TO APPEAR IN CVPR 2008.

Discriminative Modeling by Boosting on Multilevel Aggregates

Jason J. Corso
Computer Science and Engineering
SUNY at Buffalo
jcorso@cse.buffalo.edu

Abstract

This paper presents a new approach to discriminative modeling for classi cation and labeling. Our method, called Boosting on Multilevel Aggregates (BMA), adds a new class of hierarchical, adaptive features into boosting-based discriminative models. Each pixel is linked with a set of aggregate regions in a multilevel coarsening of the image. The coarsening is adaptive, rapid and stable. The multilevel aggregates present additional information rich features on which to boost, such as shape properties, neighborhood context, hierarchical characteristics, and photometric statistics. We implement and test our approach on three two-class problems: classifying documents in of ce scenes, buildings and horses in natural images. In all three cases, the majority, about 75%, of features selected during boosting are our proposed BMA features rather than patch-based features. This large percentage demonstrates the discriminative power of the multilevel aggregate features over conventional patch-based features. Our quantitative performance measures show the proposed approach gives superior results to the state-of-the-art in all three applications.

1. Introduction

We are interested in the question *Is this pixel a building-pixel? or a horse-pixel? or etc.* This core question has puzzled the vision community for decades; the difficulty stems from the great variability present in natural images. Objects in the natural world can exhibit complex appearance and shape, occur at varying scales, be partially occluded, and have broad intra-class variance. Yet, despite the challenge, good progress has been demonstrated on particular classes of objects, such as detecting faces with Adaboost [16].

Recall that Adaboost [4] defines a systematic supervised learning approach for selecting and combining a set of weak classifiers into a single so-called "strong" classifier. A weak classifier is any classifier doing better than random. Adaboost has been shown to converge to the target posterior distribution [5], i.e., giving an answer to our original question, *is this pixel a...* However, conventional Adaboost in vision relies on features extracted from fixed rectilinear patches at one or more scales. Typical features are Harr-like filters, patch-histograms of Gabor responses and intensities, and position. Features grounded in such patches can violate object boundaries giving polluted responses and have difficulty adapting to broad intra-class object variation.

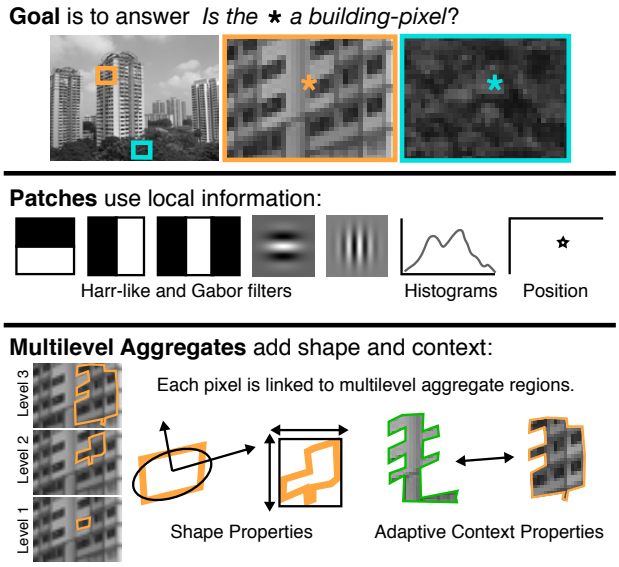

Figure 1. Illustrative overview of the proposed boosting on multilevel aggregates (BMA) approach and comparison to conventional patch-driven approaches. BMA learns discriminative models based on patches and multilevel aggregates, which capture rich hierarchical, contextual, shape, and adaptive region statistical information. Aggregate features are selected about 75% of the time.

In this paper, we propose Boosting on Multilevel Aggregates (BMA), which incorporates features from an adaptively coarsened image into the boosting framework. Figure 1 gives an illustrative overview of BMA. The aggregates are regions that occur at various scales and adapt to the local intensity structure of an image, i.e., they tend to obey object boundaries. For example, in the building case, we expect to find an aggregate for an entire window at one level, and at the next level, we expect to find it joined with

the wall. By linking a pixel with an aggregate at every level in the hierarchy, we are able to incorporate rich statistical, shape, contextual and hierarchical features into the boosting framework without adding a big computational burden or requiring a complex top-down model.

Our work is similar in the adaptive multilevel spirit to Borenstein et al. [2]. In their work, a soft multilevel coarsening procedure, based on the segmentation by weighted aggregation [11], is used to build the hierarchy. The regions in the hierarchy are then used help constrain a model-based top-down segmentation process [3] to give a final class-specific segmentation. However, our goals are different, they rely on a top-down model to jointly constrain a global segmentation energy, which is a recent trend, especially in two-class problems, e.g., [8]. In contrast, we are interested in learning a probabilistic discriminative model from bottom-up cues alone; our BMA model could then be incorporated with top-down information as is similarly done by Zheng et al. [19] or in an energy minimization framework as Shotton et al. [12] do with their TextonBoost model in a conditional random field [9].

