Finding Iconic Images

Tamara L. Berg
Computer Science Dept.
SUNY Stony Brook
tlberg@cs.sunysb.edu

Alexander C. Berg
Computer Science Dept.
Columbia University
aberg@cs.columbia.edu

Abstract

We demonstrate that is it possible to automatically find representative example images of a specified object category. These canonical examples are perhaps the kind of images that one would show a child to teach them what, for example a horse is – images with a large object clearly separated from the background.

Given a large collection of images returned by a web search for an object category, our approach proceeds without any user supplied training data for the category. First images are ranked according to a category independent composition model that predicts whether they contain a large clearly depicted object, and outputs an estimated location of that object. Then local features calculated on the proposed object regions are used to eliminate images not distinctive to the category and to cluster images by similarity of object appearance. We present results and a user evaluation on a variety of object categories, demonstrating the effectiveness of the approach.

1. Introduction

Our goal is to automatically find iconic images for object categories by mining large collections of photographs publicly available on the web. Here iconic means a clear and distinctive depiction of an object category in an image – for instance an image that might be used to teach a child about a particular category such as "tiger" or "light house".

That such iconic or canonical views of objects exist for human perception has been demonstrated in Psychology. In their seminal work [17] Rosch and Palmer find that humans agree on canonical views of objects and that recognition is faster for these views. In this paper we develop and evaluate an algorithm to identify iconic images *automatically* by sifting through millions of images from the web.

This algorithm takes a step toward building unsupervised methods for accurate image organization, browsing, and search – ideally the iconic images provide a small number of relevant and representative images that are useful in each of these applications. In addition, the output of the system may be useful for providing an alternative to the human labor intensive process of collecting datasets for recognition research. Such data sets have helped focus research in recognition [8], but few are available due to the expense of collection. The output of our approach while not perfect could reduce the effort required to collect similar and potentially much larger datasets. This combination of the good but not perfect output of an algorithm with human "clean-up" has proven successful with the "labeled faces in the wild" [13] dataset of faces labeled with names which is a cleaned up version of the results from an automatic system [4].

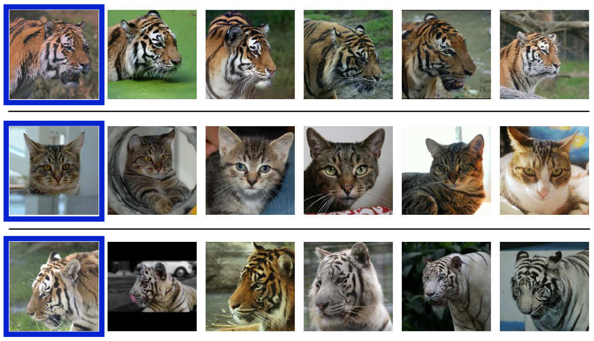

Figure 1. Input to our system is a large pool of images from the web returned by an object category query. No ground truth labeled data or prior idea of what the query object looks like are provided. Despite the fact that most of the images do not show the query object or are poor depictions, our method is able to sift through thousands of images and automatically extract a small number of iconic images that are highly representative of the category, as well as sets of photographs with appearance similar to each iconic representative. This selection is accomplished by considering both image composition and object appearance. Example results for the category "tiger" are shown with iconic images outlined in blue on the left and similar images for each iconic representative shown to the right. Note *many* people refer to their pet cats as tiger.

While the rich variety of images available on the web makes it possible to find representative iconic images, a number of substantial obstacles must be overcome. First automatically finding images that actually depict an object category is difficult. Even with labeled training data (which we do not have), category level object recognition in a general setting is far from a solved problem in computer vision. Furthermore many if not most of the images that show an object category do not do so clearly. Once these challenges have been overcome it is still necessary to find the canonical representatives, the "iconic" images.

This paper presents a computer vision based approach to address the above obstacles. While it might be possible to find objects or iconic images by enlisting people to label large amounts of data we explore the potential of an automated system based on the image content. Furthermore we are interested in a generic approach that can address *object categories*, large variations in how the categories are depicted, and even ambiguity in what constitutes a category – for instance due to polysemy.

We aim to find representative iconic images of concrete object categories, something that has not been addressed in related work. This goal differs from that of Simon *et al*. [20, 15] which relies on geometric constraints to find representative images of specific instances of rigid objects (building facades) instead of general object categories. Our focus on concrete object categories (we explicitly look for images with salient objects) separates our work from that of Raguram and Lazebnik[18] which addresses finding representative images of abstract categories such as "love". The goals of both these papers and our own differ significantly from that of a body of related work on re-ranking the results of a search engine (usually Google) [10, 9, 3, 19, 7]. The goal there is building a model for re-ranking images of a category, not finding good example images of the category. In addition the data in our work (and [20, 18, 15]) comes from `flickr.com` (Flickr) and has not been filtered by the multitude of hyper-link and anchor text features Google implements to rank it's own image and web page search results as used in [10, 9, 3, 19, 7]. In addition the Flickr images are usually photographs that depict objects in natural, complex, and therefore challenging context.

