Image analysis by counting on a grid

Alessandro Perina
Microsoft Research, Redmond

Nebojsa Jojic
jojic@microsoft.com
Microsoft Research, Redmond

Abstract

In recent object/scene recognition research images or large image regions are often represented as disorganized "bags" of image features. This representation allows direct application of models of word counts in text. However, the image feature counts are likely to be constrained in different ways than word counts in text. As a camera pans upwards from a building entrance over its first few floors and then above the penthouse to the backdrop formed by the mountains, and then further up into the sky, some feature counts in the image drop while others rise – only to drop again giving way to features found more often at higher elevations (Fig. 1). The space of all possible feature count combinations is constrained by the properties of the larger scene as well as the size and the location of the window into it. Accordingly, our model is based on a grid of feature counts, considerably larger than any of the modeled images, and considerably smaller than the real estate needed to tile the images next to each other tightly. Each modeled image is assumed to have a representative window in the grid in which the sum of feature counts mimics the distribution in the image. We provide learning procedures that jointly map all images in the training set to the **counting grid** *and estimate the appropriate local counts in it. Experimentally, we demonstrate that the resulting representation captures the space of feature count combinations more accurately than the traditional models, such as latent Dirichlet allocation, even when modeling images of different scenes from the same category.*

1. Introduction

A popular way to deal with diversity of imaging conditions and geometric variation in objects or entire scenes is to simply represent images or image regions as disordered "bags" of image features [5, 10]. Ideally, these features should be highly discriminative so that most categories of images of interest are uniquely identifiable by the presence of a handful of features. In practice, however, individual features are not sufficiently discriminative, and modeling joint variation in feature counts becomes an interesting machine learning problem.

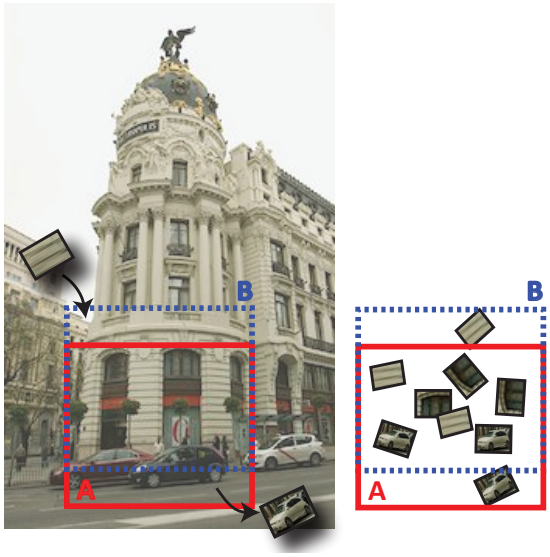


Figure 1. Feature counts change slightly as the field of view moves from region A to region B of the scene. For example, the abundance of the car features is reduced, but the counts of the features found on building facades are increased. The counting grid model accounts for such changes naturally, and it can also account for images of different scenes.

It is tempting to use here the existing discrete models, such as histograms [10], multinomial mixtures [11] or latent Dirichlet allocation (LDA) [5, 15], already extensively validated on text data. However, the bags of features extracted from natural images have an imprint of the images' spatial structure, which is evident when the bags from related images are considered *together*, and ignoring the constraints imposed on the feature counts may have negative consequences in classification tasks.

For an illustration, Fig. 2 provides a synthetic example of several images *i)* of a train station, taken as windows into the larger scene *ii)*. Just for illustrative purposes, we hand-labeled the scene with feature labels as shown in *iii)*. Assuming that a few images are taken at random from the scene, we wonder if the feature counts in these images are sufficient to predict the possible feature counts in other im-

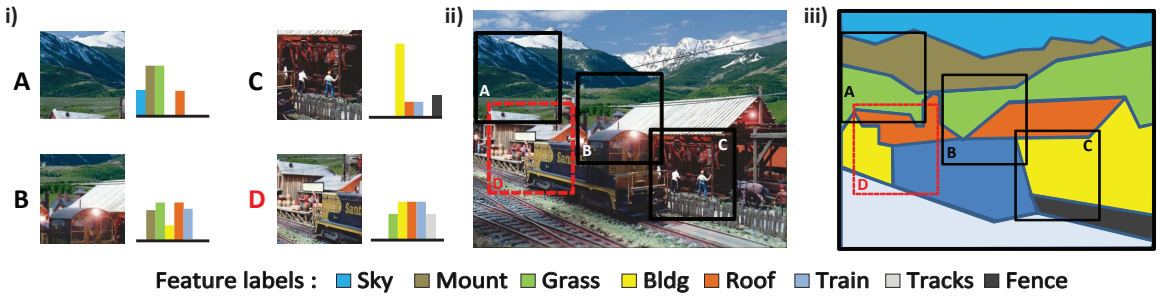


Figure 2. Counting grid illustration.