A number of papers in the literature deal with boosting in a hierarchical manner. However, these related works mainly hierarchically decompose the classification space rather than the feature space. For example, the probabilistic boosting tree (PBT) method [14] constructs a decision tree by using an Adaboost classifier at each node (we in fact use the PBT as our underlying discriminative model, see section 2.5). The AdaTree method [6], conversely builds a decision tree on the selected weak classifiers rather than a fixed linear combination of them. Torralba et al. [13] use a hierarchical representation to share weak-learners across multiple classes reducing the total computational load; although not our foremost goal, we achieve similar computation sharing in our multilevel aggregates (section 3.5).

Wu et al. [17] propose compositional boosting to learn low-to-mid level structures, like lines and junctions leading to a primal sketch-like interpretation [7]. Their method recursively learns an and-or graph; boosting makes bottom-up proposals, which are validated in a top-down DDMCMC process [15]. SpatialBoost [1] is a related method that extends Adaboost by incorporating weak classifiers based on the neighboring labels into an iterative boosting procedure, during which spatial information can be slow to circulate. In contrast, BMA requires no iterative boosting rounds and it directly incorporates spatial information in the aggregates from the coarsening procedure.

Boosting on multilevel aggregates introduces a new class of information rich features into discriminative modeling. BMA is straightforward to implement, and it can be directly incorporated into existing modeling schemes as well as complete bottom-up-and-top-down methods. We demonstrate BMA on three two-class classification problems: documents in office scenes, and buildings [18] and horses [2] in natural images. BMA is directly extensible to the multi-class case. Our results indicate that BMA features are chosen over patches a majority of the time (about 75%) during learning. Our accuracy on all three problems is superior to the state-of-the-art and validate the modeling power of multilevel aggregates over conventional patch-based methods.

2. Boosting on Multilevel Aggregates

We restate our problem in mathematical terms. Let i denote a pixel on the lattice Λ and $\mathtt{I}(x)$ denote the intensity (gray or color) at that pixel. l_i denotes a binary random labeling variable associated with pixel i taking value $+1$ if i is a building-pixel (or horse-pixel, or etc.) and -1 otherwise. We want to learn a discriminative model $P(l_i|i\ \mathtt{I})$ from which we can compute $l_i^* = \arg\max_{l_i} P(l_i|i\ \mathtt{I})$. To keep the learning and inference tractable, the conditioning on the entire image needs to be reduced. In conventional patch-based modeling, this conditioning is reduced to a local sub-image, e.g., 11×11. However, in the proposed BMA approach, we change the conditioning by replacing the image $\mathtt{I}$ with a multilevel coarsened version G, giving $P(l_i|i\ G)$; G is explained in the next section. Each pixel is then dependent on a greater portion of the image, often as much as 25%. In section (2.1), we discuss the adaptive coarsening procedure. We follow with an example in §2.2, properties in §2.4, and how we train our model in §2.5.

2.1. Adaptive Multilevel Coarsening

Define a graph on the image pixels, $G^0 = (\mathcal{V}^0\ \mathcal{E}^0)$, such that $\mathcal{V}^0 = \Lambda$. Edges in the graph are created based on lattice connectivity relations (e.g., 4-neighbors) and denoted by the predicate $N(u\ v) = 1$. We compute a hierarchy of such graphs such that $G = \{G^t \colon t = 1 \qquad T\}$ by using a coarsening procedure that groups nodes based on aggregate statistics in the image. Associated with each node, $u \in \mathcal{V}^t$, are properties, or statistics, denoted $s_u \in \mathcal{S}$, where $\mathcal{S}$ is some property space, like $\mathbb{R}^3$ for red-green-blue image data, for example. A node at a coarser layer $t > 0$ is called an aggregate W^t and describes a set of nodes, its *children*, $C(W^t) \subset \mathcal{V}^{t-1}$ from the next finer level under the following constraints: $C(W_k^t) \cap C(W_l^t) = \varnothing$ when $k \neq l$ and $\bigcup C(W_k) = \mathcal{V}^{t-1}$. Thus, each node $u \in \mathcal{V}^{t-1}$ is a child of only one coarser aggregate W^t. One can trace an individual pixel to *its* aggregate at each level in the hierarchy G.

We give pseudo-code for the coarsening algorithm in figure 2. Nodes on coarser levels ($t > 0$) group relatively homogeneous pixels. Define a binary edge activation variable e_{uv} on each edge in the current layer, which takes the value 1 if u and v should be in the same aggregate and 0 otherwise. Each coarsening iteration first infers the edge activation variables based on the node statistics (discussed in

detail in section 2.3). Then, a breadth-first connected components procedure groups nodes by visiting active edges until each new aggregate has grown to the maximum size or all nodes have been reached. The maximum size is set by a user-defined parameter called the *reduction factor*, denoted τ in figure 2. There is a inverse logarithmic relationship between the reduction factor and the necessary height (T) required to capture a full coarsened image hierarchy: $T = \lfloor \log_{\frac{1}{\tau}} n \rfloor$, for an image with n pixels. Since a shorter hierarchy requires less computation and storage, and, intuitively, yields aggregates that more quickly capture an accurate multilevel representation of the image, we set $\tau \leftarrow 0\ 05$ in all our experiments, which gives a maximum of 20 children per aggregate and a height T of 4.

Each time an aggregate is created (line 11 in figure 2), it inherits its statistics as the weighted mean over its children. The weight is the fraction of the total mass (number of pixels) each child is contributing. It also inherits connectivity from its children: an aggregate W_1 is connected to an aggregate W_2 if any of their children are connected.

