Learning Crop Regions for Content-Aware Generation of Thumbnail Images

Lyndon Kennedy, Roelof van Zwol, Nicolas Torzec, Belle Tseng
Yahoo! Labs
4301 Great America Parkway
Santa Clara, CA 95054
{lyndonk,roelof,torzecn,belle}@yahoo-inc.com

ABSTRACT

We propose a model for automatically cropping images based on a diverse set of content and spatial features. We approach this by extracting pixel-level features and aggregating them over possible crop regions. We then learn a regression model to predict the quality of the crop regions, via the degree to which they would overlaps with human-provided crops from these input features. Candidate images can then be cropped based an exhaustive sweep over candidate crop regions, where each region is scored and the highest-scoring region is retained. The system is unique in its ability to incorporate a variety of pixel-level importance cues when arriving at a final cropping recommendation. We test the system on a set of human-cropped images with a large set of features. We find that the system outperforms baseline approaches, particularly when the aspect ratio of the image is very different from the target thumbnail region.

Categories and Subject Descriptors

I.4.0 [**Image Processing and Computer Vision**]: General—*Image processing software*

General Terms

Algorithms, Experimentation

Keywords

image cropping, visual saliency, content analysis

1. INTRODUCTION

As visual information continues to be generated and disseminated at an increasingly rapid pace, real applications and systems also continue to emerge to help users browse, search, and digest all of this information. A common visual design metaphor across many of these systems is the usage of collections of smaller, cropped versions of images ("thumbnails") to represent a summary of visual content that might be found after clicking on an object, hyperlink, or group of media objects. The intelligent selection of crop regions for thumbnail images is vital for communicating the expected content to the users and guiding them as they navigate through multimedia information.

The thumbnail may be further required (due to graphic design constraints) to be at a different aspect ratio than the original image. This requires cropping out certain portions of the image, which might omit content and obscure the meaning of the original image. Therefore, there is a need for intelligent, content-aware approaches to cropping images for generating thumbnails.

In Figure 1, we see a number of examples of current commercial systems where thumbnails are selected, cropped, or otherwise generated through some automatic criteria. The driving force behind much of the automatic cropping that occurs is to fit non-uniformly-sized images into some predefined uniform shape in order to provide a more visually appealing display. In the case of images on photo sharing websites, such as Flickr, the incoming images might frequently already be constrained by the sizes typical of common digital cameras, the differences lying mainly in whether the images are vertical portraits or horizontal landscapes. Similarly, video sites might select a single keyframe as a thumbnail for an entire video. The size of the source video is likely to fall within a few pre-defined classes, such as 16:9 for newer HDTV television shows and 4:3 for older standard definition ones. The sets of possible proportions might vary much more significantly, however, if the sources of images are more diverse, such as general web images displayed along with image search refiners on Yahoo! search or videos captured from mobile devices and shared on YouTube. The algorithms deployed to generate the thumbnails in these systems are not disclosed publicly, but they appear to simply place the largest possible bounding box of the desired aspect ratio within the limits of the image and evenly crop out areas around this region.

In this work, we propose a machine-learned approach for the automatic generation of cropped thumbnails to represent larger, disparately-proportioned images. We implement this by defining pixel-level feature maps, which capture the relative importance of a given pixel region based on its adjacency to any number of visually-important features of the image, including: the spatial center of the image, face regions, interest points, and saliency maps.

For each of these pixel-level features, we can aggregate their values within any proposed region or bounding box. In essence, a good cropping bounding box should encap-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ICMR '11, April 17-20, Trento, Italy
Copyright ©2011 ACM 978-1-4503-0336-1/11/04 ...$10.00.