ages of the scene. In particular, we consider images taken from the regions close to A,B, and C in *iii)* and ask the question if the image D would fit the so defined train station class. The literature uses two sets of approaches to this problem. In one set of approaches, kernel or nearest-neighbor techniques start with the comparisons of the feature counts in the test image and each of the previously studied exemplars [1, 8, 12, 13, 19]. Although this comparison can be done in many different ways, we note here that these approaches would be complicated by the fact that none of images A,B, C have the combination of all five features that are present in D. The other approach is to consider all bags of features together and generalize [5, 7, 17]. A simplest approach to this would be to simply merge the bags. In this case, there is a danger of overgeneralization. For this particular example, there is a need for interpolating between the feature count vectors for A,B,C and other images. However, this interpolation is best performed by spatial reasoning. Given that in some training images we see, from the top to bottom, roof, train, tracks, and in others mountain, grass, roof, train, we can infer that the existence of grass, roof, train, tracks combination is likelier than the existence of the mountain, roof, train, tracks combination of features. Furthermore, the proportions of different features in the images carry the information about the thickness of the layers of these features, which should be useful for inferring which previously unseen feature count combinations can be found elsewhere in the scene. Surprisingly, not much of the spatial organization of the features in the training images needs to be retained in order to perform the spatial reasoning about which feature combinations are likely. In Fig.3-*i)* we show the counting grid inferred by iterating Eqs. 7 and 8 on the label counts from 50 windows into the scene taken at random, but avoiding all windows that contain all five of the features in D in any proportion. Each training image was represented as a set of 2×2 feature bags (upper left, lower left, upper right, lower right, see *ii)*), and without using the original location information, the counting grid was computed so that for each training image, a window into the counting grid can be found so that the appropriate sections have matching histograms. The resolution of the reconstructed feature layout of the large scene goes well beyond what would be expected from a crude 2×2 tessellation of the input images (the height of each section is roughly 20% of the large scene and only the feature counts in each section were used, not their spatial layout). Although none of the training examples was taken from the area close to D where all five of D's features can be seen in a single image, that part of the scene is reconstructed as well, and D's histogram can be matched well.

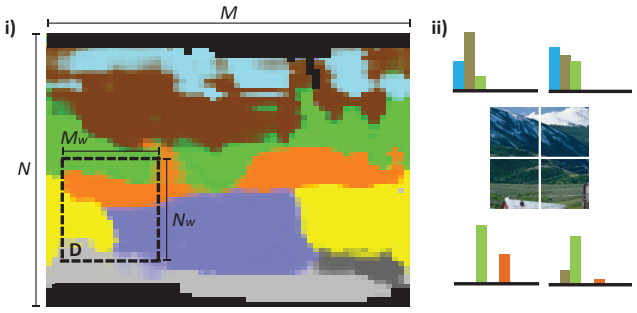


Figure 3. Inferred counting grid (*i)*) and a sample of the input used to learn it (*ii)*).

In this simple example, the training images are different views of a single scene. However, at the feature level, images of other train stations are likely have a similar layout, and so they could be used to learn a counting grid. In practice, we rarely have access to highly discriminative and reliable features, and so instead of the 8 fake features in our example, in our experiments we had to use hundreds of simpler (real) features, and infer the counting grids from related images of different scenes. We found that our representation captured the space of possible feature count combinations for various image categories significantly better than other generalization techniques, and that our simple generative model, which can be used for unsupervised learning and clustering, too, often rivals the state of the art based on discriminative techniques that require supervision.

2. Imprint of spatial organization in disordered bags of words

As discussed above, we would like to understand the hidden constraints that govern the often-practiced simplifica-

tion of images into bags of features. This simplification has two stages. First, image features $z_{i,j}$ are extracted on a grid inside the image. These features are discrete, $z \in [1..Z]$, and they point to a codebook of features obtained by clustering the multidimensional real-valued features calculated by local image processing, e.g., SIFT [14]. (In some of our experiments, we also simply use quantized image colors as discrete features). Next, the feature counts are computed $c_z = \sum_{i,j} \mathbf{I}[z_{i,j} = z]$, where $\mathbf{I}[\cdot]$ is the indicator function. Only the counts c_z are then retained, and the spatial distribution $z_{i,j}$ is typically forgotten, with the justification that establishing correspondence for individual image locations across different images of the same thing would be prohibitively expensive, and that in practice only the presence or absence of features is informative, not their spatial distribution. However, if we consider a set of such bags of words from related images we can see that the feature counts in these disordered bags of features may still indirectly follow the rules of spatial organization. For example, if the bags $\{c_z^t\}$, indexed by t are extracted from several overlapping windows from a larger image, then the spatial structure of that image is imprinted in the particular count combinations in these bags. Furthermore, the spatial layout of the features in the large image may even be recoverable from these disordered bags! If the bags $\{c_z^t\}$ are created from *all* the overlapping windows from a large image, and if the source location for each bag is known, then we can easily see that under minimal additional assumptions regarding the boundaries in the image, we can reconstruct feature indices z at each location in the large image by solving the system of linear equations that arise from the count constraints. [1] In this way, we can reconstruct a large grid of features such that any of the count combinations we see in the given bags can be found in an appropriate window in this reconstruction. But this implies that the bags of features from the images of the same scene, when considered jointly, obey very strong constraints and thus taking these constraints into account will likely improve image analysis tasks that depend on the feature count representations. This insight leads to several interesting problems which we address in the next section.