Our approach uses multiple stages to first filter images and then find representatives (fig 2 shows the processing pipeline). First a simple text query for the name of a category is used to retrieve 100,000 images for each of 17 categories from the very large photo sharing site `flickr.com` (Flickr). This is a significant reduction from the billions of photos available on Flickr.

It is important to note that although one might think the problem is solved because humans specified the tags, this is not at all true. In experiments, more than half of the images with a category tag are not representative of the object category, either because they do not depict the category at all or because the depiction is poor or abstract. Note that the criteria here is that an image be a "good representative" not that it simply depicts the object category in any way. Furthermore experiments show that there is too much noise in these initial results for simple clustering to identify iconic examples (fig 4).

The next stage is a novel aspect of our approach – starting not by trying to recognize specific object categories, but by trying to identify images that contain a large clearly shown object of any sort. This is done using a model of image composition based on cues often used for saliency operators (sec 4). Only images that are highly ranked with respect to this measure are considered for later processing. To our knowledge this is the first use of a saliency-like measure for image retrieval, and for finding representative images for an object category. The highest ranked images often contain large clear depictions of objects. Features for later appearance based processing are taken only from the estimated object location.

The resulting images for each query are further automatically filtered to eliminate those that are not distinctive to that query. This is done using a simple k-nearest neighbor test explained in Section 5.1.

The final stage consists of clustering the remaining images to produce results such as those shown in Figure 1. It is important to note that without the preceding stages – filtering out images without distinct object and further removing images that are not distinctive to a category – clustering does not work nearly as well (fig 4).

A large user study evaluates the performance of the proposed system showing favorable results on a set of 17 categories: small (bug), large (lighthouse), textured (tigers), difficult to recognize (sheep, chair) etc. (fig 4).

2. Related Work

To our knowledge, finding iconic images of general object categories has not been addressed in previous research. At the same time there are several areas of related work.

Our approach begins by querying a search engine for images tagged with an object category name and then sifting through the results to find iconic images and sets of similar images. Previous work addressing clustering search results in Content Based Image Retrieval (CBIR) clusters images by content [6] but usually does so after a query by example image (instead of text) and does not focus on an object region or emphasize images with a clear object as we do. Previous work on re-ranking angle image search results starts with a text query and builds classifiers from the noisy results to perform the re-ranking [10, 9], but does not address finding multiple clusters of appearance (polysymy) or stress a clearly depicted object. Work on clustering art images [1] clusters images based on content and associated text, but does not deal with the type of very noisy data sets we collect from Flickr photos. None of these works address choosing a very small number of representative iconic images.

Recently a few papers have addressed finding a small

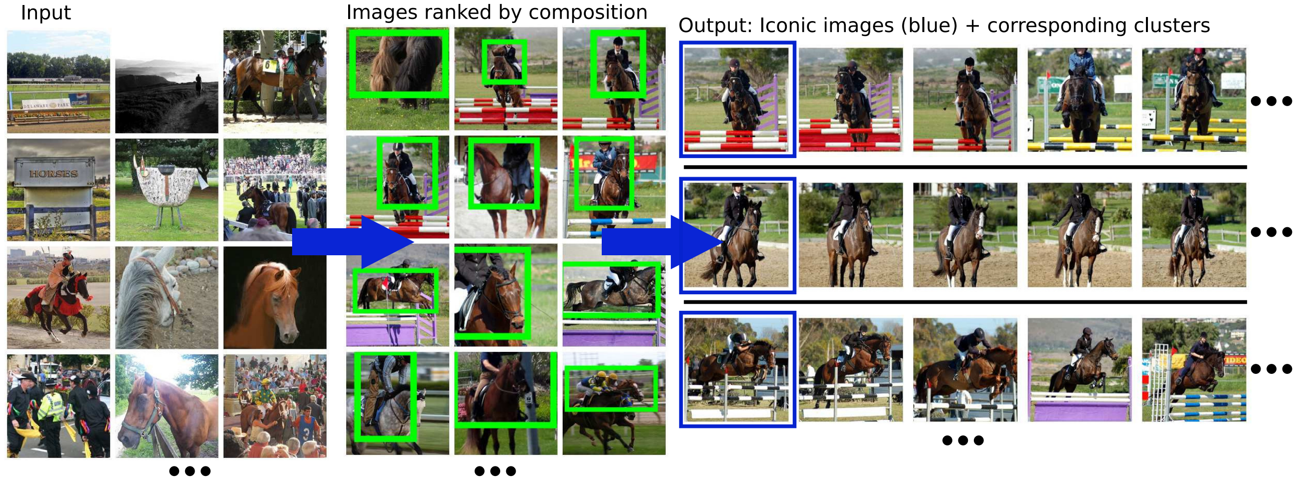
Input
Images ranked by composition
Output: Iconic images (blue) + corresponding clusters