ADAPTIVE MULTILEVEL COARSENING
Input: Image `I` and reduction factor τ.
Output: Graph hierarchy with layers $G^0, \ldots, G^T$.
0 Initialize graph G^0, $T = \lfloor \log_{\frac{1}{\tau}} n \rfloor$, and $t \leftarrow 0$.
1 **repeat**
2 Compute edge activation in G^t; (1) in §2.3.
3 Label every node in G^t as OPEN.
4 **while** OPEN nodes remain in G^t.
5 Create new, empty aggregate W.
6 Put next OPEN node into queue Q.
7 **while** Q is not empty and $|W| < 1/\tau$.
8 $u \leftarrow$ removed head of Q.
9 Add v to Q, s.t. $N(u, v) = 1$ and $e_{uv} = 1$.
10 Add u as a child of W, label u as CLOSED.
11 Create G^{t+1} with a node for each W.
12 Define s_W as weighted means of its children.
13 Inherit connectivity in G^{t+1} from G^t.
14 $t \leftarrow t + 1$.
15 **until** $t = T$.

Figure 2. Pseudo-code for the coarsening algorithm.

2.2. Coarsening Example

We now show an example of the coarsening procedure on a typical natural image scene (from figure 1). The scene has three major region types: sky, building, and tree. In figure 3, we render the four coarse layers of the hierarchy ($T = 4$) horizontally with coarser layers on the right. The top row assigns a random gray value to each unique aggregate. The bottom shows reconstructions of the image using the mean intensity of each aggregate. We note how the perceptual content of the image is preserved even at the coarse levels of the hierarchy. The three different region types coarsen distinctly. The sky coarsens roughly isotropically, due to the pixel homogeneity, but the tree regions coarsen more *randomly*. The building has coarse nodes with very sharp boundaries that can be seen as early as the first level of coarsening (left column). It is variations like these that provide rich information for discriminative modeling of the various region types; such features are discussed in §3.

Figure 3. Example of the adaptive multilevel coarsening on the image from figure 1. See text for explanation.

2.3. Activating Edges During Coarsening

During coarsening, edge activation variables are inferred by making a quick Bayesian decision based on the aggregated statistical properties of each node, including, for example, mean intensity. We consider a statistical interpretation of the affinity between nodes u and v. Given a set of labeled training data, which is assumed for each of the experiments we discuss, we separate the pixels into two pseudo-classes: (1) off boundary and (2) on boundary (separating two class regions, e.g., building and not building). For both pseudo-classes, we compute the distribution on the L1-norm of the statistics, $|s_u - s_v|$. Figure 4-(a) plots the two distributions. The two curves resemble exponential distributions, validating the conventional affinity definition in vision $\exp[-\alpha |s_u - s_v|]$. We compute the Bayesian decision boundary (with equal costs) δ, which is rendered as a dotted vertical line in the figure. Then, we use the rule

$$e_{uv} = \begin{cases} 1 & \text{if } |s_u - s_v| < \delta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

to infer the edge activation variables. The same rule is used at all levels in the hierarchy during coarsening.

2.4. Coarsening Properties

Stability: *Repeated applications of the algorithm* ADAPTIVE MULTILEVEL COARSENING *on an image will yield equivalent coarsened graph hierarchies.* Since the algorithm makes a deterministic decision about activating edges and grouping pixels based on mean aggregate statistics, the coarsening process is stable on equivalent input.

Complexity: *Algorithm* ADAPTIVE MULTILEVEL COARSENING *is log-linear in the number of pixels, $O(n \log_{\frac{1}{\tau}} n)$, in the worst case and linear, $O(n)$, in the typical case for an image with n pixels.* The total number of nodes $\bar{n}$ is bounded such that $\bar{n} \leq nT = n \log_{\frac{1}{\tau}} n$.

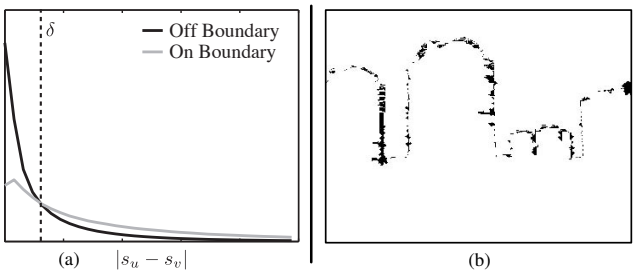

Figure 4. (a) Empirical statistics of feature distance between pixels on and off a boundary, used during edge activation variable inference. (b) Pixels suitable for training on the building class for the image from figure 1. White means suitable.

However, this conservative bound is reached only in the case that all edge activation variables are always turned off, which *never* happens in practice. Rather, the expected total number is $\bar{n} = n + \tau n + \tau^2 n + \cdots + \tau^T n$. Since $\tau \leq 0\ 5 \implies \bar{n} \leq 2n$, the expected complexity is linear. The number of operations per node is constant.

Memory Cost: *Algorithm* ADAPTIVE MULTILEVEL COARSENING *requires memory log-linear in the number of pixels.* The discussion has three parts: i) The cost to store statistics and other properties of a node is constant per node. ii) The cost to store the hierarchical parent-child information is log-linear in the number of pixels (due to the hard-ness of the hierarchy). iii) The cost to store the edges is log-linear in the number of pixels (the number of total edges in the graph is never increasing because of the reduction at each level, and there are log-order levels).