1. *Joint estimation of the feature layout and the matching of the bags to windows into it*: If the bags of features (feature counts) from many – but *not all* – overlapping windows from a large scene are provided, and if the original locations of these windows are *withheld*, can we still reconstruct at least some of the original spatial arrangement of the features?

2. *Category modeling*: If the bags of features are not coming from the windows into a single scene, but instead from different but related images (e.g. of a particular image category or an object class), would these bags, when considered jointly, imply some spatial layout of the features, and would this layout help predict which combinations of feature counts are more likely in bags of features extracted from new images of the category in question?

3. *Using more of the original structure*: Given that in practice we typically have access to the original images, can more of their spatial structure be used in learning the spatial layout of features that would in turn constrain the bag of words representation in a useful way?

3. The counting grid model

The counting grid, $\pi_{i,j,z}$ is a set of normalized counts of features indexed by z on the grid $(i,j) \in [1..N] \times [1..M]$. Thus, $\sum_z \pi_{i,j,z} = 1$ everywhere on the grid. A given bag of image features, represented by counts $\{c_z\}$ is assumed to follow a count distribution found somewhere in the counting grid. In other words, the bag can be generated by first averaging all counts in the window $W_{k,\ell} = [k..k+N_W-1] \times [\ell..\ell+M_W-1]$, to form the histogram $h_{k,\ell,z} = \frac{1}{N_W M_W} \sum_{(i,j)\in W_{k,\ell}} \pi_{i,j,z}$, and then generating a set of $N_W \cdot M_W$ features in the bag. In other words, the position of the window k,ℓ in the grid is a 2D latent variable given which the probability of the bag of features $\{c_z\}$ is

$$p(\{c_z\}|k,\ell) = \prod_z (h_{k,\ell,z})^{c_z} = \alpha \prod_z \Big(\sum_{(i,j)\in W_{k,\ell}} \pi_{i,j,z} \Big)^{c_z}$$

where the constant $\alpha = (\frac{1}{N_W M_W})^{N_W M_W}$. Assuming, for simplicity, the uniform prior over positions k,ℓ in the grid, the joint distribution over all bags of features $\{c_z^t\}$, indexed by t and their corresponding latent window positions k^t, ℓ^t in the counting grid is

$$p\Big(\big\{\{c_z^t\}_{z=1}^Z, k^t, \ell^t\big\}_{t=1}^T\Big) \propto \prod_t \prod_z \Big(\sum_{(i,j)\in W_{k^t,\ell^t}} \pi_{i,j,z} \Big)^{c_z^t},$$

3.1. Inference and learning

To compute the log likelihood of the data, $\log P$, we need to sum over the latent variables k,ℓ before computing the logarithm, which, as in mixture models, or as in epitomes [2], which are much more similar to the counting grids, makes it difficult to perform assignment of the latent variables (in our case positions in the counting grid) while also

[1] Consider two horizontally neighboring windows: The count differences are completely determined by the feature identities of the only two columns that the two do not share. To separate the effect of the two columns, we can consider another pair of overlapping images whose count differences depend on only one of those two columns. To further break each column apart, we can consider vertically neighboring windows, etc. As long as the image has a thick enough border with only a single feature present, we can propagate these constraints until any given location's feature is uniquely determined.

estimating the model parameters. This makes an iterative exact or a variational EM algorithm necessary [16]. Bounding (variationally), the non-constant part of $\log P$, we get

$$\log P \geq B = -\sum_t \sum_{k^t,\ell^t} q_{k^t,\ell^t} \log q_{k^t,\ell^t} + \quad (1)$$
$$+ \sum_t \sum_{k^t,\ell^t} q_{k^t,\ell^t} \sum_z c_z^t \log \sum_{(i,j)\in W_{k^t,\ell^t}} \pi_{i,j,z},$$

where q_{k^t,ℓ^t}, or in shorthand, $q^t_{k,\ell}$, is the variational distribution over the possible latent mappings of the t-th bag .For a given counting grid π, the bound is maximized when each distribution q^t is equal to the exact posterior distribution. This is a standard variational derivation of the exact E step, which leads to

$$q^t_{k,\ell} \propto \exp\left(\sum_z c_z^t \log h_{k,\ell,z}\right) \quad (2)$$

which simply establishes that the choice of k, ℓ should minimize the KL divergence between the counts in the bag and the counts $h_{k,\ell,z} = \sum \pi_{i,j,z}$ in the appropriate window $W_{k,\ell}$ in the counting grid. For each t, the above expression is normalized over all possible window choices k, ℓ. To optimize the bound B with respect to parameters we note first that it is the second term in Eq.1 that involves these parameters, and that it requires another summation before applying the logarithm. The summation is over the grid positions i, j within the window $W_{k,\ell}$, which we can again bound using a variational distribution and the Jensen's inequality:

$$\log \sum_{(i,j)\in W_{k^t,\ell^t}} \pi_{i,j,z} = \log \sum_{(i,j)\in W_{k^t,\ell^t}} r^t_{i,j,k,\ell,z} \frac{\pi_{i,j,z}}{r^t_{i,j,k,\ell,z}}$$
$$\geq \sum_{(i,j)\in W_{k^t,\ell^t}} r^t_{i,j,k,\ell,z} \log \frac{\pi_{i,j,z}}{r^t_{i,j,k,\ell,z}} \quad (3)$$

where $r^t_{i,j,k,\ell,z}$ is a distribution over locations i, j, i.e. r is positive and $\sum_{(i,j)\in W_{k,\ell}} r^t_{i,j,k,\ell,z} = 1$, and is indexed by k, ℓ as the normalization is done differently in each window, and is indexed by z as it can be different for different features, and indexed by t as the term is inside the summation over t, so a different distribution r could be needed for different bags $\{c^t_z\}$. This distribution could be thought of as information about what proportion of the c_z features of type z was contributed by each of the different sources $\pi_{i,j,z}$ in the window $W_{k,\ell}$. However, by performing constrained optimization (so that r adds up to one), we find that assuming a fixed set of parameters π, the distribution $r^t_{i,j,k,\ell,z}$ that maximizes the bound is independent of t, i.e., the same for each bag:

$$r^t_{i,j,k,\ell,z} = \frac{\pi_{i,j,z}}{\sum_{(i,j)\in W_{k,\ell}} \pi_{i,j,z}} = \frac{\pi_{i,j,z}}{N_W M_W h_{k,\ell,z}} \quad (4)$$

If we do consider distributions r as a feature mapping to the counting grid, then this result is again intuitive. If all we know is that a bag containing c_z features of type z is mapped to the grid section $W_{k,\ell}$, and have no additional information about what proportions of these c_z features were contributed from different incremental counts $\pi_{i,j,z}$, then the best guess is that these proportions follow the proportions among $\pi_{i,j,z}$ inside the window. If we assume now that r and q distributions are fixed, then combining Eqs. 2,3 and minimizing the resulting bound wrt parameters $\pi_{i,j,z}$ under the normalization constraint over features z, we obtain the update rule,

$$\hat{\pi}_{i,j,z} \propto \sum_t \sum_{(k,\ell)|(i,j)\in W_{k,\ell}} q^t_{k,\ell} c^t_z r^t_{i,j,k,\ell,z} \quad (5)$$

which by Eq. 4 reduces to

$$\hat{\pi}_{i,j,z} \propto \pi_{i,j,z} \sum_t c^t_z \sum_{(k,\ell)|(i,j)\in W_{k,\ell}} \frac{q^t_{k,\ell}}{h_{k,\ell,z}} \quad (6)$$

The steps in Eqs. 2 and 6 constitute the E and M step which can be iterated till convergence (within a desired precision). The first step aligns all bags of features to grid windows that (re)match the bags' histograms, and the second re-estimates the counting grid so that these same histogram matches are even better. Thus, starting with non-informative (but symmetry breaking) initialization, this iterative process will jointly estimate the counting grid and align all bags to it. To avoid severe local minima, it is important, however, to consider the *counting grid as a torus*, and consider all windowing operations accordingly, as was previously proposed for learning epitomes [2, 3, 4]. This prevents the problems with grid boundaries which otherwise could not be crossed when more space is needed to grow the layout of the features.

3.2. Alternative EM steps

The described algorithm works remarkably well given that its task is essentially to infer a (probabilistic) image not from many image patches as is the case for epitome models, but only from summary statistics for such patches (Fig. 4 and Suppl. Material). The task is formidable because no directionality is provided in the bag of features representation, and the iterative algorithm may start to lay out the features topologically correctly, but following inconsistent directions in different parts of the counting grid, leading to slow convergence and/or local minima. However, it is straightforward to update the model and its E and M rules to deal with image representations that consist not of one, but several (S) bags of words, each corresponding to a section of the image. For each feature image $z^t_{i,j}$, we define S bags of words, defined by counts in different sections $\{c^{t,s}_z\}$. When inferring the mapping of the set of section bags, the

window $W_{k,\ell}$ is tessellated into the sections $W^s_{k,\ell}$ the same way images are tessellated (a 2×2 tessellation into upper left, upper right, lower left and lower right, for example), and the histogram comparisons are done accordingly,

$$q^t_{k,\ell} \propto \exp\left(\sum_s \sum_z c^{t,s}_z \log h^s_{k,\ell,z}\right) \quad (7)$$

The M step using section bags is

$$\hat{\pi}_{i,j,z} \propto \pi_{i,j,z} \sum_t \sum_s c^{t,s}_z \sum_{(k,\ell)|(i,j)\in W^s_{k,\ell}} \frac{q^t_{k,\ell}}{h^s_{k,\ell,z}}. \quad (8)$$

Figure 4 shows that even just considering n representation consisting of four bags of features in image sections (upper left, upper right, lower left and lower right) provides enough symmetry breaking that good counting grids can be estimated. Another obvious alternative is to use the existing layout of features $z^t_{i,j}$ in each of the training images when updating the counting grid. In this case, the M step becomes equivalent to what the epitome models would prescribe for the case of discrete measurements:

$$\pi_{i,j,z} \propto \sum_t \sum_{(k,\ell)|(i,j)\in W_{k,\ell}} q^t_{k,\ell} \mathbf{I}[z^t_{i-k+1,j-\ell+1} = z] \quad (9)$$

where $\mathbf{I}[\cdot]$ denotes the indicator function.