Figure 2. The flow of our system: **Left:** Large pools of images from the web are collected for a specified object category (a random set returned for the query "horse" are shown here). Notice that many of these images do not show the object category or provide unsatisfactory depictions. **Center:** In a single linear pass images are ranked according to a category independent composition model that predicts whether they contain a large clearly delineated object of any category, and outputs an estimated location of that salient object (green boxes). Only the most highly ranked images are retained for more expensive pairwise local feature comparisons. **Right:** Local features calculated on the proposed object regions are used to eliminate objects not distinctive to the category, and to cluster images by similarity of object appearance. Resulting iconic images are shown outlined in blue followed by images with similar object appearance to the right.

number of representative images of landmarks [20, 15]. These concern specific 3D objects (buildings or monuments), and the techniques use constraints available for 3D objects that do not exist for object categories. Related work [18] arranges images returned by abstract queries in a 2D embedding, but does not focus on finding clear pictures of objects or on finding concrete objects at all.

One key aspect of our work is using a simple classifier to find the likely locations of large objects in images and subsequently rank the images themselves by this measure in order to focus on images that may have a large clear object. This is related to work on **saliency**. Most methods for computing saliency are based on bottom up approaches [5, 12, 14], our approach is more top down like that of [16]. We incorporate features similar to their center-surround histograms when computing composition probabilities, but our use of the output for ranking images is significantly different.

Work on object discovery attempts to identify repeated objects in image collections [21, 22]. Kim et al [11] attack this problem by constructing Visual Similarity Networks and inferring information using link analysis techniques. Our approach is more strongly focused on iconic images and does not explicitly use feature correspondence. Generally image datasets for object discovery are less varied than the data our algorithm obtains from Flickr for input.

3. Outline of Approach

We outline our approach for finding iconic images (fig 2 shows the processing pipeline). Numbers corresponding to the experiments are included for clarity. Details for computing composition scores are in section 4. The metric for comparing images is discussed in Section 5. Experiments and discussion are found in Section 6 including comparisons with a baseline technique where steps 2 and 3 are skipped and a random sample of the I_X are used as R_X for each category.[1]

1. For each object category X collect up to 100,000 images I_X associated with X by a search engine (Flickr tag search).
2. For each image i in I_X compute the object/background division with the best composition score. Take the 1000 images in I_X with the highest composition scores, call these selected images S_X.
3. Compare each of the object regions in S_X to each other and to a random sample of 1000 images from all of the other $S_{Y:Y \neq X}$. Throw away any images which have more than 10 of their 20 nearest neighbors in $S_{Y:Y \neq X}$, leaving remaining images R_X.
4. Cluster the images in R_X into $\leq$ 20 clusters with cluster centers $c_{X1} \ldots c_{X20}$ using k-medoids, throwing out small clusters. These selected cluster centers $c_{X1} \ldots c_{Xr}$ are the iconic images.

4. Ranking by Composition

The subject of interest in an iconic image should be large and easily separated from its background. We have developed a class independent model of image composition to

[1] Specific numbers are given for clarity with the understanding that they can be altered without fundamentally changing the algorithm.

