Classification of photographic images based on perceived aesthetic quality

Jeff Hwang JHWANG89@STANFORD.EDU
Department of Electrical Engineering, Stanford University

Sean Shi SSHI11@STANFORD.EDU
Department of Electrical Engineering, Stanford University

Abstract

In this paper, we explore automated aesthetic evaluation of photographs using machine learning and image processing techniques. We theorize that the spatial distribution of certain visual elements within an image correlates with its aesthetic quality. To this end, we present a novel approach wherein we model each photograph as a set of image tiles, extract visual features from each tile, and train a classifier on the resulting features along with the images' aesthetics ratings. Our model achieves a 10-fold cross-validation classification success rate of 85.03%, corroborating the efficacy of our methodology and therefore showing promise for future development.

1. Introduction

Aesthetics in photography are highly subjective. The average individual may judge the quality of a photograph simply by gut feeling; in contrast, a photographer might evaluate a photograph he or she captures vis-a-vis technical criteria such as composition, contrast, and sharpness. Towards fulfilling these criteria, photographers follow many rules of thumb. The actual and relative visual impact of doing so for the general public, however, remains unclear.

In our project, we show that the existence, arrangement, and combination of certain visual characteristics does indeed make an image more aesthetically-pleasing in general. To understand why this may be true, consider the two images shown in Figure 1. Although both are of flowers, the left photograph has a significantly higher aesthetic rating than the right photograph. One might reason that this is so simply because the right photograph is blurry. This conjecture, however, is imprecise because the left photograph is, on average, blurry as well. In fact, the majority of the image is even blurrier than the right photograph. Here, it seems that the juxtaposition of sharp salient regions with blurry regions and their locations in the frame positively influence the perceived aesthetic quality of the photograph.

Figure 1. A photograph with a high aesthetic rating (left) and a photograph with a low aesthetic rating (right).

Accordingly, we begin by identifying generic visual features that we believe may affect the aesthetic quality of a photograph, such as blur and hue. We then build a learning pipeline that extracts these features from images on a per-image-tile basis and uses them along with the images' aesthetics ratings to train a classifier. In this manner, we endow the classifier with the ability to infer spatial relationships amongst features that correlate with an image's aesthetics.

The potential impact of building a system to solve this problem is broad. For example, by implementing such a system, websites with community-sourced images can programmatically filter out bad images to maintain the desired quality of content. Cameras can provide real-time visual feedback to help users improve their photographic skills. Moreover, from a cognitive standpoint, solving this problem may lend interesting insight towards how humans perceive beauty.

2. Related work

There have been several efforts to tackle this problem from different angles within the past decade. Pogačnik et al [1] believed that the features depended heavily on identification of the subject of the photograph. Datta et al [2] evaluated the performance of different machine learning models (support vector machines, decision trees) on the problem. Ke et al [3] focused on extracting perceptual factors important to professional photographers, such as color, noise, blur, and spatial distribution of edges.

Also, in contrast to our approach, it is interesting to note that these studies have focused on extracting global features that attempt to capture prior beliefs on the spatiality of visual elements within high-quality images. For example, Datta et al attempted to model rule-of-thirds composition by computing the average hue, saturation, and luminance of the inner thirds rectangle of the image, and Pogačnik et al defined features that assessed adherence to a multitude of compositional rules as well as the positioning of the subject relative to the image's frame.

3. Dataset

Our learning pipeline downloads images and their average aesthetic ratings from two separate datasets.

The first is an image database hosted by `photo.net`, a photo sharing website for photographers. The index file we use to locate images was generated by Datta et al. Members of `photo.net` can upload and critique each others photographs and rate each photograph with a number between 1 and 7, with 7 being the best possible rating. Due to the wide range and subjectivity of ratings, we choose to only use photographs with ratings above 6 or below 4.2, which yields a dataset containing 1700 images split evenly between positive labels and negative labels.

The second comprises images scraped from DPChallenge, another photo sharing website that allows members to rate community-uploaded images on a scale of 1 to 10. The index file we used to locate images was generated by Murray et al [4]. Following guidelines from prior work, we choose to use photographs with ratings above 7.2 or below 3.4, resulting in a dataset containing 3000 images split evenly between positive and negative labels.

4. Feature extraction

Prior to extracting features, we partition each image into equally-sized tiles (Figure 2). By extracting features on a per-tile basis, the learning algorithm can identify regions of interest and infer relationships between feature-tile pairs that indicate aesthetic quality. For example, in the case of the image depicted in Figure 2, we surmise that the learning algorithm would be able to discern the well-composed framing of the pier from the features extracted from its containing tiles with respect to those extracted from the surrounding tiles.