2.5. Training

We use the probabilistic boosting tree (PBT) framework [14], which learns a decision tree using Adaboost classifiers [4, 5] as its nodes, and the standard stump weak classifier [16]. The coarsening algorithm is slightly adapted when we use it for training. Each pixel is given a label l_i. During coarsening when we compute the statistics at a new aggregate W (line 12 in figure 2), we also compute a single label l_W. It is the most common label among W's children:

$$l_W = \arg\max_l \sum\nolimits_{u \in C(W)} \delta(l = l_u) m(u) \qquad (2)$$

where $m(u)$ denotes the mass of node u (number of pixels).

Then, during training, we ag a pixel i as *suitable for training* if it has the same label as the aggregate to which it is associated at each level in the hierarchy. This suitability test is quickly computed by creating a set of label images with one for each hierarchy level such that each pixel is given the label of its aggregate at each level. Because aggregate boundaries tend to obey object boundaries, the percentage of suitable pixels is typically high, about 95% per image; an example of building-suitable pixels for the image from figure 1 is in figure 4-(b).

3. Adaptive Multilevel Aggregate Features

We present a rich set of features that can be measured on the aggregates. With only a few exceptions, evaluating these features is nearly as efficient as the conventional fixed-patch Harr-like filters. These features are measured on the aggregates and capture rich information including regional statistics, shape features, adaptive Harr-like filters, and hierarchical properties. We add the following notation to decribe the properties of an aggregate u (for clarity, we use u instead of W in this section to denote an aggregate):

$L(u)$	set of pixels it represents.
$N(u)$	set of neighbors on same level.
$\min_x(u)\ \min_y(u)$	minimum spatial location.
$\max_x(u)\ \max_y(u)$	maximum spatial location.
$x(u)\ y(u)$	spatial location.
$g(u)\ a(u)\ b(u)$	intensity and color (Lab space).

Recall that these properties are computed during coarsening; no further evaluation is required for them. Where necessary below, we give the mathematical definition of each.

3.1. Photometric and Spatial Statistical Features

Average statistics are computed based on the adaptively coarsened aggregate and avoid polluted statistics that would result if computing them in a patch-based paradigm. During aggregation, they are computed by

$$m(u) = \sum_{c \in C(u)} m(c) \qquad (3)$$

$$x(u) = \frac{1}{m(u)} \sum_{c \in C(u)} m(c) x(c) \qquad (4)$$

The second equation is computed for features y, g, a, and b too. The value for each of these functions at the pixel-level is the obvious one with the initial mass of a pixel being 1.

Aggregate moments take input directly from each pixel in an aggregate. We take the central moment about the aggregate's mean statistic:

$$M_x^k(u) = \frac{1}{m(u)} \sum_{i \in L(u)} (x(i) - x(u))^k \qquad (5)$$

We again compute this for features y, g, a, and b.

Adaptive histograms of the intensities, colors, and Gabor responses are computed directly over the aggregate's pixels $L(u)$. For example, the intensity histogram H_g bin b is

$$H_g(u\ b) = \frac{1}{m(u)} \sum_{i \in L(u)} \delta\left(g(i) - b\right) \qquad (6)$$

Each histogram bin weight is directly considered a feature.

3.2. Shape Features

The spatial moments (5) are simple statistical shape features. Here, we discuss additional aggregate shape features.

Elongation measures the shape of the aggregate's bounding box by taking the ratio of its height to width. The bounding box properties of each aggregate are computed during coarsening by the following equation, for x,

$$\min_x(u) = \min_{c \in C(u)} \min_x(c) \tag{7}$$

The $\max_x$, $\min_y$ and $\max_y$ are similarly computed. A pixel's bounds are its spatial location. Elongation is

$$e(u) = \frac{h(u)}{w(u)} = \frac{\max_y(u) - \min_y(u)}{\max_x(u) - \min_x(u)} \tag{8}$$

Rectangularity measures the degree to which the bounding box of an aggregate is filled by that aggregate. For example, rectangularity is minimum for a single diagonal line and maximum for an actual rectangle. It is defined as

$$r(u) = w(u)h(u) - m(u) \tag{9}$$

where $w(u)$ and $h(u)$ are the width and height from (8).

PCA similarly measures global aggregate shape properties, and indeed the features PCA gives are related to the elongation and rectangularity. We compute the two eigenvectors $\lambda_1(u)$ and $\lambda_2(u)$ of the 2D spatial covariance matrix and use four features from it: the off-diagonal covariance, $\lambda_1(u)$, $\lambda_2(u)$, and the ratio of $\lambda_2(u)$ $\lambda_1(u)$.