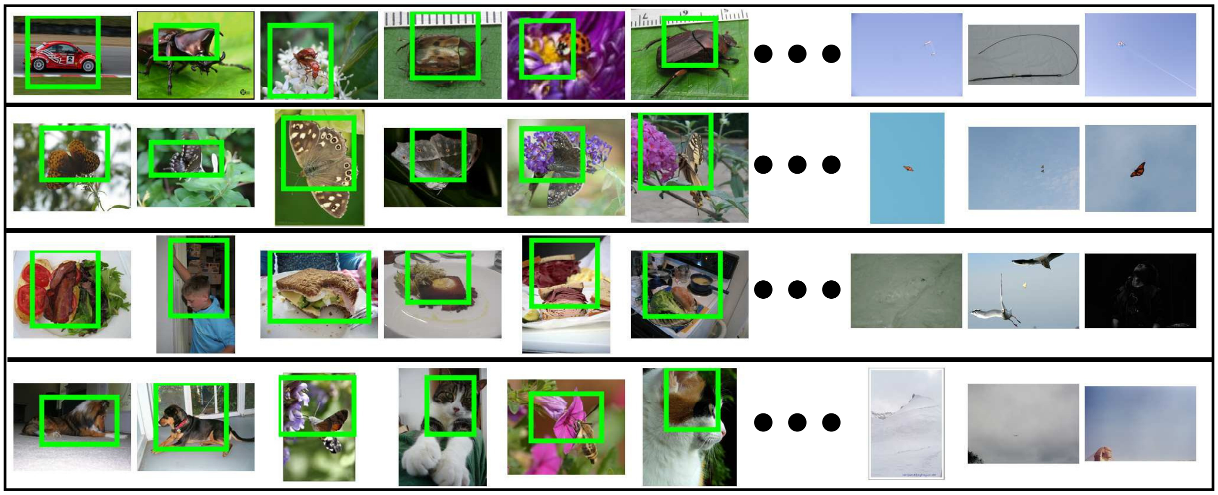

Figure 3. Images ranked by our class independent composition model that predicts whether an image contains a large, clearly delineated object of any category and outputs an estimate of where that object is located (green boxes). Rankings are shown top to bottom for the beetle, butterfly, bread, and sphinx categories. Images containing a large salient object tend to be near the top of the rankings (left), while images at the bottom of the rankings (right) tend not to contain any salient object or only contain small inferior depictions. Because our composition model is class independent, the ranking is based only on image layout not object category. Thus, we are not guaranteed that those images at the top of the ranking will contain the specified object. However, we exploit the fact that the images were collected because they had been associated with the specified category keyword, making it likely that some of these highly ranked images will contain good examples of the object category.

evaluate how well a particular image fits this criteria and predict the object location.

This model is learned solely from a set of training images that do not overlap in subject (or image) with the test categories. No category specific information is used and the single learned model is applied to all object categories.

In this paper we consider layouts consisting of a foreground rectangle with the remainder of the image as background. The model examines all possible layouts for an image and the highest score for each image is used to re-rank the images obtained for an object category. Only features from the foreground rectangle that resulted in the highest score are used for later processing (steps 4 & 5 above).

This stage of processing helps in several important ways:

- Percolates the good/interesting images up in the ranking.
- Provides a rough division of the image into object and background regions.
- Eliminates "junk" images that can confuse clustering.
- Can be performed efficiently on the enormous sets of images available on the web because it is linear in the number of images and independent of category.

4.1. Composition Model

We use Naive Bayes to model image composition. There are factors in the model for object appearance, background appearance, and appearance contrast between object and background. Five of the features used are related to perceptual contrast: hue (H), saturation (S), value (V), focus (C), and texture (T). We also use two cues directly related to the spatial nature of iconic compositions: object size and location.

For any given layout cues are computed on the foreground object rectangle and on the background region (remainder of the image). Hue, saturation and value cues are modeled as histograms with 11 uniformly spaced bins. Focus is computed as the ratio of high pass energy to low pass energy. Texture is modeled as a histogram (with 11 uniformly spaced bins) of total response to a set of 5 oriented bar filters and a spot filter (square-root of sum of squares of filter responses).

Though there are many possible layouts for an image ($\text{rows}^2 * \text{cols}^2/4$ divisions into object and background), we compute the features for all possible layouts efficiently using summed area tables, making the composition model very fast to evaluate.

The probability of any given layout, L, with features, F:

$$\begin{array}{rcl} P(L|F) & = & \begin{array}{c} P(L)\prod_i P(F_i|L) \\ P(F_1,F_2,...F_n) \end{array} \\ & = & \begin{array}{c} P(L)\prod_i P(F_i|L) \\ P(L)\prod_i P(F_i|L) + P(L)\prod_i P(F_i|L) \end{array} \end{array}$$

For simplicity, we assume that $P(L)$ and $P(L)$ are equal.

Naive Bayes Features: We train 6 probability distributions for each type of image cue. These distributions describe, for both good and bad layouts: the distribution of cue contrast computed using Chi-Squared distance, the distribution over object histograms, and the distribution over background histograms.

For the contrast distributions we simply histogram the observed Chi-squared distances (between object and background histograms) over the training images and learn the distribution of values, for both correct and incorrect layouts. This gives 5x2 features for our model: $P(H_c|L)$, $P(H_c|\overline{L})$, $\ldots P(T_c|L)$, $P(T_c|\overline{L})$.