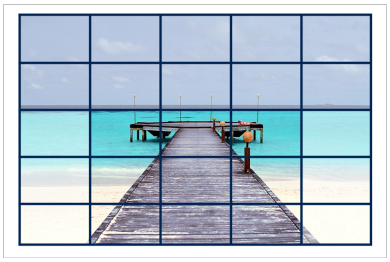

Figure 2. Tiling scheme applied to image by learning pipeline.

Below, we describe the features we extract from each image tile.

Subject detection: Strong edges distinguish the image's subject from its background. To quantify the degree of subject-background separation, we apply a Sobel filter to each image tile, binarize the result via Otsu's method, and compute the proportion of pixels in the tile that are edge pixels:

$$f_{sd} = \frac{\sum_{(x,y)\in \text{Tile}} 1\{I(x,y) = \text{Thresholded edge}\}}{|\{(x,y)|(x,y) \in \text{Tile}\}|}$$

Color: A photograph's color composition can dramatically influence how a person perceives a photograph. We capture the color diversity within an image tile using a color histogram that subdivides the three dimensional RGB color space into 64 equally sized bins. Since each pixel can take on one of 256 discrete values in each color channel, this results in each bin being a cube with 16 possible values in each dimension. We normalize each bin's count by the total pixel count so that it is invariant to image dimensions.

We also measure the average saturation and luminance of each tile's pixels. Finally, for the entire image, we compute the proportion of pixels that correspond to a particular hue (red, yellow, green, blue, and purple).

Detail: Higher levels of detail are generally desirable for photographs, particularly for its subject. To approximate the amount of detail, we compare the number of edge pixels of a Gaussian filtered version of the

image tile to the number of edge pixels in the original image tile, i.e.

$$f_d = \frac{\sum_{(x,y)\in \text{Tile}} 1\{I_{filtered}(x,y) = \text{Edge}\}}{\sum_{(x,y)\in \text{Tile}} 1\{I(x,y) = \text{Edge}\}}$$

For an image tile that is exceptionally detailed, many of the higher-frequency edges in the region would be removed by the Gaussian filter. Consequently, we would expect f_d to be closer to 0. Conversely, for a tile that lacks detail, since few edges exist in the region, applying the Gaussian filter would impart little change to the number of edges. In this case, we would expect f_d to be closer to 1.

Contrast: Contrast is the difference in color or brightness amongst regions in an image. Generally, the higher the contrast, the more distinguishable objects are from one another. We approximate the contrast within each image tile by calculating the standard deviation of the grayscale intensities.

Blur: Depending on the image region, blurriness may or may not be desirable. Poor technique or camera shake tends to yield images that are blurry across the entire frame, which is generally undesirable. On the other hand, low depth-of-field images with blurred out-of-focus highlights ("bokeh") that complement sharp subjects are often regarded as being pleasing.

To efficiently estimate the amount of blur within an image, we calculate the variance of the Laplacian of the image. Low variance corresponds to blurrier images, and high variance to sharper images.

Noise: The desirability of visual noise is contextual. For most modern images and for images that convey positive emotions, noise is generally undesirable. For images that convey negative semantics, however, noise may be desirable to accentuate their visual impact. We measure noise by calculating the image's entropy.

Saliency: The saliency of the subject within a photograph can have a significant impact on the perceived aesthetic quality of the photograph. We post-process each image to separate the salient region from the background using a center-vs-surround approach described in Achanta et al [5]. We then sum the number of salient pixels per image tile and normalize by the tile size.

5. Methods

Figure 3 depicts a high-level block diagram of the learning pipeline we built. The pipeline comprises three main components: an image scraper, a bank of feature extractors, and a learning algorithm.

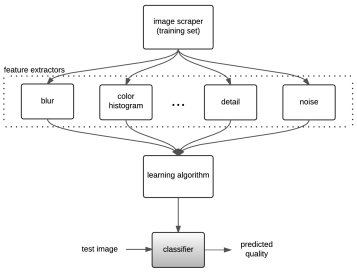

Figure 3. Block diagram of learning pipeline.

For each of the features we identified, there exists a feature extractor function that accepts an image as an input, calculates the feature value, and inserts the feature-value mapping into a sparse feature vector allocated for the image. We rely on image processing algorithms implemented in the `scikit-image` and `opencv` libraries for many of these functions [6, 7].