3.3. Computational efficiency

Careful examination of the steps reveals that by the efficient use of cumulative sums, all versions of the E and M steps are linear in the size of the counting grid, except for the last version of the M step, the epitome M step, Eq. 9. This last version of the counting grid update utilizes the feature layout of the original images $z^t_{i,j}$, which requires the a convolution operation, which is of the still manageable $O(N \log N)$ complexity. Both E and M steps of the algorithm Eq. 7, 8 require computing $\sum_{(i,j)\in W_{k,\ell}} f_{i,j}$, which can be done by first computing, in linear time, the cumulative sum $F_{m,n} = \sum_{(i,j)\le(m,n)} f_{i,j}$, and then setting

$$\sum_{(i,j)\in W_{k,\ell}} f_{i,j} = F_{k+N_W,\ell+M_W} - F_{k-1,\ell+M_W} - F_{k+N_W,\ell-1} + F_{k-1,\ell-1} \quad (10)$$

(See also integral images technique in [20].) This procedure is used to compute all window histograms h in the counting grid, as well as in either of the M step versions Eqs. 6, 8, which only use the counts $c^{t,s}_z$, and not the original feature layout $z^t_{i,j}$. Efficiency of the computation over multiple section bags in Eqs. 7, 8 can be increased if the sections break the window uniformly along each direction. The section histograms $h^s_{k,\ell,z}$ are then shifted versions of each other.

4. Experiments

In scene/object classification tasks, the image features are typically clustered around hundreds of centers and image locations (i, j) are associated with pointers z to these discretized features. For example, in our classification experiments below, we use Z=200 features. The illustration in Fig. 2 does not provide enough insight into how well the counting grids can be inferred when such large sets of features are considered. Visualizing the feature identities on a grid is difficult, and so, in order to simply study the properties of the counting grid estimation procedures discussed above, we ran the first set of tests on fifty 16×16 color patches taken at random from a drawing (available in Matlab: load trees) subsampled to the resolution of 33×40 (Fig. 4 *i)*). The patches were first transformed into maps of pointers $z^t_{i,j}$, each pointing to one of Z=64 colors obtained by approximating the color map. Then, 1×1, 2×2 and 4×4 histograms were computed in the appropriate sections of the feature maps to obtain the section bags of words for the algorithm defined by Eqs. 7, 8. The algorithm was then run on each section bag representation separately[2], to obtain the counting grids in *ii)*, *iii)* and *iv)*. Finally, the plate *v)* shows the result of the final algorithm: the combination of the counting grid E step, i.e. mapping of the windows based only on the single bag of words, Eq. 2, and the epitome M step, Eq. 9, which uses the original layout of features $z_{i,j}$ in each patch when updating the counting grid. To visualize the different counting grids, each counting grid location (i, j) was assigned the color equal to the average of the Z=64 colors in color map, weighted by the normalized local feature counts $\pi_{i,j,z}$. The image in *ii)* is therefore an attempt at reconstructing the image in *i)* from fifty color histograms for which we did not provide any additional information about their source: Image *i)* was not provided to the algorithm, nor were the locations of the images from which the fifty histograms were extracted. Note also that the algorithm is not aware of any similarities among the 64 colors, as these are treated as discrete features. Remarkably, a lot of the spatial structure in feature distributions was reconstructed from these 50 histograms. The algorithm discovers that the dark, red and brown tones go together and that they are bordered by green. Elongated dark structures against the blue background are discovered, as is the coast/island boundary. In this sense, the counting grid provides a good model for interpolating among the original 50 histograms, as the histograms from the original image are also likely under the inferred counting grid. Using 2×2 bags as a representation of images is already sufficient to break some symmetry problems and reconstruct almost the entire scene. This improvement is also remarkable, as in this case, osten-

[2] In the 1×1 tessellation this is trivially equivalent to iterating Eqs. 2, 6

Figure 4. The source of 50 image patches taken from random locations *i)*, and counting grids estimated by various versions of the algorithm. Most remarkably *ii)* is the reconstruction obtained using *only* 50 histograms of image features, and for reconstruction in *iii)* we used only 50 sets of 4 histograms (from 2 X 2 sections of the input images). The colors were treated as unrelated 64 discrete features.