RIGHTSLINK

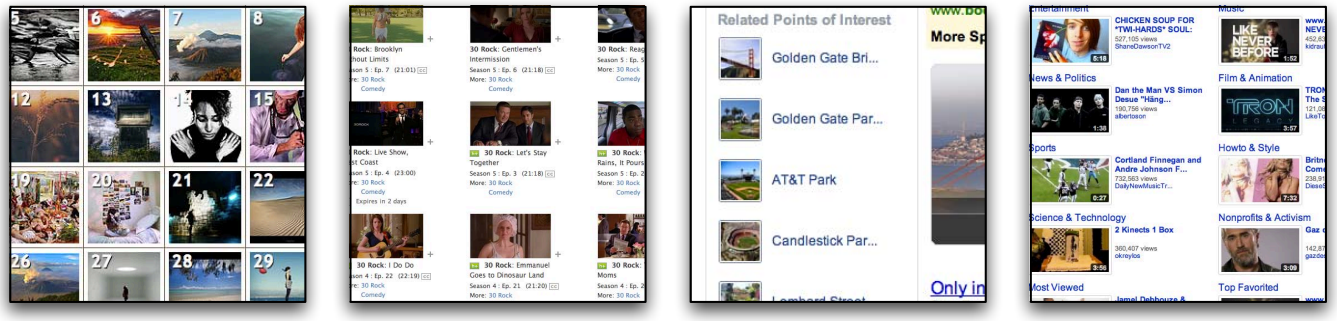


Figure 1: Examples of automatic thumbnails generated in a number of current real applications: (a) Flickr (http://flickr.com) explore page, (b) Hulu (http://hulu.com) show page, (c) Yahoo! Web Search refiners (http://search.yahoo.com), and (d) YouTube (http://youtube.com) recommended videos.

sulate as large of a proportion of the total saliency of an image as possible; however, the measures of saliency come from diverse cues, so we must also learn how to balance the trade-offs between various cues. For example, bounding boxes that cover the center of the image are frequently likely to capture the essence of the image, but if there is a face in a distant corner, that may be a stronger cue for use in region selection. We need to arrive at a method for taking these various cues into account when selecting optimal crops. To do this, we define an objective function to measure the quality of a possible bounding box with respect to human-generated ideal bounding boxes. We then learn a regression model to map between the diverse visual measures of bounding box quality to the ground truth score. Given a new, unseen image, we can apply this model to various candidate bounding boxes and select the top-scoring box for generating the thumbnail.

We apply this method to a collection of images, covering several classes of entities, with hand-labeled crop regions. We find that machine learned models can outperform well-designed (though still naive) baselines. We further find that the contributions of the approach are largest when the images are further from the target aspect ratio. The proposed method frequently approaches the level of accuracy of human annotators. We evaluate the efficacy of individual types of features and find that traditional visual saliency features are, indeed, useful selecting cropping regions, but the most useful features rely on the spatial location of the crop region within the image. We further find that the optimal combination of features changes somewhat based on the class of the image at hand: an image of a person should be cropped using different criteria than an image of a point of interest.

The primary contribution of this paper is a machine-learned approach for cropping images using a diverse set of cues and an investigation into the relative utility of these cues across various types of images.

The remainder of the paper is organized as follows. We provide an overview of the system in Section 3. We describe our experimental settings in Section 4 and discuss the results in Section 5. Section 6 provides conclusions and directions for future work. First, we will proceed by providing a review of related work.

2. RELATED WORK

Beyond the basic approaches deployed in the systems detailed above and shown in Figure 1, some previous work has also considered various aspects of image content as a method for cropping and resizing images.

Much of this work has considered the image in terms of a saliency map, either as measured directly by saliency features or indirectly by face detection [12, 5, 11, 4, 14]. The methods then apply a bounding box region over this saliency map that captures as much of the saliency as possible. These approaches, however, rely mostly on heuristics and conjecture for the relative importance of these saliency regions and how much of this saliency energy ought to be captured by the cropped image. The approach from Marhcesotti et al. [9] is the most similar to our approach in that both use a machine-learned approach to learn the visual qualities of crop regions from human-labeled training sets. This previous work, however, uses the input labels as a mechanism for learning a new saliency map that reflects the preferences of human croppers and therefore still requires some heuristic approaches to forming a bounding box over this new learned saliency map, much like the above-mentioned prior works. Our work, on the other hand, aims to learn the crop region via diverse cues extracted directly from the training examples, not just a new saliency map.

A great deal of recent work has also been leveraging the notion of "seam carving" [1, 10] for resizing images. Here, the resizing can be done adaptively, cutting away "seams," or 8-connected paths of pixels across the images, choosing paths through low-saliency regions of the images. Since these seams may be from anywhere in the image, this approach is fundamentally different from cropping, where a unified region must be selected. We mention this work, though, because it is related in its usage of saliency maps. Indeed many new works on generating better saliency detection approaches choose to evaluate the methods in terms of their utility as an input map for seam-carving approaches.

All previous work also seems to focus on the utilization of a single saliency feature (or one unified from various sources), as an input for creating cropped or seam-carved images. In this work, we present a method for considering a number of sources and learning the relative efficacy of each of the individual features.