After the pipeline generates feature vectors for all images in the training set, it uses them to train a classifier. For the learning algorithm, we experimented with `scikit-learn`'s implementations of support vector machines (SVM), random forests (RF), and gradient tree boosting (GBRT) [8]. We focus our attention on these algorithms because they can account for non-linear relationships amongst features.

SVM: The SVM learning algorithm with ℓ_1 regularization involves solving the primal optimization problem

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i,\ i = 1, ..., m \end{aligned}$$

, the dual of which is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C,\ i = 1, ..., m \\ & \sum_{i=1}^{m}\alpha_i y^{(i)} = 0 \end{aligned}$$

Accordingly, provided that we find the values of α that maximize the dual optimization problem, the hypothesis can be formulated as

$$h(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{m}\alpha_i y^{(i)}\langle x^{(i)}, x\rangle + b \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Note that since the dual optimization problem and hypothesis can be expressed as inner products between input feature vectors, we can replace each inner product with a kernel applied to the two input

vectors, which allows us to train our classifier and perform classification in a higher-dimensional feature space. This characteristic of SVMs makes them well-suited for our problem since we speculate that non-linear relationships amongst features influence image aesthetic quality. For our system, we choose to use the Gaussian kernel $K(x, y) = \exp\left(\gamma||x - y||_2^2\right)$, which corresponds to an infinite-dimensional feature mapping.

Random forest: Random forests comprise collections of decision trees. Each decision tree is grown by selecting a random subset of input variables to use for splitting at a particular node. Prediction then involves taking the average of the predictions of all the constituent trees:

$$h(x) = \text{sign}\left(\frac{1}{m}\sum_{i=1}^{m} T_i(x)\right)$$

Because of the way each decision tree is constructed, the variance of the average prediction is less than that of any individual prediction. It is this characteristic that makes random forests more resistant to overfitting than decision trees, and, thus, generally have much higher performance.

Gradient tree boosting: Boosting is a powerful learning method that sequentially applies weak classification algorithms to reweighted versions of the training data, with the reweighting done in such a way that, between every pair of classifiers in the sequence, the examples that were misclassified by the previous classifier are weighted higher for the next classifier. In this manner, each subsequent classifier in the ensemble is forced to concentrate on correctly classifying the examples that were previously misclassified.

In gradient tree boosting, or gradient-boosted regression trees (GBRT), the weak classifiers are decision trees. After fitting the trees, the predictions from all the decision trees are weighted and combined to form the final prediction:

$$h(x) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i T_i(x)\right)$$

In literature, tree boosting has been identified as being one of the best learning algorithms available [9].

6. Experimental results and analysis

For each learning algorithm, we measure the performance of our classifier using 10-fold cross validation on the `photo.net` dataset and the DPChallenge dataset. We run backward feature selection to eliminate ineffective features to improve classification performance. For the final set of features, we remove each feature from the set and run cross-validation on the resulting set to verify that the feature contributes positively to the classifier's performance.

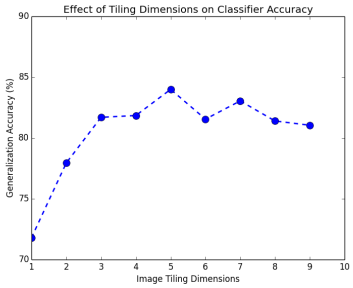

Figure 4. Classifier 10-fold cross-validation accuracy versus tiling dimension (SVM with $C = 1$ and $\gamma = 0.1$, DPChallenge dataset).

In experimenting with different tiling dimensions, we found that dividing each image into five-by-five tiles gave the best performance (Figure 4). A tiling dimension of 1 corresponds to extracting each feature across the entire frame of the image. As anticipated, this yields significantly worse performance. Larger tiling dimensions should theoretically work better provided that we have enough data to support the associated increase in the number of features. Unfortunately, given the limited sizes of our datasets, the addition of more features causes our classifier to overfit and thus degrades its accuracy for dimensions larger than 5.

For SVM, we tuned our parameters using grid search, which ultimately led us to use $C = 1$ and $\gamma = 0.1$. For random forest, we used 300 decision trees. We determined this value by empirically finding the asymptotic limit to the generalization error with respect to the number of decision trees used. For gradient tree boosting, we used 500 decision trees and a sub-sampling coefficient of 0.9. Using a sub-sampling coefficient smaller than 1 allows us to trade off variance for bias, which thereby mitigates overfitting and hence improves generalization performance.