3.3. Adaptive Region and Contextual Features

Adaptive relative Harr-like features capture an aggregate's gradient structure, and complement the adaptive histograms defined in section 3.1. Our adaptive Harr-like features are defined in a similar manner to the patch version [16], except that the coordinates are relative to the aggregate's bounding box. Each feature is composed of set of weighted boxes $\mathcal{B} = \{B_1 \quad B_k\}$ with each box B_i being a tuple $\{x_L \ y_L \ x_U \ y_U \ z\}$, with $x_L \ y_L \ x_U \ y_U \in [0 \ 1]$ and $z \in \{-1 \ +1\}$. $(x_U \ y_U)$ is the upper-left corner of the box and $(x_L \ y_L)$ is the lower-right corner. Let $\mathcal{I}$ be the integral image computed from the input image `I`. Then, feature f defined by box-set $\mathcal{B}$ is computed by

$$\begin{aligned} f_{\mathcal{B}}(u) = \sum_{B_i \in \mathcal{B}} z * \Big[& \mathcal{I}(x_U \ y_U) + \mathcal{I}(x_L \ y_L) - \\ & \mathcal{I}(x_L \ y_U) - \mathcal{I}(x_U \ y_L) \Big] \\ x = \min_x(u) + w(u)x \ \text{ and } \ & y = \min_y(u) + h(u)y \end{aligned} \tag{10}$$

The original integral image is directly used to compute the adaptive Harr-like features. These adaptive relative features capture more of a global representation of the gradient structure for a given class type than the standard patch-based Harr-like filters. For example, consider modeling a leopard class; patch-based Harr-like features would give ambiguous responses across most of the image region (because they are measuring local gradients but the texture pattern has a more global nature). In contrast, the adaptive relative Harr-like features, taken at the relevant levels in the hierarchy (we take them at all and let the boosting procedure choose the best), would measure the response with respect to each sub-region, in this case a leopard spot, across the entire region giving a more reliable reponse.

Contextual features capture an aggregate's joint spatial-feature context by measuring similarity to its neighbors. Conceptually, these are neighborhood features that measure affinity at a region-level rather than a pixel-level. Let $D(u \ v)$ be some distance measure on a statistic, e.g., intensity, of aggregates u and v. A min-context feature is

$$f(u) = \min_{v \in N(u)} D(u \ v) \tag{11}$$

and we similarly define max- and mean-context features. The context features serve two purposes: i) they capture differences along aggregate boundaries (e.g., high-intensity sky regions to low-intensity tree regions) and ii) they make a type of homogeneity measurement inside a large object when defined at finer levels in the hierarchy.

3.4. Hierarchical Features

The two hierarchical features capture the aggregative properties of each class. The **mass** of an aggregate $m(u)$ measures rough homogeneity of the region. Sky, for example, is likely to have very high mass (e.g., figure 3). The **number of neighbors**, $|N(u)|$, captures the local complexity of the image region. Building aggregates, for example, are likely to have many neighbors.

3.5. Feature Caching

There are broadly two types of features: those that rely only on aggregate statistics and those that need to perform an operation over the aggregate's pixels. For type-one, the statistics are immediately available from the coarsening procedure. However, for type-two, the aggregate must trace down to the leaves to compute the statistic; this can be burdensome. Fortunately, during inference and training, it must be computed only once for each aggregate rather than for each of its pixels. By construction, multiple pixels will share the same aggregate at various layers in the hierarchy. So, we cache the result of any type-two feature directly at the aggregate after the first time it is computed for an image, achieving some degree of feature sharing.

4. Experimental Results

We implement and test the boosting on multilevel aggregates method on three two-class problems: documents in office scenes, and buildings and horses in natural scenes. We note that BMA is directly extended to the multi-class case as it uses the PBT [14] modeling framework, which is already multi-class. For each of the three datasets, we construct a pool of weak classifiers comprised of patch-based features and the four classes of BMA features (statistical, shape, region, and hierarchical) totalling about 5000 weak classifiers. Before discussing the details about the three datasets, in figure 5 we present a set of histograms describing the different types of features that are automatically selected during the boosting procedure. We see that the majority of the selected features, about 75%, are the proposed BMA features. The performance results in the following sections will further validate the discriminative potential of the proposed BMA features over patch-based and other state-of-the-art methods. In terms of speed, BMA is fast: the coarsening executes in just a few hundred milliseconds for typical vision images (e.g., 400x300). Inference on a such an image takes about a minute for a full BMA-based PBT model and roughly the same amount of time for a patch-only PBT model, indicating that BMA is not adding any significant computational burden into the model.

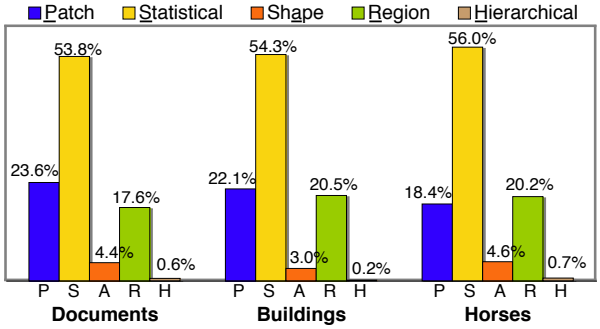

Figure 5. Histograms of the filters automatically selected during the learning procedure for the three problems. The patch-based features are selected only 25% of the time.

4.1. Document Detection

We captured and labeled a set of 550 images of an office scene to support the task of detecting documents for a smart office project. We randomly selected 200 images for training and the rest for testing, and used a single Adaboost classifier rather than the full PBT for simplicity on this problem. While the office environment has less variability than the other two problems, it presents some difficult characteristics (e.g., the computer keyboard) that help elucidate the potential of the BMA method.

