\text{s.t. } \boldsymbol{q}_i^\intercal \boldsymbol{z}_{ij} = w_{ij},\ \sum_{j=1}^{m} z_{ijk} = 1,\ 0 \le z_{ijk} \le 1 \qquad (7)$$

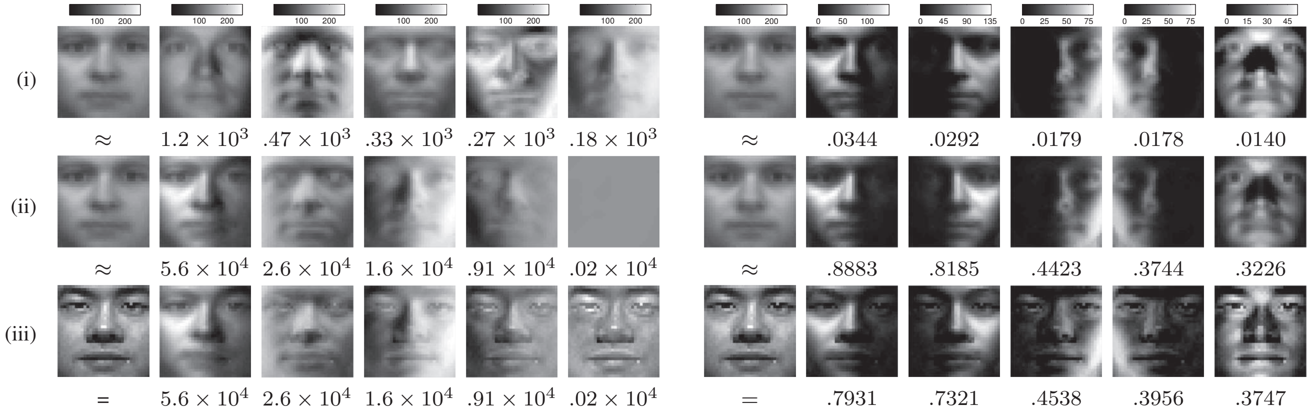
Note that we replace the quadratic inequality constraint from Section 3.1.2 with an equivalent penalty term by introducing a trade-off parameter λ that controls the level of smoothing. Its value is chosen through cross validation.

5. Experiments

To demonstrate the effectiveness of our method, we evaluate it against a number of datasets with varying levels of superivsion.

First, we consider synthetic data with minimal controlled intra-class variation. Specifically, we use 500 training images and 200 testing images generated by first selecting one of three backgrounds from the Salzburg Texture Image Database [20] and then randomly placing up to 7 rescaled objects segmented from the MSRC2 dataset [35], for a total of 10 classes. There is a maximum of 50% overlap with other objects and the image edges (simulating occlusion), and there are 2.9 classes per image on average.

Table 1 shows the segmentation performance of our algorithm with varying levels of supervision using a BoW dictionary size of 1024 with smoothness regularization parameter $\lambda = 0.05$ and using a robust trimmed average with $P = 20\%$. In the unsupervised setting, clusters were permuted and assigned to class labels in order to maximize average training accuracy. As unsupervised baselines, we also compare k-medians clustering of both ground-truth segments and independent superpixels. Even though our method is based on superpixels, its performance is very close to the clustering of ground-truth segments, even performing better on smaller classes. This is likely due to the


(i)
(ii)
(iii)
≈ 1.2×10^3 $.47 \times 10^3$ $.33 \times 10^3$ $.27 \times 10^3$ $.18 \times 10^3$
≈ 5.6×10^4 2.6×10^4 1.6×10^4 $.91 \times 10^4$ $.02 \times 10^4$
= 5.6×10^4 2.6×10^4 1.6×10^4 $.91 \times 10^4$ $.02 \times 10^4$
≈ .0344 .0292 .0179 .0178 .0140
≈ .8883 .8185 .4423 .3744 .3226
= .7931 .7321 .4538 .3956 .3747
100 200
0 50 100
0 45 90 135
0 25 50 75
0 15 30 45

(a) Unconstrained (PCA)

(b) Non-Negativity Constraints (NMF)

Figure 5. A comparison of the components found through SCA with (a) PCA and (b) NMF. The first column in each row shows a reconstructed image while the next five columns show the components used and the corresponding coefficients that minimize its reconstruction error. Row (i) uses the basis found through matrix factorization (either PCA or NMF), (ii) the shared exemplar components $\boldsymbol{b}_j$ of SCA, and (iii) the instance components $\boldsymbol{h}_{ij}$ of SCA that exactly reconstruct the image. The qualitative similarity between these components and the comparable reconstruction performance suggests a close relationship between SCA and traditional matrix factorization, despite their different objective functions. (Note that each column is normalized to the same scale for visualization.)