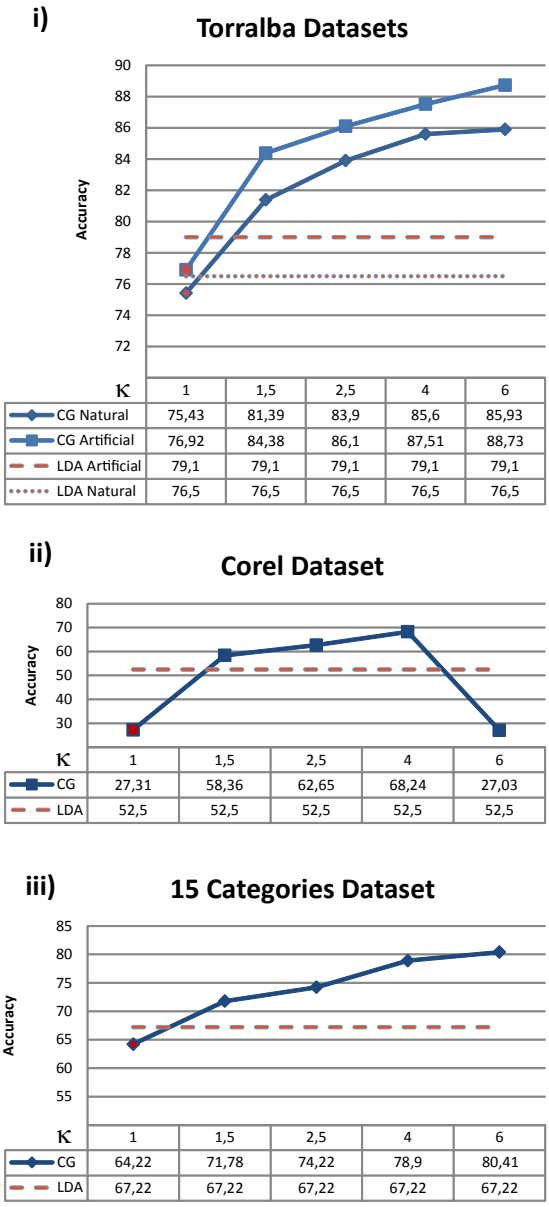
sibly very little information about the 50 image patches is used: The source image *i)*, or locations of the 50 patches in it are **not** available to the algorithm, and the algorithm only uses fifty sets of 4 histograms (upper left, upper right, lower left, lower right) over Z=64 colors found in appropriate sections to reconstruct the island and the trees. The most accurate reconstruction is obtained in *v)* by iterating Eqs. 2, 9), which is interesting from the epitome modeling point of view. If the counting grid is considered a feature epitome (as used at low resolutions in [3]), from which detailed feature maps $z^t_{i,j}$ are generated, rather than simply bags of features, then the inference step that only considers the patch histograms efficiently replaces the convolutional E step of the epitome model (if it were extended to have feature distribution in each image location, rather than real-valued Gaussian models). Furthermore, in this case we also found that this combination is less prone to local minima than the epitome models or the pure counting grid inference and learning of Eqs. 7, 8. Finally we note here that in the extreme case of tessellating the patches down to individual pixels, the counting grid becomes the feature epitome model.

4.1. Scene classification

We next show that these procedures can be used to analyze images that are related by the fact that they belong to the same category, rather than a larger single scene, and that the resulting generalization over the space of possible bag of feature count distributions far surpasses the standard count models including other latent models, such as LDA. In the following experiments, we used SIFT features clustered into Z=200 discretized features. The SIFT processing was based on 16x16 pixel patches spaced 8 pixels apart. In this way, each of the dataset images was transformed into a feature map $z_{i,j}$ and then the bag of features c_z was created. In all experiments below we iterated Eqs. 2, 9. In supervised learning tasks, counting grids for individual categories are first learned, and then likelihood comparisons are used for classification. Note that this essentially involves looking for the closets matching histogram $h_{k,\ell}$ in the counting grid for a given bag of features, i.e., even though the spatial layout of feature is used in learning to improve the local minima, it is not used in testing, where the images are treated as disordered bags of features. In unsupervised learning tasks, images of multiple categories (or with various labels) are all used jointly to estimate a large counting grid, and then the physical distance between the mapping to the grid is used as a primary source of information about image similarity.

We performed experiments on three scene datasets: (1) Torralba (OT) [8], (2) Corel Dataset [7], and (3) the 15 categories dataset (LFP) presented in [13]. Torralba dataset has two 4-class collections of images: the natural images, and the human-built environments. This dataset has been subsequently extended into the 15 class dataset [13]. For each dataset and for each category we estimated a counting grid model of size that is larger than the size of an individual image by the factor κ which we varied as $\kappa \in \{1, 1.5, 2.5, 4, 6\}$. Note that the choice ($\kappa = 1$) reduces the counting grid model to a single histogram because the grid is defined on a torus, and so all windows $W_{k,\ell}$ contain all grid locations, making all histograms $h_{k,\ell}$ the same. However, as soon as the grid is even slightly enlarged a large number of different windows and thus new histograms become available. The model, however does not easily overtrain as the parameters $h_{k,\ell}$ follow strong constraints, which we argued above are also present in natural images. In each test, we randomly picked a half of images for training (around 130 per class for the Torralba datasets), and used the remaining half for testing. The mean classification rates were computed over 3 random train/test splits.