For the object and background distributions, we learn the distribution over histogram values for each bin independently and then compute a final probability as the product of probabilities over the bins. This gives us 5x2 features for the object model: $P(H_o|L)$, $P(H_o|\overline{L})$, $\ldots P(T_o|L)$, $P(T_o|\overline{L})$, and $5x2$ features for the background model: $P(H_b|L)$,$P(H_b|\overline{L})$, $\ldots P(T_b|L)$,$P(T_b|\overline{L})$.

For the distribution related to object size and location, we bin the object region size and location from the training images into a normalized 4-d histogram. The probability of any given size and location of a layout, $P(SizeLoc|L)$ can then be computed by a lookup in this table. Because any incorrect layout is equally likely, $P(SizeLoc|L)$ is set to one over the total number of possible layouts.

Training Data: We have trained the Naive Bayes composition model using a single set of 500 hand labeled images selected as examples of good compositional layout from a set of random Flickr images uploaded in January of 2007. For each of these training images we hand label the correct layout and one random incorrect layout. For 100 of the images we also select a sky region as an extra incorrect object region, because dividing an image into sky vs everything else will often have high contrast, indicating incorrectly a good layout.

4.2. Ranked Results

Though finding a division between foreground object and background is in general an extremely challenging open problem, our specific problem - finding this segmentation for iconic images - is often somewhat easier because by definition in an iconic image this division should be clear. By ranking the images according to how well they separate into object and background we have effectively percolated those images most likely to be iconic to the top of the results, and focused on those images where the segmentation is most likely to be accurate.

Highly ranked images based on our compositional model tend to contain good visual examples of some object clearly delineated from its background (fig 3 left), while images at the bottom of the ranking (fig 3 right) tend not to contain any salient object or contain small or inferior depictions. The predicted best layout for each image is shown in green and tends to be quite effective at delineating the salient object in each image. Since the model is category independent these images are not guaranteed to contain any specific object. However, we exploit the fact that the input images were collected based on a shared tag, making it likely that these highly ranked images will contain some good examples of the object category (e.g. beetles or butterflies in fig 3).

5. Analyzing Object Appearance

For each category we select from the entire set of (up to 100,000) images, the 1000 most highly ranked images that the composition model has predicted to consist of a large object well separated from its background. In addition to selecting images with potentially good object representations, this reduces the number of images to be considered considerably. We can now afford to use more complex local feature based methods for comparing object appearance something which might have been too computationally intensive to apply to the entire collection.

We would like to find modes in the distribution of object appearances (appearances that occur frequently in the set) as these are likely to correspond to representative examples for the class. We would also like to be robust to changes in background appearance. To accomplish these two goals we first filter out images that are not distinctive to the category, and then find sets of similar images using a k-medoids clustering based on local features computed only within the proposed object regions. The medoids of these clusters are presented as the iconic images for the object category and images with similar appearance to each iconic image can be explored by accessing the corresponding cluster.

Geometric Blur Features: are shape descriptors [2] that have been shown to perform quite well on object recognition tasks [24, 25]. We use these as our measure of local appearance for clustering. The geometric blur descriptor first produces sparse channels from the grey scale image, in this case half-wave rectified oriented edge filter responses at three orientations yielding six channels. Each channel is blurred by a spatially varying Gaussian with a standard deviation proportional to the distance to the feature center. The descriptors are then sub-sampled and normalized.

Similarity Measure: We measure similarity between two images using a spatially restricted feature match score. For each feature in image i, f_i^k, we find its best match in image j, f_j^l, where features can only match to features within a radius of 30% of the diagonal of the estimated object region. The similarity between image i, and image j is then the mean best match score over the set of features:

$$S(i,j) = \frac{1}{n}\sum_k max_l(sim(f_i^k, f_j^l))$$

where n is the number of features in image i, and feature similarity is computed by normalized correlation. We further symmetrize the similarity matrix, S, as $(S + S^T)/2$.