Table 1. Accuracies with different levels of supervision.

	aeroplane	cow	building	car	sheep	tree	grass	marble	stone	bark	total
	1	2	3	4	5	6	7	8	9	10	
Cluster (GT)	02	66	41	52	00	96	98	96	49	96	77
Cluster (Super)	00	29	96	32	03	39	52	29	31	28	36
SCA (None)	76	73	84	65	84	01	86	88	81	75	77
SCA (Weak)	80	81	90	74	86	53	81	83	82	88	82
SCA (Full)	77	78	85	74	78	55	86	88	88	92	85

Table 2. MSRC2 total segmentation accuracies.

	[39]	[2]	[25]	**SCA (None)**	**SCA (Weak)**	**SCA (Full)**
Total Acc.	67	69	71	60	70	77

Table 3. Sift Flow average segmentation accuracies.

	[39]	[40]	[41]	**SCA (None)**	**SCA (Weak)**	**SCA (Full)**
Avg. Acc.	14	21	28	14	19	25

joint assignment of all classes within an image according to the image formation constraints in $\mathcal{C}_i$. Simply clustering superpixels results in very poor performance because small regions do not contain enough class-specific features.

While increasing the level of supervision improved accuracy somewhat (especially for "tree", which is visually similar to the background classes such as "grass"), our algorithm was generally able to cluster the image regions into the correct semantic classes even with minimal training. Class confusion matrices and example segmentations are shown in Fig. 6.

We also evaluated our algorithm on the MSRC2 dataset [35], which contains 591 images segmented into 21 ground truth classes. We first applied our method in the unsupervised setting with $m = 10$ latent classes. Smoothness regularization was used with $\lambda = 2$ along and exemplar components were computed using the median, i.e. with $P = 50\%$. Example qualitative results are shown in Fig. 7. Note that the resulting groups are semantically related and generally give a good separation between classes. For example, nearly all "aeroplane" pixels were assigned to cluster 2, which also included pixels associated with other man-made objects such as "car" and "boat".

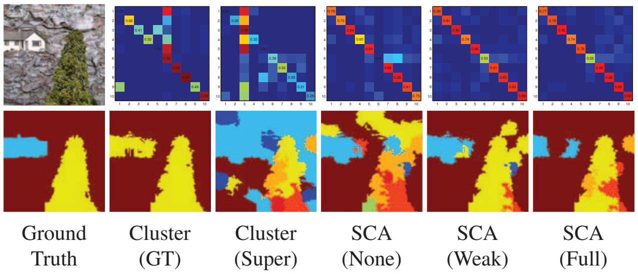


Figure 6. Top: Confusion matrices for the accuracies in Table 1. Bottom: Example segmentations for the different methods. Increasing levels of supervision improve segmentation consistency.

We tested our algorithm with weak labels provided at both training and testing time. To provide context, we also show results of our method when training without any labels and with full pixel-level annotations. We used the standard method for separating the training and testing data [35]. Table 2 summarizes our results in terms of total pixel accuracy in comparison to other methods. Despite the simplicity of our algorithm, we achieve comparable performance to many state-of-the-art systems specifically engineered for the task. Fig. 8 shows some example successful and unsuccessful segmentations.