RIGHTSLINK

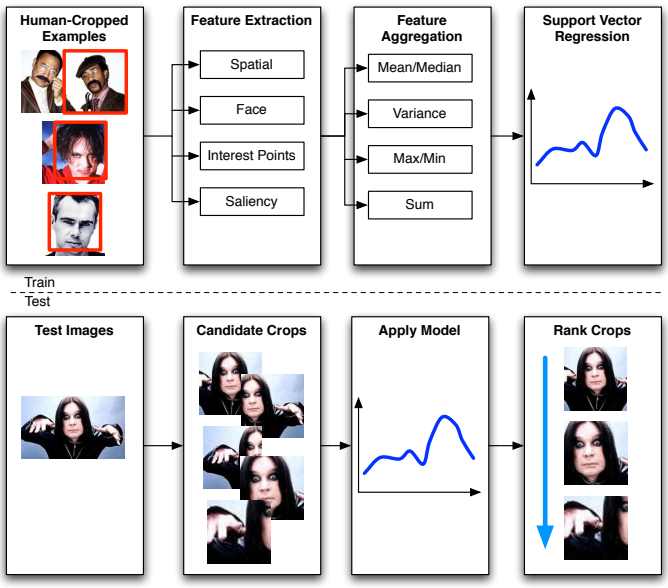


Figure 2: Architecture of the proposed cropping system. Pixel-level features are extracted and then aggregated over candidate boxes. A support vector regression model is learned from training examples. A sweep across candidate crops of test images is conducted. The highest-scoring crop is selected.

3. PROPOSED SYSTEM

We propose a system for learning to predict the quality of any given crop of an image based on the spatial location and visual properties of the pixels that it contains. The system allows for any number of features to be incorporated and it is designed to match crops that would be selected by human labelers. An overview of our proposed system is shown in Figure 2.

The system is data driven, so the cropping models are learned based on images that are manually cropped by humans. For any given image, we can treat the crop regions provided by humans as the ideal crop. We can further score any candidate crop region against this ideal region based on the degree to which it overlaps with the ideal region: an exact match would be best, a completely different region would be the worst, and some partial overlap would be in between. Given a candidate crop region, we can learn a regression model to predict the quality of the crop (the degree to which it would overlap with the human label) from an input feature space representing the characteristics of the visual content of the crop.

The input feature space can include a diverse set of features capturing the relative importance of the pixels contained by the crop region. In particular, any given method for determining pixel-level saliency can be utilized. Other methods might capture the proximity of the pixel to the center of the image or its proximity to interest points or face regions. Each of these pixel-level scores can effectively be treated as an energy map. Any candidate crop region can be converted into a feature space by taking statistics of the relative amounts of energy that it captures.

We can finally learn the model by extracting the pixel-level features for each of a set of training images. We can extract crop regions by iteratively sweeping across all possible locations and sizes of possible crops (or via some random sampling of this crop space). We then know the aggregate energy features for each of these regions as well as the quality of the region as measured against a human labeler. We can then pass these examples to a Support Vector Regression Machine [6] (or any other regression model) to learn a predictive model of crop quality, given any number of pixel-level measures of importance.

Once the model has been learned, new unseen test images can be automatically cropped by extracting the pixel-level features and then sweeping across the image to exhaustively search for various candidate crop regions. Each crop region can be fed through the model and scored. We can then rank the regions based on the model's score and automatically select the top-ranked region as the final crop, or pass a set of images along to a human editor as candidates from which to make the final selection.

Category	Count	Overlap	Total
Person	5076	625	5701
Location	3978	625	4603
Movie	3000	250	3250
TV Show	1708	250	1958
Sports Team	738	125	863
Album	1500	125	1625
Totals	**16000**	**2000**	**18000**

Table 1: Distribution of labeled example images across the various classes of entities.

4. EXPERIMENTS

We have implemented and evaluated our system using a set of images with human-provided ground truth and a relatively complete set of local interest map features. In this section, we will describe these data and experimental settings in depth.

4.1 Experimental Data

We conduct our experiments using a set of 16,000 images. Each image is directly associated with a specific entity of one of the following classes: persons, locations, television shows, movies, albums, and sports teams. We presented each image to human annotators, along with the name and type of the entity, and asked them to draw a square bounding box, which captures the entity and summarizes the image. In the annotation tool, the labelers are immediately shown a thumbnail, scaled to 70 pixels, square, based on the bounding box that they have provided. The bounding box stays in place over the full image and the annotators may refine the locations of the bounding box corners and view the preview until they are sufficiently satisfied with their labeled region.

We present 12.5% of the images to two separate editors to yield 2000 images, which have two different annotations. The redundant bounding boxes can further be used to assess the degree to which the cropping process can be reliably reproduced by different human labelers.

An overview of the data is shown in Table 1. The images are selected from a large repository where the association between the entity and the image content is known. The relative frequencies of each class reflect the anticipated utility of these categories in refining Web search results.