In table 1, we show the pixel accuracy scores over the whole testing set for a patch-only classifier and for a BMA classifier. BMA consistently outscores the patch-only classifier by about 6%. This is mostly due to a reduction in false positives. Image results in figure 6 explain where this occurs: typically on the keyboard and the clothing. The key observation is that the patch-based features can only capture local information while the BMA features incorporate more long-range information.

	Patch-based		BMA	
	BG	Doc	BG	Doc
BG	92.8%	7.2%	97.7%	2.3%
Doc	8.8%	91.2%	7.2%	92.8%
Total Accuracy	92.7%		97.3%	

Table 1. Confusion matrices for the document classification problem. Background is abbreviated as "BG" and document as "Doc."

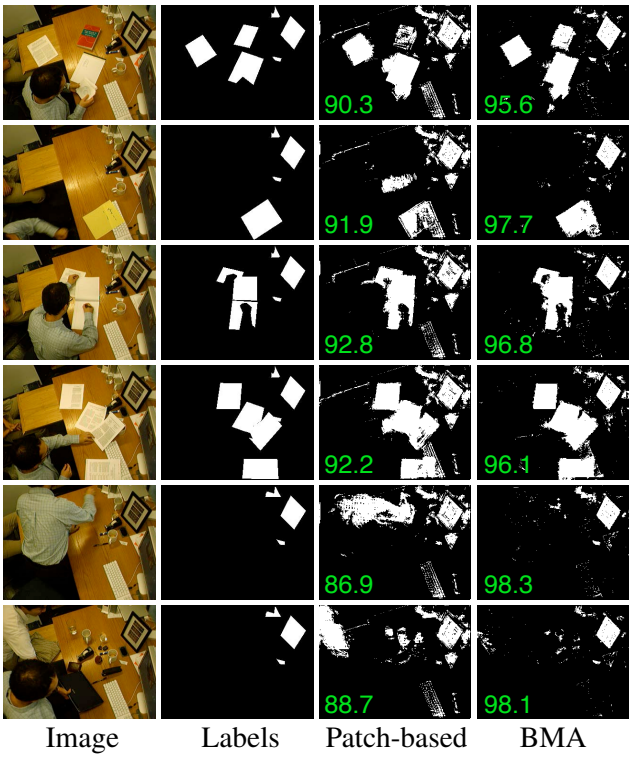

Figure 6. Test image results for the document classification compared with conventional patch-based modeling. The bottom-right two show more difficult cases for the proposed BMA method. The score in green shows the pixel accuracy.

4.2. Building Classification

We use a subset of the recently proposed Lotus Hill Institute dataset of natural images [18] and focus on classifying buildings in typical outdoor scenes. Buildings range from skyscrapers to row-houses to single-standing gazebos and present great variability. We split the 412 image dataset into a training set of 200 images and a testing set of 212 images. We learn two full PBT models: i) using patch-only features and ii) the full BMA features. Figure 7 shows the precision-recall graph of the two models on the testing set. The BMA curve is substantially better than the patch-only

curve; again, the patches have difficulty modeling the variability of building / non-building pixels. Figure 9 shows the inferred probability maps for some testing images. The numeric scores in the top-right corner of each map are the F-measure computed as $\frac{2pr}{p+r}$, where p is precision and r is recall (higher is better). The BMA maps often show a probability very close to 1 for building pixels; we explain this as being caused by the aggregates at coarser layers in the hierarchy representing the majority of the features in the model. These coarse scale aggregates capture very strong features such as region context to neighboring sky and shape moments. In contrast, the patch-only features are restricted to using only local information, mostly texture in this case, which results in patches of grass and tree having high-probability and building-faces with little texture having low-probability. The three image sets on the right column show more difficult cases for both BMA and patches; these cases demonstrate a need for high-level information not incorporated into the proposed BMA method.

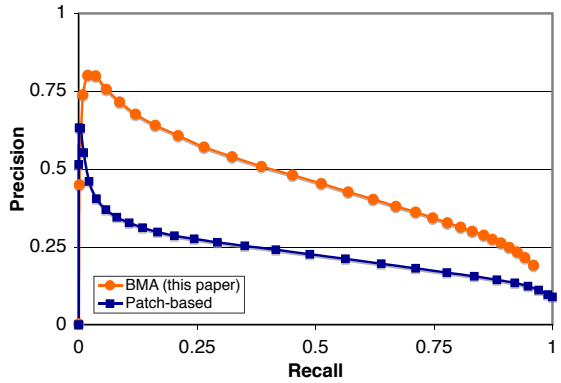

Figure 7. Precision-recall graphs for the building problem.

4.3. Horse Classification

We use the horse figure-ground dataset of [2, 3], taking 100 random images of the 328 for training and the rest for testing. We again learn a BMA-based PBT and a patch-only PBT. We also compare our results to Ren et al. [10] because they provide region precision-recall scores for low and mid-level data (L+M)–which parallels the type of information modeled by BMA (i.e., no high-level model)–and the low, mid, and high-level model (L+M+H). Their method uses gray images only (we test both). We cannot compare directly to [2, 3, 19] because they only provide scores for the horse boundaries, which we are not trying to learn. Our precision-recall graph (figure 8) shows that the BMA features outperform the Ren et al. [10] L+M results and perform almost as good as the L+M+H model which includes high-level features. Interestingly, the patch-only classifier scores roughly the same as L+M; this is reasonable since L+M mainly models local cues. We show some resulting probability maps in figure 10 on testing (color) images.