In order to compare with the sophisticated latent models of word/feature counts, we studied the literature to find previously reported good numbers for the number of topics T for latent Dirichlet allocation [5]. Based on this we tested the LDA for $T \in [30 \dots 50]$, and reported here only the best result we obtained. Extensions to a larger number of topics is infeasible due to overtraining. But the counting grid can easily capture very large number of histograms. For example, after the described feature mapping procedure, the Torralba images are reduced to 31×31 feature maps from which the feature counts were extracted for the



K	1	1,5	2,5	4	6
CG Natural	75,43	81,39	83,9	85,6	85,93
CG Artificial	76,92	84,38	86,1	87,51	88,73
LDA Artificial	79,1	79,1	79,1	79,1	79,1
LDA Natural	76,5	76,5	76,5	76,5	76,5

K	1	1,5	2,5	4	6
CG	27,31	58,36	62,65	68,24	27,03
LDA	52,5	52,5	52,5	52,5	52,5

K	1	1,5	2,5	4	6
CG	64,22	71,78	74,22	78,9	80,41
LDA	67,22	67,22	67,22	67,22	67,22

Figure 5. Scene classification results for the three datasets considered. CGs outperform with large margin LDA [5]

bag of features representation. Choosing the grid size with the $\kappa = 1.5$ enlargement provides $16 \times 16 = 256$ different histograms $h_{k,\ell}$ in the counting grid, but these histograms are based on shared grid counts in a way that mimics the expected structure in real images, and thus generalize with a much reduced overtraining risk.

Comparisons are reported in figure 5. We found that counting grids did not overtrain until $\kappa > 10$, but the performance only marginally rose beyond $\kappa = 6$. The exception is the Corel dataset where overtraining happens at $\kappa = 6$ because some categories are represented by less than 40 exemplars. Direct generative classification based on counting grids outperforms support vector machines trained on the same input (discrete sift histograms, $\approx 88\%$ on OTs [8] and 72,2% on the 15 classes dataset [13]) and this seems to be better than any other classification based on the generative model for scene classification (including feature epitomes, previously used for similar tasks [3], and which are the special case of the counting grids when the tessellation to obtain section bags is taken to the extreme).

We obtained a further boost in classification when we used the latent structure in an additional discriminative training step [6]: We simply described each sample by a vector of its generative class-likelihoods and trained a linear SVM on this representation. We fixed $\kappa = 6$. The results (Tab.1) are very close the state of the art [9, 18], and match the performance of the spatial matching kernel [13] (81,4% on LFP). The counting grids also outperform techniques from [1, 7, 12] and Fisher kernel on LDA [6].

Table 1. Comparisons of the different CGs M-steps (Eq.6,9), and with State-of-the-art: Spatial Reasoning methods [12, 13, 18], and based on LDA [1, 18]. *[18] uses rbf-SVM.

Algorithm	*OT-Nat*	*OT-Art*	*15-Sc*
CG (Eq. 6)	83,50%	84,83%	59,31%
CG (Eq. 9)	85,93%	88,73%	80,41%
CG+[6]	91,12%	91,77%	82,02%
[13] (Spat. Pyr. Kernel)	n.a.	n.a.	81,4%
[1] (pLSA+KNN)	90,2%	92,5%	73,4%
[18] (Spat. pLSA+SVM*)	n.a.	n.a.	83,31%
[12] (Single Feature)	n.a.	n.a.	80,1%

4.2. Experiments on SenseCam Data and comparison with Eptiome-like models

SenseCam dataset [4] consists of images obtained by a wearable camera, taken at the rate of one frame every 20 seconds during all waking hours of a human subject. Following the procedure from the previous subsection, we analyzed a labeled subset of 300 images, divided in 10 categories[3]. Each category presents images taken in a particular *place* such as house rooms or office environments, or outdoors locations. Images within a category come with significant illumination and viewing angle variations since they are shot at different instants of the subject's life. In [4], the authors found that the previous approaches to scene recognition provide only modest recognition rates.

As the goal of the dataset was to provide the summary of the subject's life, we have trained the counting grids in an unsupervised way (combining images of all categories together) and then investigated if the images are separated in the counting grid in agreement with human labels. We

[3] research.microsoft.com/en-us/um/people/jojic/aihs/

compared with other summarization approaches that lay out the visual input on a larger grid, the epitomic approaches [2, 3, 4]. Epitome and counting grid size was roughly $\kappa = 12$ times the size of the individual image after its reduction to a feature map $z_{i,j}$. Upon learning, each test image was labeled by the label of the closest mapped training image. We used a similar strategy for LDA [5]: We learned a single model and then we used K-NN on the topic proportions. The results are reported in Figure 6.