RIGHTSLINK

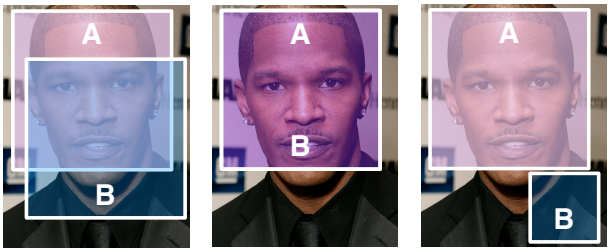


Figure 3: Demonstration of $S(A, B)$ values in various scenarios: (a) slight overlap, $S(A, B) = 0.5$, (b) complete overlap, $S(A, B) = 1$, (c) no overlap, $S(A, B) = 0$.

4.2 Evaluation Metric

To assess the relative similarity between any two given thumbnail bounding boxes, we propose a metric based on the extent to which two proposed bounding boxes overlap with each other. Essentially, we take the ratio of the area contained within *both* of the bounding boxes to the area contained by *either* of the bounding boxes. More formally, we express the image as a set of pixels I. Bounding box A is the subset of pixels I_A and bounding box B is the subset of pixels I_B. The similarity between two bounding boxes is then simply:

$$S_{A,B} = \frac{|I_A \bigcap I_B|}{|I_A \bigcup I_B|} \quad (1)$$

or the size of the intersection of I_A and I_B divided by the size of the union of I_A and I_B.

The values of $S_{A,B}$ lie between 0 (if the two bounding boxes are completely disjoint) and 1 (if the two are identical). A visualization of the metric is shown in Figure 3. The similarity is symmetric, so $S_{A,B}$ is equivalent to $S_{B,A}$.

In certain applications, we may take a human-provided bounding box as a target and use this similarity metric to compare various automatic approaches' ability to approximate the human labelers. In other applications, we can use human-defined bounding boxes for both A and B in order to assess inter-labeler agreement.

4.3 Features

We have developed and applied a number of features that can be measured at every point within an image in order to give a measure of the significance of each pixel. We then further aggregated these features at a region level to characterize the fitness of a proposed bounding box as a candidate cropping region. The individual features and the aggregation methods are described in this section.

4.3.1 Pixel-Level Features

We calculate pixel-level features based on the pixel's position in the image relative to a variety of points of interest. Since the scales of images and relative areas of interest can vary considerably, the distances that we measure are carefully normalized, the details of which are described below. A sampling of some of the pixel-level maps can be seen in Figure 4.

Spatial Features.

Among the most powerful (and easy-to-produce) features are spatial features, related to the distance of any given pixel to the center of the image. This feature map can be produced entirely from the width and height of the image, as given by the image metadata. We take the center of the image to be at the coordinate given by half the width and half the height. We then score each pixel in the image based on its Euclidean distance from the center. Since image scales vary, we normalize this value by the furthest distance possible (from the center to a corner), such that all values lie between 0 and 1. We then flip this feature into a positively-correlated value (such that being closer to the center yields a higher score), by subtracting the value from one. In addition to the Euclidean distance, we also calculate features similarly related to the distance from the vertical and horizontal dividing lines of the image. This yields three inter-related features expressing the spatial location of pixels.

Face Features.

We extract face regions using the detector implemented in OpenCV [2], which is based on cascaded boosted classifiers built on top of Haar-like features. The output of this detection is a set of square bounding boxes for each detected face region. A first-order feature to generate with this classification is a binary map: points are equal to 1 if they are within a face region and 0 if they are not. The face regions are tightly bound around the eyes and mouth, while we see that our annotators tend to include additional area outside this tight bounding box, such as the forehead, hair, and chin. Therefore, we adopt a more variable set of features based on a pixel's distance from the face region. The distance from the center of the region is a natural choice. We further observe that the annotators center their boxes non-uniformly around the face region, so we also independently measure the distance from each edge of the face bounding box. Once again, the scales of faces varies from image to image, so all of the above-described distances are measure in terms of the size of the face region. The distance in pixels is simply divided by the width of the face region (all face regions are square). The distances are once again inverted and normalized to fall between 0 and 1.

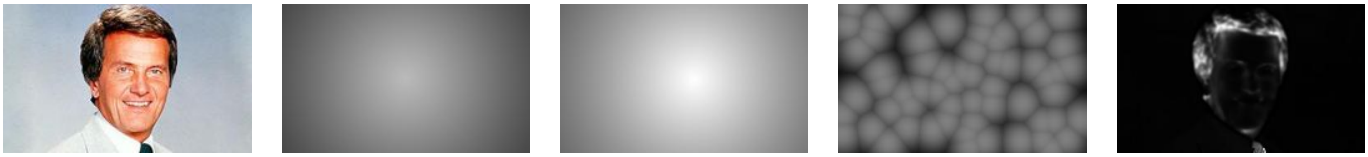
Interest Points.