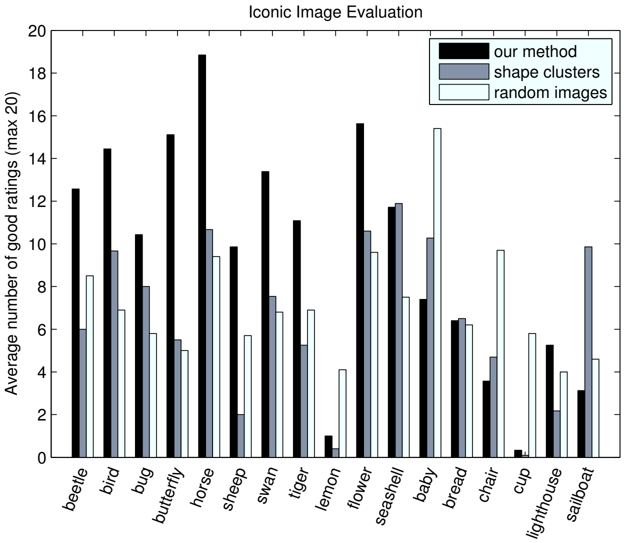

Figure 4. User evaluation results. For each object category, we show users a random permutation of images from – our full method, a "shape" only simplification that does not use layout, and randomly selected images. We show each category to 20 users and performance is reported as the number of users that selected each image as a good representative divided by the number of iconic images output by the method (max score is 20 if every user selects every image for a method). Our method performs quite favorably for many of the object categories (e.g. horse, sheep, swan, bird, etc). Notable exceptions include cup and baby where for instance Flickr tags consistently denote sporting "cups" or pets called "baby" that our human evaluators did not consider good representatives. Figure 5 shows that the iconic images are still quite reasonable, and in fact indicates that the user evaluation probably under-estimates the advantage of our full system over the shape only version. See section 6.2 for more discussion of the evaluation.

5.1. Finding distinctive images

We want images that are distinctive with respect to the specified object category. The composition based ranking algorithm is category independent so some or many of the highly ranked images will not depict the category. We remove many of these "junk" images with a simple density estimate. Using the similarity measure on geometric blur features just described, we compare each image to the 1000 most highly ranked images of the same query, and 1000 highly ranked images from the other object queries. If more than 50% of the 20 nearest neighbors for an image are out of class the image is removed.

5.2. Clustering

We use k-medoids clustering with $k = 20$ and the similarity measure defined in section 5 to find representative images and their corresponding clusters for each object class. Small clusters (≤ 10 images) are removed and clusters are ordered for presentation by the mean similarity of images within the cluster to the medoid image.

6. Results

For evaluation, we consider 17 object categories and compare our method for selecting iconic images to two others: 1. Images selected at random from the set of images for each category. 2. Images selected by a baseline clustering. The baseline clustering algorithm, referred to as "shape" clustering in the experiments, is essentially a handicapping of our method without the composition model or distinctiveness filtering so that we can see how much these steps contribute to the results – First 1000 images are selected at random from the set for each category (instead of images highly ranked by the composition model). Then geometric blur features are computed across the whole image (instead of just on the object regions). Finally k-medoids clustering is used to find the representative images with the same similarity measure used by our system.

6.1. Qualitative Evaluation

The medoid images from each cluster form the iconic images for a category see Figure 5 for examples. Notice that these images show a variety of representations for each category, including representative images for quite challenging categories like bird. Birds vary greatly between photographs with neither a distinctive texture (like tigers) nor a very repeatable appearance (like horses). Despite this variation we are able to find good representative images including: bird heads, birds in flight, perched birds and even a cooked bird.

For categories where the tag is inherently ambiguous, our method selects representative images showing commonly labeled senses. For example the selected images for the tiger category include wild tigers as well as house cats called tiger, representatives for the beetle category depict insects as well as Volkswagens, and representatives for the cup category show images from various sporting cups

Each representative image is the medoid of a cluster of images with similar appearances that can be browsed for related images. A few of these clusters are shown in figures 1 and 2. In many cases the clusters show coherent object appearances.

6.2. Quantitative Evaluation

We evaluated the relevance of our iconic images using Amazon's Mechanical Turk service which provides access to a large body of users for a small fee per task. Part of the code for doing this evaluation was graciously provided by Alex Sorokin [23]. For each of the 17 categories we created a HIT (human intelligence task) consisting of a single web page displaying all images output by: our method, the baseline clustering method, and 10 images selected at random from the input images. These images were randomly ordered on the page and users were asked to evaluate them. For example, for the category horse the instructions were:

Click on all images that show good representative examples of the category "horse". The horse should be:

- *Large (covering at least $\frac{1}{4}$ of the picture)*
- *Easily identified*
- *Near the center of the photo*

Figure 4 shows the results of our user evaluation on the 17 object categories with 20 users evaluating each category. For each method we plot the number of users that selected each image as a good example divided by the number of iconic images our algorithm produced for the category (max score would be 20 if every user clicked on every image in the category).

For many of the object categories (horse, sheep, tiger, bird, swan, flower, butterfly, beetle, bug, seashell, lighthouse) our method performs quite favorably compared to both the randomly selected images and the baseline clustering method. This implies that our methods for analyzing image composition and finding images distinctive to a category are helpful for selecting representative images. These methods work best when the object is self-contained and has distinctive visual characteristics.