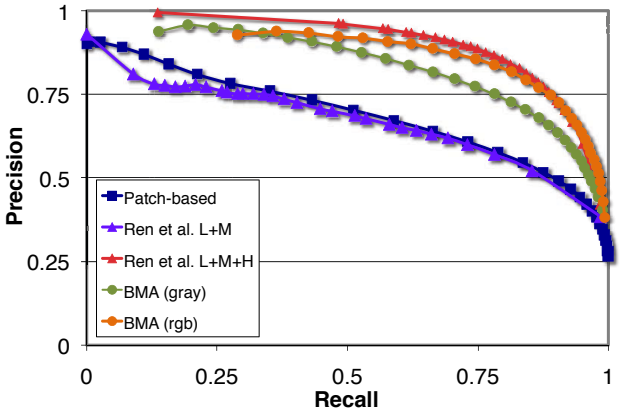

Figure 8. Precision-recall graphs for the horse problem.

5. Conclusion

In summary, we present a new information rich class of features for discriminative modeling. Our approach, called Boosting on Multilevel Aggregates (BMA), links each pixel with a set of aggregate regions at multiple scales, and computes descriptive features on each aggregate, such as statistical, shape, contextual, and hierarchical properties. The aggregates are computed during an adaptive multilevel coarsening procedure, which rapidly decomposes an image into a multiscale graph hierarchy. A Bayesian view of region affinity drives the coarsening process to yield a procedure that is stable, is computationally efficient (log-linear, running time in just a few hundred milliseconds), and tends to obey object boundaries.

We have applied BMA on three two-class problems: documents in office scenes, and buildings and horses in natural scenes. Our analysis indicates that the BMA features are selected about 75% of the time over conventional patch-based features. On all three problems, the BMA model greatly outperforms the patch-only model in quantitative precision-recall and accuracy scores. In the horse problem, we achieve a stronger precision-recall result than the existing state-of-the-art method. In future, we plan to explore BMA in the multiclass case. Since we use the PBT modeling framework, this extension is straightforward. We also plan to incorporate the BMA features as part of the discriminative term in a conditional random field segmentation.

References

[1] S. Avidan. Spatialboost: Adding spatial reasoning to adaboost. In *Proc. of ECCV*, pages 386–396, 2006.

[2] E. Borenstein, E. Sharon, and S. Ullman. Combining Top-down and Bottom-Up Segmentation. In *Proc. of CVPR*, 2004.

[3] E. Borenstein and S. Ullman. Class-Specific, Top-Down Segmentation. In *Proc. of ECCV*, 2002.

[4] Y. Freund and R. E. Schapire. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *J. Comp. and Sys. Sci.*, 55(1):119–139, 1997.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. Tech. Rpt., Statistics, Stanford Univ., 1998.

[6] E. Grossmann. Adatree: Boosting a weak classifier into a decision tree. In *Proc. of CVPR Workshop* v. 6, 2004.

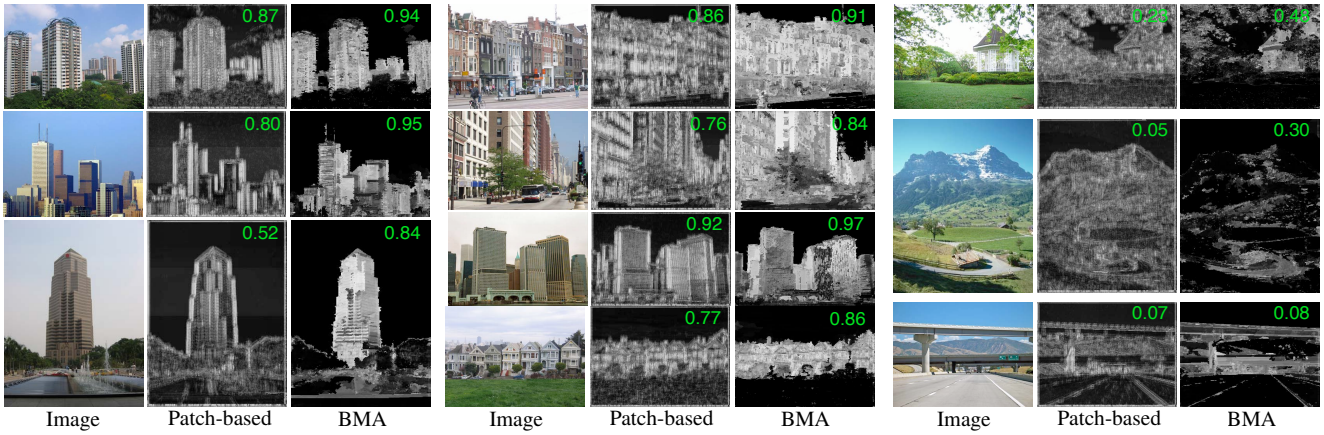

Figure 9. Probability maps for the building problem compared with conventional patch-based modeling (white means probability 1). All images are from the test set except the upper-left one. The best F-measure for each image is displayed in the upper-right corner. Images on the right are difficult for both patch-based and the proposed BMA method. Other images show the benefits of using the multilevel aggregative features that incorporate rich information as shape measurements and context information.