We extract interest points using the difference of gaussians approach, which is commonly deployed in scale-invariant feature transform (SIFT) extraction, though in practice, any interest point detector could be used as a starting point for these features. The output from the interest point detector is essentially an identification of which points are of interest and which are not: a binary pixel map. We propagate these to all pixels in the image based on a number of factors. First, for each pixel, we measure the distance to the nearest interest point. To account for scale, distances are normalized by the longest possible distance in the image (the diagonal) to fall between 0 and 1 and then inverted. Second, for each pixel, we count the number of interest points that fall within a given radius (equal to 10% of the image's diagonal) of the pixel. This yields two interest point-related features per pixel reflecting the total distance to a point of interest and the total local density of interest.

Saliency Maps.

We extract a saliency map using a well-known technique [8], which combines multiscale image features into a single

RIGHTSLINK



(a) Image (b) Spatial (c) Face (d) Interest Points (e) Saliency

Figure 4: Examples of pixel-level features generated by various methods: (a) the original image, (b) the distance from the image center, (c) the distance from the nearest face, (d) the distance from the nearest interest point, and (e) the saliency detector.

topographical map of areas of visual interest. The saliency map technique results in a per-pixel value between 0 and 1 and we simply use this value directly without any further modification. We have chosen this approach because of its availability, but it is also plausible to include any number of more-developed saliency detection approaches that have been proposed recently in the literature [7, 13].

4.3.2 Aggregated Region Features

The above-described features all give various cues about the relative importance of each pixel in the image. In general, each of the features is designed such that we would expect a higher feature value to reflect a higher likelihood that the pixel should be included in the thumbnail. However, we require a method for characterizing the relative quality of a proposed region based on the relative amount of this feature energy that it retains. To do this, we normalize each feature map, such that the values in the map sum to unity. We then extract all the pixels that are contained within the bounding box and aggregate several different statistics over them, namely: the sum, mean, median, variance, minimum, and maximum values.

The sum of the normalized energy in the bounding box is easily understood as the proportion of the total image energy that is contained within the proposed bounding box, meanwhile the mean/median values, along with the variance and the minima and maxima, might be intuitively interpreted as reflecting how strongly concentrated the energy is within the bounding box. Since there are six total statistics to calculate, the feature space representing a bounding box is six times the number of per-pixel energy maps that we have calculated. This gives a 78-dimensional space for learning.

The interplay between these features and the trade-offs that might exist between capturing one type of importance versus another is still an open problem. To handle this, we use training examples where we know the quality of the bounding box and learn a regression model to predict the quality of new images and bounding boxes based on the statistics of the energy that the boxes contains. This is described in greater depth in the next section.

4.4 Learning

To learn the mapping between aggregate features and bounding box quality we take our set of ground truth images and bounding boxes and generate candidate bounding boxes by sweeping across the image and sub-selecting the image at various locations and scales. In general, we can define a candidate bounding box based on the location of its top-left corner in the image and the size of the bounding box. In our system, the bounding box is constrained to be square, but any pre-defined aspect ratio could be used. For any proposed bounding box, we can calculate its quality against the hand-defined boundaries that we have using Equation 1 and we can also extract the aggregate features described in Section 4.3.2. From this collection of data, we can learn the mapping from features to quality using any given regression method. We elect to evaluate Support Vector Regression (SVR) Machines [6] as implemented in the LibSVM toolkit [3].

To do this, we divide our available data into some training and testing sets. This division is conducted along the lines of which images have been annotated only once and which have been annotated by two labelers. We train on the singly-labeled images, where there is no ambiguity about the label, since only one is provided. We test on doubly-labeled images, randomly selecting one of the human-provided labels as ground truth. This allows us to obtain an upper bound on human performance by evaluating the additional human-provided labels on the test set against the selected ground-truth one. This has the benefit of being an apples-to-apples numerical comparison for any method that we happen to be evaluating. The training set is further sub-divided into 67% for actual learning and 33% for validating and selecting parameters for the SVR models. The optimal parameters are found by conducting a grid search across a range of parameter settings and selecting the set of parameters that provide the best performance on the validation data. The data is further column-normalized using score normalization to force the sample mean to zero and the sample standard deviation to unity.

This continuous sub-selection of training, development, and test sets depletes the total number of images that we have for building our models; however, the true number of training samples is further based on the number of bounding boxes that we extract from each image. Even with a very modest sweep over bounding box size parameters, we will have on the order of hundreds of sample bounding regions per training image, yielding plenty of data for learning.