For some categories there is an inherent disconnect between what photographers label with an object category and what users click on as representative photographs. For example, many photograph owners tag their house cats as "tiger", but users evaluating an iconic image corresponding to the house cat images will not label them as good examples of the category "tiger". For "cup" many of the images within the collection depict sporting events like the World Cup, car racing or horse racing cups. For lemon many of the images depict foods prepared with lemon. So, while we produce coherent clusters and good representative iconic images they are not always marked as relevant by the human evaluators. As a result the quantitative evaluation may underestimate the success of the approach, as can be seen in the contrast between the representative images chosen by the "shape"only method shown in Fig. 6 and the results of our full method in Fig 5.

6.3. Conclusion

We presented an approach to automatically find iconic images for object categories, with surprisingly good results. A user study verifies this performance on a variety of object categories despite variations in appearance, pose and polysemy (beetle can refer to either a bug or a Volkswagen). This is a promising initial step toward building unsupervised methods for accurate image organization, browsing, and search. Another potential use of our system is to automatically build enormous labeled datasets of object categories for developing recognition systems. One key feature is that our layout analysis is category independent so processing is linear in the number of initially retrieved images. Only a small fraction that are likely to contain large clear objects need be clustered.

The full set of thousands of results is available online. [2]

References

[1] K. Barnard, P. Duyguly, and D. Forsyth. Clustering art. In *CVPR*, June 2001. 2

[2] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, June 2001. 5

[3] T. Berg and D. Forsyth. Animals on the web. In *CVPR*, 2006. 2

[4] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, E. Learned-Miller, Y. Teh, and D. A. Forsyth. Names and faces. In *CVPR*, 2004. 1

[5] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *NIPS*, 2005. 3

[6] Y. Chen, J. Wang, and R. Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *Transactions on Image Processing*, pages 14: 1187–1201, 2005. 2

[7] B. Collins, J. Deng, L. Kai, and L. Fei-Fei. Towards scalable dataset construction: An active learning approach. In *ECCV*, 2008. 2

[8] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. In *PAMI*, In Press. 1

[9] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, Oct. 2005. 2

[10] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, May 2004. 2

[11] K. Gunhee, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories. In *CVPR*, 2008. 3

[12] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006. 3

[13] G. B. Huang, M. Ramesh, T. L. Berg, and Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *UMASS tech report 07-49*, 2007. 1

[14] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 1998. 3

[15] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV*, 2008. 2, 3

[16] T. Lui, J. Sun, N.-K. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. In *CVPR*, 2007. 3

[17] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. In *Attention and Performance*, 1981. 1

[18] R. Raguram and S. Lazebnik. Computing iconic summaries for general visual concepts. In *1st Internet Vision Workshop*, 2008. 2, 3

[19] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007. 2

[20] I. Simon, N. Snavely, and S. Seitz. Scene summarization for online image collections. In *ICCV*, 2007. 2, 3

[21] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *ICCV*, 2005. 3

[22] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008. 3

[23] A. Sorokin and D. A. Forsyth. Utility data annotation with amazon mechanical turk. In *1st Internet Vision Workshop*, 2008. 6

[24] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007. 5

[25] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, June 2006. 5

[2] Some of this research was performed while the authors worked for Yahoo! Research.