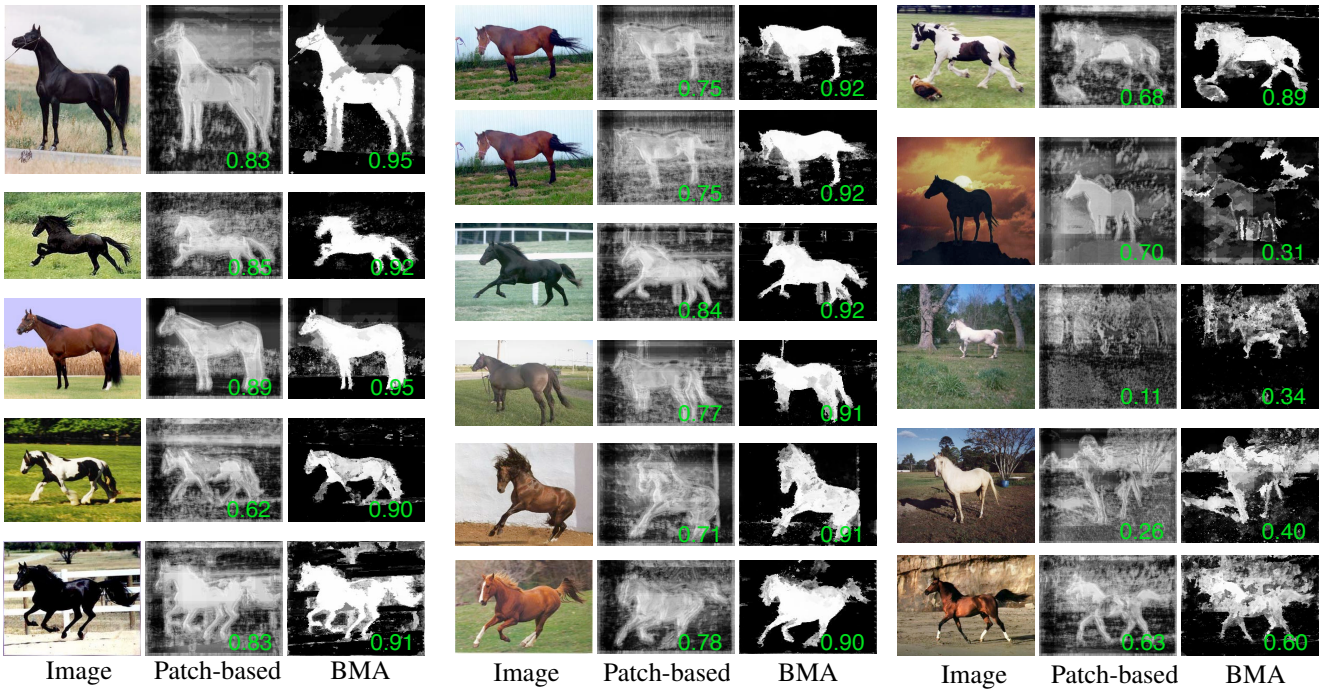

Figure 10. Probability maps for testing images from the horse problem compared with conventional patch-based modeling (white means probability 1). F-measure for each image is displayed in the upper-right corner.

[7] C. E. Guo, S. C. Zhu, and Y. N. Wu. Primal sketch: Integrating texture and structure. *J. of Comp. Vis. and Img. Und.*, 2006.

[8] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *Proc. of CVPR*, pp. 18–25, 2005.

[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML*, 2001.

[10] X. Ren, C. C. Fowlkes, and J. Malik. Cue Integration for Figure/Ground Labeling. In *Proc. of NIPS*, 2005.

[11] E. Sharon, A. Brandt, and R. Basri. Fast Multiscale Image Segmentation. In *Proc. of CVPR*, v. I, pp. 70–77, 2000.

[12] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *Proc. of ECCV*, 2006.

[13] A. Torralba, K. Murphy, and W. T. Freeman. Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection. In *Proc. of CVPR*, 2004.

[14] Z. Tu. Probabilistic Boosting-Tree: Learning Discriminative Models for Classification, Recognition, and Clustering. In *Proc. of ICCV*, 2005.

[15] Z. Tu and S. C. Zhu. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *IEEE Trans. on PAMI*, 24(5):657–673, 2002.

[16] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *Intl. J. of Comp. Vis.*, 57(2):137–154, 2004.

[17] T. F. Wu, G. S. Xia, and S. C. Zhu. Compositional boosting for computing hierarchical image structures. In *Proc. of CVPR*, pp. 1–8, 2007.

[18] Z. Yao, X. Yang, and S. C. Zhu. Introduction to a Large Scale General Purpose Ground Truth Dataset: Methodology, Annotation Tool, and Benchmarks. In *Proc. of EMMCVPR*, 2007.

[19] S.-F. Zheng, Z. Tu, and A. Yuille. Detecting Object Boundaries Using Low-, Mid-, and High-Level Information. In *Proc. of CVPR*, 2007.