We extract training bounding boxes from the images by placing the top-left corner of the box at increments vertically and horizontally away from the image origin. At each location we also increment the size of the bounding box. We conduct a complete sweep of the image and retain any candidate bounding box where all points within the box are also within the bounds of the image itself. In our experiments, we set the increment to be 10% of the height or width of the image, whichever is smaller. This builds some error into the process, since we are not guaranteed to find an optimal match in this sweep, but the maximal deviation will be small and the improvements in speed may justify the trade-

RIGHTSLINK

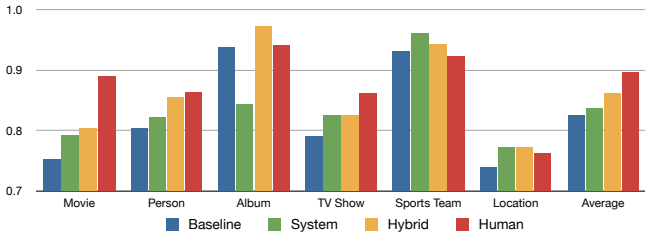


Figure 5: The performance of the the proposed system compared against the baseline "center fit" approach and inter-human agreement.

off. The increase in accuracy and decrease in speed may be adjusted via the size of this increment by the system designer. In training, we randomly select 30 boxes from each training image. This decreases learning time significantly by reducing the number of examples. We did not see significant changes in performance when increasing the sampling beyond this level. In testing, we evaluate and score *every* candidate bounding box and then select the highest-scoring box as the one to be returned by the system.

4.5 Baselines

We identify a few baselines against which to judge our system. Here we will define those methods, which we will reference in our evaluations. First, we implement a method that we call **center fit**, which takes the largest square that will fit in the center of the image: equally removing top and bottom pixels in the case of portrait-oriented images or left and right pixels in the case of landscape-oriented ones. We also identify a **human** method, which gives an upper bound of performance, by measuring the bounding box provided by one human labeler against a ground-truth one provided by a second human. This gives a best-case performance, which has no automated component.

5. EVALUATION

We evaluate our system along a number of dimensions, including the quality of the thumbnails generated, the relative importance of different features, and the value of generating different models for different entity types. These results are presented in the following section.

5.1 Thumbnail Quality

The most important aspect of the system to evaluate is the quality of the thumbnails that are being generated. We have learned and applied models across each of the entity types in our dataset along with a number of baselines. The results are shown in Figure 5. In general, the models provide a slight, but consistent, improvement over the baseline center fit approach. Some exceptions to this improvement are the cases of the "sports team" and "album" classes of images. Here, the baseline method is actually on par with human labelers and it is difficult for the content-aware system to improve much here. We further observe that these two classes are dominated by images that are square in shape: the source image has the same height/width ratio as the target thumbnail and the center fit baseline is recommending to keep the entire image, which is consistent with the labels provided by humans. Therefore, we propose a "hybrid" approach between our content-aware model and the baseline model. If

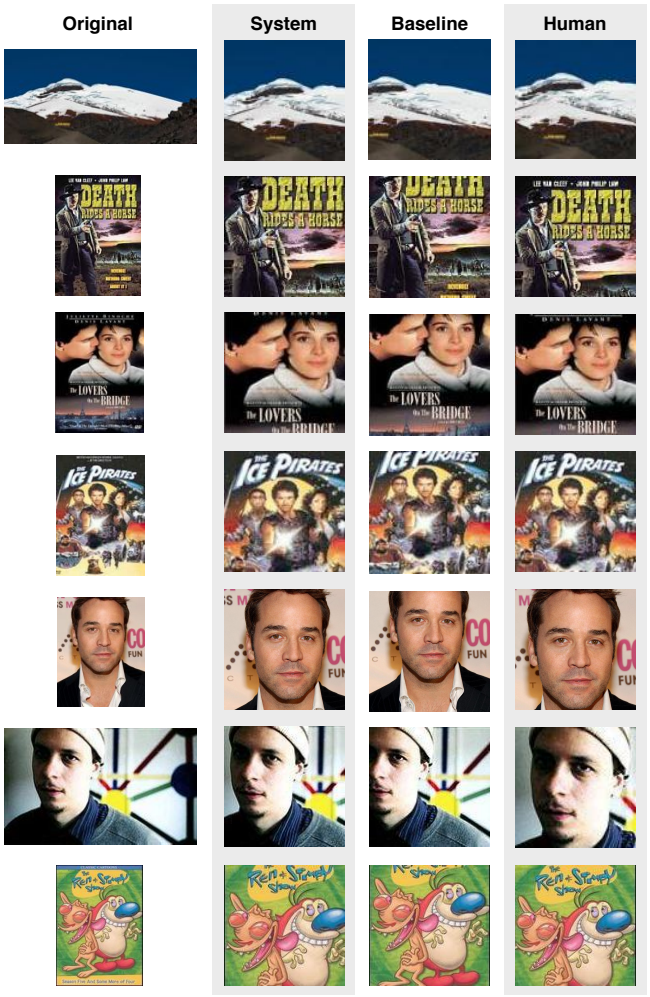


Figure 6: Example results of image crops generated by the system compared against the baseline system and crops provided by human labelers.