Finally, we evaluated performance on the challenging

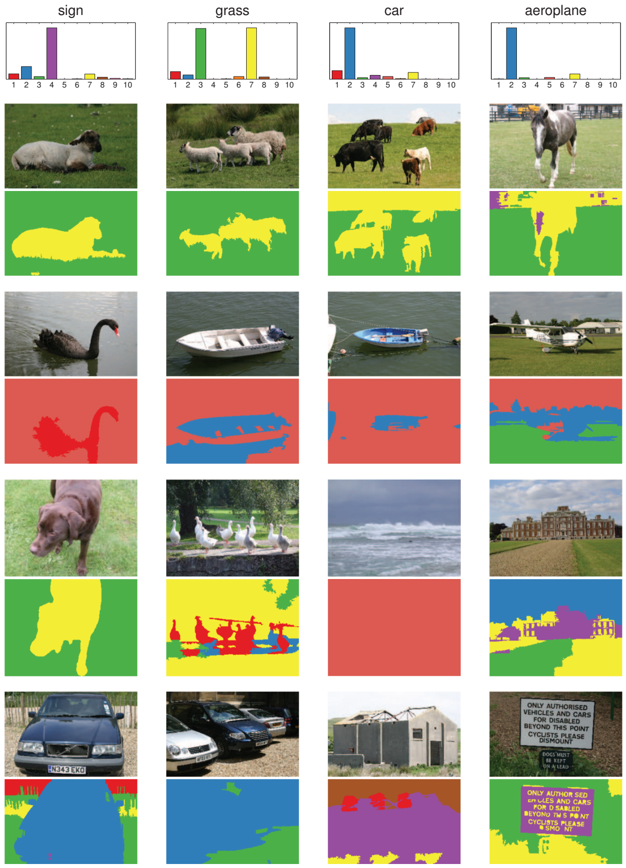


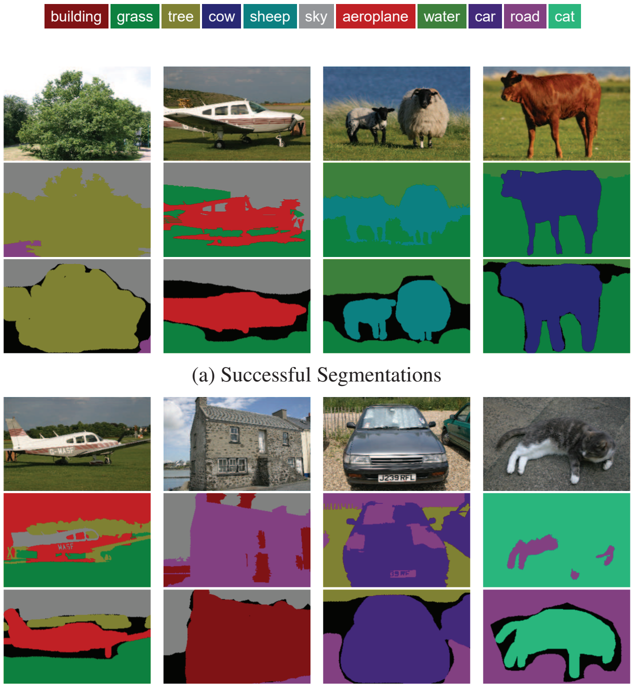
Figure 7. Example unsupervised segmentation results on the MSRC2 dataset. The bar plots on top show the proportion of pixels associated with a given ground truth class that were assigned to each of the 10 unsupervised clusters. Below are example images and the resulting segmentations achieved by our algorithm, showing clear separation into semantically-meaningful groups.

Sift Flow dataset [24], which contains 2688 total images (200 of which are used only for testing) and 33 classes, with an average of 4.43 classes per image. Following [41], we predict weak labels of testing images using linear SVMs trained on 4096-dimensional features extracted from the last fully-convolutional layer (fc6) in the pre-trained Caffe CNN [17]. Table 3 shows average class accuracy in comparison to other methods. Results from unsupervised and fully-supervised training are also shown for comparison. We again achieve comparable performance to other methods that are designed specifically for weakly-supervised semantic segmentation and use much richer feature sets (color, GIST, and superpixel locations) and priors (e.g. objectness, ILP, and discriminative appearance models.)

6. Conclusion

In this paper, we outlined a general framework for explicitly introducing interpretability to CA. This was accomplished through an alternative objective function (rather than the traditional least squares reconstruction error from matrix factorization) that exposes instance components which can be constrained using prior information. Specifically, we formalized an intuitive observation: images tend to be partitioned into spatially-consistent, non-overlapping regions that belong only to a single class. Despite their simplicity, these constraints allow for the semantically-meaningful clustering of image regions. Requiring only BoW features and superpixel color similarities, our algorithm is easily-implementable, efficient, and robust to initialization. Furthermore, varying levels of supervision can be incorporated trivially.

Even without manual engineering, fine-tuning, or overfitting to a particular dataset, we achieve competitive performance on standard weakly-supervised semantic segmentation tasks. Our approach is general, allowing for the simple inclusion of additional constraints and priors with the potential to improve these results even further. SCA could also be easily adapted to numerous other applications beyond semantic segmentation, including time series analysis, background modeling in videos, etc.



(a) Successful Segmentations

(b) Failure Cases

Figure 8. Example weakly-supervised segmentations from the MSRC2 dataset showing both (a) successful and (b) unsuccessful cases. Typical failure cases occur because of confusion between visually similar classes that commonly co-occur (e.g. sky and road) or when different classes have very similar color (e.g. the gray cat sitting on the road.)

Acknowledgments: This work was partially supported by the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program.