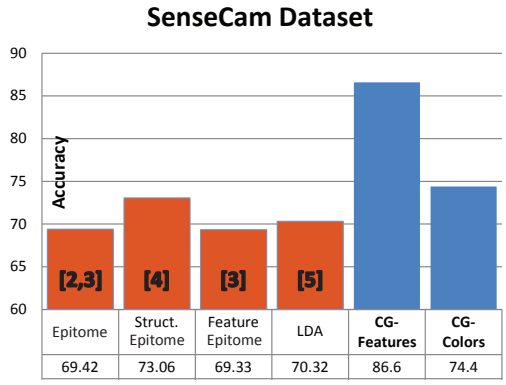


Figure 6. Unsupervised clustering of SenseCam images..

5. Conclusions

We introduced the counting grid (CG) model of images which captures natural constraints on image feature histograms by assuming that these can be represented by averaging of feature distributions from a window into the grid. In CGs, the flexibility of the bag of words representation is enriched (indirectly) by the spatial constraints of epitome-like models. A closer look at the actual observation model, reveals that the counting grid is not attempting to model the spatial constraints in a single test sample explicitly, as has been often done in the past [12, 13, 18, 19]. Instead, the counting grid model uses implicit spatial layout constraints over a a set of training bags of words considered jointly to produce a large mixture of histograms whose parameters are constrained in a way that is a natural consequence of the fact that images from which the features are collected live in an ordered 2D space.

Despite their simplicity, both conceptual and algorithmic (the Matlab code for counting grid estimation fits half a page), and that the ultimate parameterization used for likelihood computation is simply a set of histograms, this generative model significantly outperforms other histogram-based representations in a variety of tasks and is often approaching the discriminative state of the art (and the features extracted from the generative model can often be used within discriminative models to further improve them [6]). Computationally, the algorithm is efficient, and the computational steps also lend themselves to further improvement of the model to add more scale/rotation reasoning.

References

[1] A.Bosch, A.Zisserman and X.Muoz. *Scene Classification Via pLSA*, ECCV 2006 1986, 1991

[2] N.Jojic, B.J. Frey and A.Kannan. *Epitomic analysis of appearance and shape*, ICCV 2003 1987, 1988, 1992

[3] K.Ni, A. Kannan, A.Criminisi and J. M.Winn. *Epitomic location recognition*, CVPR 2008 1988, 1990, 1991, 1992

[4] N.Jojic, A.Perina and V.Murino. *Structural Epitome: a way to summarize one's visual input*, NIPS 2010 1988, 1991, 1992

[5] Fei-Fei Li and P.Perona. *A Bayesian Hierarchical Model for Learning Natural Scene Categories*, CVPR 2005 1985, 1986, 1990, 1991, 1992

[6] A.Perina, M.Cristani, U.Castellani, V.Murino and N.Jojic. *Free Energy score space*, NIPS 2009 1991, 1992

[7] J.Vogel and B.Schiele. *Semantic Modeling of Natural Scenes for Content-Based Image Retrieval*, Int. Jrn. of Computer Vision, 72, 2007 1986, 1990, 1991

[8] A.Oliva and A.Torralba. *Modeling the shape of the scene: A holistic representation of the spatial envelope*, Int. Jrn. of Computer Visionn, 42, 2001 1986, 1990, 1991

[9] M.Bicego, A.Perina et al. *Combining Free Energy Score Space with information theoretic kernels: application to scene classification*, ICIP 2010 1991

[10] G.Csurka, C.R. Dance, L.Fan, J.Willamowski and C.Bray. *Visual Categorization with Bags of Keypoints*, Workshop on Statistical Learning in Computer Vision, 2004 1985

[11] N.Bouguila. *Count Data Modeling and Classification Using Finite Mixtures of Distributions*, IEEE Transactions on Neural Networks 22(2), 2011 1985

[12] T. Harada, H.Nakayama and Y. Kuniyoshi. *Improving Local Descriptors by Embedding Global and Local Spatial Information*, ECCV 2010 1986, 1991, 1992

[13] S.Lazebnik, C.Schmid, J.Ponce. *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*, CVPR 2006 1986, 1990, 1991, 1992

[14] D.Lowe. *Distinctive Image Features from Scale-Invariant Keypoints,* Int. Jrn. of Computer Vision, 60, 2004. 1987

[15] T. Hoffman. *Probabilistic Latent Semantic Analysis*, UAI 1999 1985

[16] R.Neal and G.E.Hinton. *A View Of The Em Algorithm That Justifies Incremental, Sparse, And Other Variants,* Learning in Graphical Models, Kluwer Academic Publishers, 1998 1988

[17] M.R.Boutell, J.Luo and C.M. Brown. *Scene Parsing Using Region-Based Generative Model*, IEEE Transactions on Multimedia 9(1), 2007 1986

[18] E.Ergul and N.Arica. *Scene Classification Using Spatial Pyramid of Latent Topics*, ICPR 2010 1991, 1992

[19] A.Bosch, A.Zisserman and X. Muoz. *Image Classification using Random Forests and Ferns*, ICCV 2007 1986, 1992

[20] P.A.Viola and M.J.Jones. *Rapid object detection using a boosted cascade of simple features*, CVPR 2001 1989