the image is square, we use the baseline center fit approach, otherwise, we apply the visual model. In this case, we see a much larger improvement over the baseline. This implies that our model is strong in cases where the source image is dimensionally very different from the target crop and weaker when the source image already has the correct dimensions. The model helps more significantly when the input image dimensions make the cropping task more difficult. We will revisit this point in more depth in the following section.

In Figure 6, we show some examples of the cropped images generated by our proposed system. In general, the system captures more central regions of the image. The saliency or interest features seem to capture text regions and prevent them from being cut out. Similarly, face regions are detected and prevented from being partially omitted. In the case of portrait-oriented shots of people, the system captures the entire face region in the crop with a bias towards including some space below the chin, much like the human labelers. In the case of landscape-oriented photos of people, the system is able to select a crop offset from the center of the image enough to capture the whole face.

RIGHTSLINK

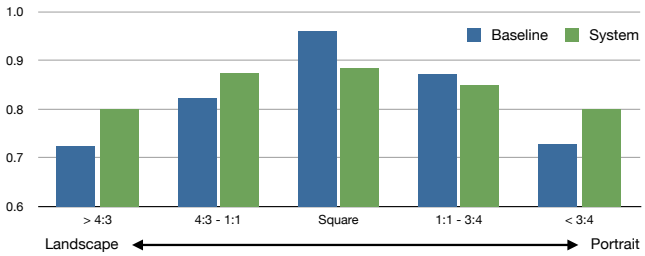


Figure 7: The performance of the the proposed system compared against the baseline "center fit" approach shown against the aspect ratio of the input image. The baseline performs higher for square images, but the model improves as the aspect ratio gets comparatively taller or wider. The x-axis shows aspect ratios in terms of width:height.

5.2 Aspect Ratio

The above results indicated that the initial aspect ratio of the original image may have implications on the efficacy of the automatic cropping techniques that are applied. For example, if the image is already the same aspect ratio as the desired thumbnail, then no cropping may need to be applied and any image analysis may be superfluous. In Figure 7, we have shown the relative performances of the baseline and system cropping methods as a function of the aspect ratio of the original input image. Starting in the center of the figure, we show the performance on square images, where the baseline approach performs very well, since we really require no cropping in these cases. As we move left from the center, the aspect ratios become exceedingly landscape-oriented and the relative improvement from our proposed model also grows. Similarly, as we move right away from the center, the images become increasingly portrait-oriented and we observe a similar growth in improvement when comparing the automatic model against the baseline approach.

5.3 Cue Importance

A key feature of our system is its ability to integrate a diverse set of cues for considering the relative importance of image regions. We wish to further evaluate the contributions of each feature type. To do this, we have grouped the features into four groups: spatial, face regions, interest points, and saliency maps, as enumerated in Section 4.3.1. We then use a leave-one-out approach where we build models wherein we omit each feature class from the used feature set and evaluate the relative drop in performance caused by the omission of each feature set.

Interestingly, in general, we find that the removal of one set of features does not measurably impact the quality of the classifier learned. This is likely due to the fact that the features exhibit a great deal of redundancy at capturing various types of interesting regions. For example, dropping the features that are related to an image's proximity to interest points might not be so detrimental since the saliency map features largely cover the same types of features. Similarly, the saliency maps and interest points might capture the general locus of faces, which dampens the effect of removing face features all together. Even the spatial features can be removed with little detriment. This is most likely because the other features still capture some information about the size or location of the proposed crop, since the features capture the amount of total image energy that is encapsulated by the crop.

Person	Location	Movie	TV Show	Sports	Album
+2.3%	0%	+1%	+1.1%	0%	0%

Table 2: Relative differences in performance when using a general model, compared to earlier results with class-specific models. The performance always increases or stays the same.

A notable exception is the case of "movie" images, wherein we find that the removal of spatial features causes a significant dip in crop performance. The movie images are primarily comprised of posters for the films, which have a predictable portrait orientation. They also have a predictable format of having a central scene and text at either the top or the bottom of the image. The spatial features can ensure that the system selects a crop that captures the horizontal center of the image and either the top or bottom of the poster (and rarely the exact center).