References

[1] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *Conference on Computer Vision and Pattern Recognition*, 2012. 2

[2] F. Briggs, X. Z. Fern, and R. Raich. Rank-loss support instance machines for miml instance annotation. In *Knowledge Discovery and Data Mining (KDD), ACM SIGKDD International Conference on*, pages 534–542, 2012. 7

[3] R. Cabral, F. De La Torre, J. Costeira, and A. Bernardino. Matrix completion for weakly-supervised multi-label image classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1–1, 2014. 3

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *European Conference on Computer Vision*, 2012. 2

[5] R. G. Cinbis, J. Verbeek, C. Schmid, et al. Multi-fold mil training for weakly supervised object localization. In *Conference on Computer Vision and Pattern Recognition*, 2014. 3

[6] F. De la Torre. A least-squares framework for component analysis. *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, 34(6):1041–1055, 2012. 2

[7] F. De la Torre and M. J. Black. Robust parameterized component analysis: Theory and applications to 2d facial appearance models. *Computer Vision and Image Understanding*, 91:53–71, 2003. 2

[8] B. J. Frey and N. Jojic. Transformed component analysis: Joint estimation of spatial transformations and image components. In *Conference on Computer Vision and Pattern Recognition*, 1999. 2

[9] A. Gandhi, K. Alahari, and C. Jawahar. Decomposing bag of words histograms. In *International Conference Computer Vision*, pages 305–312. IEEE, 2013. 2

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2014. 3

[11] L. Gubin, B. Polyak, and E. Raik. The method of projections for finding the common point of convex sets. *Computational Mathematics and Mathematical Physics*, 7(6), 1967. 5

[12] S. Hauberg, A. Feragen, and M. J. Black. Grassmann averages for scalable robust pca. In *Conference on Computer Vision and Pattern Recognition*, 2014. 2, 6

[13] M. Henniges, R. E. Turner, M. Sahani, J. Eggert, and J. Lücke. Efficient occlusive components analysis. *Journal of Machine Learning Research*, 15:2689–2722, 2014. 2

[14] M. Hoai, L. Torresani, F. De la Torre, and C. Rother. Learning discriminative localization from weakly labeled data. *Pattern Recognition*, 47(3):1523–1534, 2014. 3

[15] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004. 2

[16] A. Ion, J. Carreira, and C. Sminchisescu. Probabilistic joint image segmentation and labeling by figure-ground composition. *International Journal of Computer Vision*, 2014. 2

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 3, 8

[18] N. Jojic and B. J. Frey. Learning flexible sprites in video layers. In *Conference on Computer Vision and Pattern Recognition*, 2001. 2

[19] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2010. 2, 4

[20] R. Kwitt and P. Meerwald. Salzburg texture image database (STex). http://wavelab.at/sources/STex/. 6

[21] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 2

[22] A. S. Lewis and J. Malick. Alternating projections on manifolds. *Mathematics of Operations Research*, 33(1):216–234, 2008. 5

[23] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Conference on Computer Vision and Pattern Recognition*, 2001. 2

[24] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12), Dec 2011. 8

[25] Y. Liu, J. Liu, Z. Li, J. Tang, and H. Lu. Weakly-supervised dual clustering for image semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2013. 7

[26] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982. 4

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015. 3

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[29] M. H. Nguyen, L. Torresani, F. De la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *International Conference on Computer Vision (ICCV)*, 2009. 1

[30] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? weakly-supervised learning with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2015. 3

[31] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015. 3

[32] D. Pathak, P. Krähenbühl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. *arXiv preprint arXiv:1506.03648*, 2015. 3

[33] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, 2015. 3

[34] O. Russakovsky, A. L. Bearman, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. *ArXiv preprint arXiv:1506.02106*, 2015. 3

[35] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006. 6, 7

[36] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *International Conference Computer Vision*, 2005. 2

[37] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *European Conference on Computer Vision*, 2010. 2

[38] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Conference on Computer Vision and Pattern Recognition*, 1991. 2

[39] A. Vezhnevets, V. Ferrari, and J. Buhmann. Weakly supervised semantic segmentation with a multi-image model. In *International Conference Computer Vision*, 2011. 3, 7

[40] A. Vezhnevets, V. Ferrari, and J. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2012. 7

[41] J. Xu, A. G. Schwing, and R. Urtasun. Tell me what you see and i will show you where it is. In *Conference on Computer Vision and Pattern Recognition*, 2014. 3, 7, 8

[42] W. Zhang, S. Zeng, D. Wang, and X. Xue. Weakly supervised semantic segmentation for social images. In *Conference on Computer Vision and Pattern Recognition*, 2015. 3