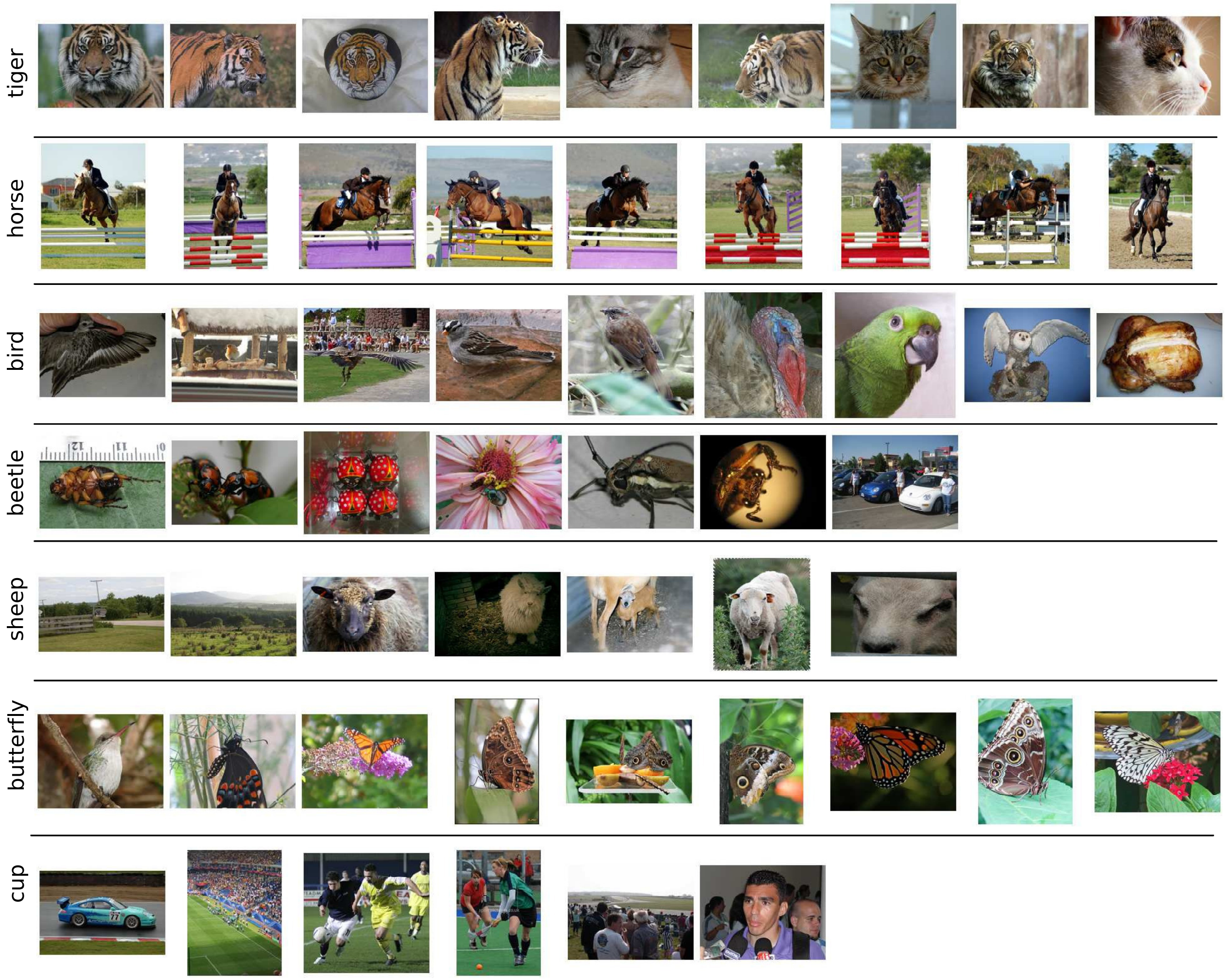

Figure 5. Iconic images selected by our system. Many show canonical representatives of the object category (e.g. horse, tiger, bird, and beetle images). In the selected images the specified object tends to be the main subject of the photograph and often nearly fills the entire image. Even for object categories that vary widely in appearance such as bird the algorithm finds a variety of iconic representations – birds in flight, bird heads, perched birds, and even cooked birds. For ambiguous object categories the output contains representatives depicting the various senses (e.g. tiger images show wild cats and house cats, beetle images show insects and cars, cup images show representative images from various sporting cups).

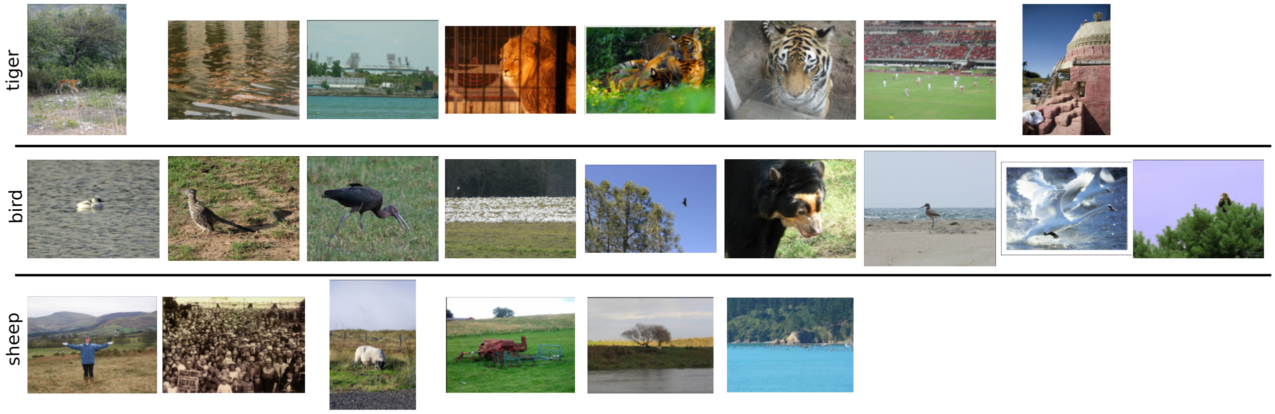

Figure 6. Images output by the "shape" version of our system that does not filter by layout. Note that the images selected often do not show large clear objects. Compare to the results of our full system using layout in Fig. 5.