5.4 Class Dependency

In the above-described evaluations, we have generated different models for each of the various types of entities; however, we would also like to evaluate the performance of a more general model, which does not consider the type of entity represented by the image. To construct this model, we simply compile all of the training examples available across all entities and build a single model. We then apply the model to the same test examples used in previous evaluations. Table 2 shows the results of this experiment in terms of the relative change in performance when using the general model over the type-specific one. Interestingly, we find that the general model out-performs the class-dependent models slightly in most cases (at worst, the performance stays the same). This is despite the fact that availability of training data is heavily skewed towards a few of the classes, which would lead us to expect drops in performance for undersampled classes. This result implies that there is a degree of commonality across the feature classes and that an increase in data may trump an increase in specificity (in terms of class) when constructing these models. An alternate explanation might be that the undersampled cases (such as sports team) are comparatively easier to learn when compared to classes that feature faces and other areas of interest. In either case, computational complexity can still be a motivating factor in favor of using class-dependent models. The general case yields a very large model (with many support vectors), which leads to much slower classification at test time. The class-dependent models are lighter weight and considerably faster.

To further explore the interplay between cropping tasks for various types of images, we take the single-class models that we have learned and try to apply them to crop classification tasks on other classes of images. Table 3 shows the relative performance drops per class by using a model trained on some other class relative to the correct model. For example, the first column shows the results of applying a model trained on "person" images to each of the other classes. We see that "movie," "TV show," and "album" images all sustain only minor losses in performance when using the "per-

RIGHTSLINK

	P	L	M	T	S	A
Person	-	25	21	42	99	36
Location	22	-	23	32	99	30
Movie	11	43	-	79	91	24
TV Show	5	1	22	-	98	24
Sports Team	49	49	49	49	-	0
Album	2	20	48	20	92	-

Table 3: Relative drops in performance (expressed as percentages) when applying a model trained on one class to a cropping another class of images. For example, the first column shows the application of a model trained on the "person" class towards cropping each other class of images: applying the "person" model to the "movie" images results in an 11% drop relative to applying the "movie" model to the same task.

son" model in place of their own respective models. This makes some sense, since each of these classes likely features faces and other person-centric areas of interest, while "sports teams" and "locations" do not. In contrast, the "sports team" model applied to any other class results in very poor performance, which is likely due to the fact that the images are quite different from the other classes and the fact that there are very few training images.

6. CONCLUSIONS AND FUTURE WORK

We have presented an automatic approach towards identifying and extracting content-aware cropped thumbnails from images. Our method applies a machine-learning framework to learn the relative importance of various pixel-level measures of local saliency, so the final automatically-selected crop may consider diverse cues such as the adjacency to the image center, face regions, interest points or any other measure of saliency. To our knowledge this is the first work to be driven by ground-truth human-provided crop data. We find that this approach is close to optimal performance, when compared against human labelers.

The proposed solution requires an exhaustive sweep over each image to be cropped, which includes both the position and size of the candidate crop regions. Future iterations might explore the possibility of conducting a smart region selection either by dynamic programming or a grid-oriented sweep, cutting down the total number of regions to be considered, without significantly degrading the accuracy of region selection. The system is also open to the inclusion of any other cue that might be related to region significance, such as the detection of text regions, corners, or edges. It may also incorporate the more sophisticated saliency maps that are being developed in recent work [7, 13].

7. REFERENCES

[1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics*, 26(3):10, 2007.

[2] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] L. Chen, X. Xie, X. Fan, W. Ma, H. Zhang, and H. Zhou. A visual attention model for adapting images on small displays. *Multimedia systems*, 9(4):353–364, 2003.

[5] G. Ciocca, C. Cusano, F. Gasparini, and R. Schettini. Self-adaptive image cropping for small displays. *Consumer Electronics, IEEE Transactions on*, 53(4):1622–1627, 2008.

[6] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in neural information processing systems*, pages 155–161, 1997.

[7] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2376–2383. IEEE, 2010.

[8] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1254–1259, 2002.

[9] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2232–2239. IEEE, 2009.

[10] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch. Automatic image retargeting. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, pages 59–68. ACM, 2005.

[11] F. Stentiford. Attention based auto image cropping. In *The 5th International Conference on Computer Vision Systems, Bielefeld*. Citeseer, 2007.

[12] B. Suh, H. Ling, B. Bederson, and D. Jacobs. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 95–104. ACM, 2003.

[13] W. Wang, Y. Wang, Q. Huang, and W. Gao. Measuring Visual Saliency by Site Entropy Rate. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2368–2375. IEEE, 2010.

[14] M. Zhang, L. Zhang, Y. Sun, L. Feng, and W. Ma. Auto cropping for digital photographs. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, page 4. IEEE, 2005.

RIGHTSLINK