Table 1 shows our 10-fold cross-validation accuracy for each learning algorithm. For both datasets, we got the highest performance with GBRT, with accuracies of 80.88% and 85.03%. The difference in performance may have resulted from the DPChallenge dataset's having higher-resolution images than the `photo.net` dataset, which makes certain visual features more distinct in the former than the latter. Nonetheless, the similarity in results suggests that our methodology generalizes well to different datasets.

Figure 5 shows the confusion matrix for 10-fold cross-

	SVM	RF	GBRT
`photo.net`	78.71%	78.58%	80.88%
DPChallenge	84.00%	83.15%	85.03%

Table 1. 10-fold cross-validation accuracy

		Predicted label 1	0
Actual label	**1**	TP 85.80%	FN 14.20%
	0	FP 15.73%	TN 84.27%

Figure 5. Confusion matrix for 10-fold cross validation with GBRT on DPChallenge dataset.

validation using GBRT on the DPChallenge dataset. The true positive and false negative rates are approximately symmetric with the true negative and false positive rates, respectively, which signifies that our classifier is not biased towards predicting a certain class. This also holds true for the `photo.net` dataset.

To analyze the shortcomings of our approach, we examine images that our classifier misclassified.

Figure 6 shows an example of a negative image from the `photo.net` dataset that the classifier mispredicted as being positive. Note that the image is compositionally sound – the subject is clearly distinguishable from the background, fills most of the frame, is well-balanced in the frame, and has components that lie along the rule-of-thirds axes. The hot-pink background, however, is jarring, and the subject matter is mundane and lacks significance. Unfortunately, because it discretizes color features so coarsely, the classifier is likely not able to effectively differentiate between the artificial pink shade of the image's background and the warm red shade of a sunset, for instance. Moreover, it has no way of gleaning meaning from images. We therefore believe that it is primarily due to these shortcomings that our classifier misclassified this particular image.

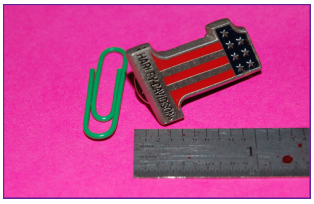

Figure 6. Negative image classified as positive by the model.

Figure 7. Positive images classified as negative by the model.

Figure 7 shows two photographs from the DPChallenge dataset that our classifier misclassified as being negative. While the left photograph follows good composition techniques, the subject has few high frequency edges, so the classifier would likely need to rely more on saliency detection to pinpoint the subject. Unfortunately, the current method of detecting the salient region is not consistently reliable, so despite this photograph's having a distinct salient region, the classifier may deemphasize the contributions of this feature. We believe that improving our salient region detection accuracy across all images may enable the classifier to utilize the saliency feature more effectively, and thus correctly classify this photograph. In the right image in Figure 7, the key visual element is the strong leading lines that draw attention to the hiker – the subject of the image. Leading lines, however, are global features that are not well-captured by our tiling methodology, and, thus, are likely not considered by the classifier.

In sum, although our system performs respectably, examining the images it mispredicts reveals many potential areas of improvement.

7. Future work and conclusions

We have demonstrated that modeling an image as a set of tiles, extracting certain visual features from each tile, and training a learning algorithm to infer relationships between tiles yields a high-performing photographic aesthetics classification system that adapts well to different image datasets. Thus, our work lays a sound foundation for future development. In particular, we believe we can further improve the accuracy of our system by deriving global visual features and parsing semantics from photographs. Our model should also apply to regression for use cases where numerical ratings are desired. Finally, augmenting the system with the ability to choose a classifier depending on the identified mode of a photograph, e.g. portrait or landscape, may lead to more accurate classification of aesthetic quality.

References

[1] Pogačnik, D., Ravnik, R., Bovcon, N., & Solina, F. (2012). Evaluating photo aesthetics using machine learning.

[2] Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2006). Studying aesthetics in photographic images using a computational approach. In *Computer Vision–ECCV 2006* (pp. 288-301). Springer Berlin Heidelberg.

[3] Ke, Y., Tang, X., & Jing, F. (2006, June). The design of high-level features for photo quality assessment. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (Vol. 1, pp. 419-426). IEEE.

[4] Murray, N., Marchesotti, L., & Perronnin, F. (2012, June). AVA: A large-scale database for aesthetic visual analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (pp. 2408-2415). IEEE.

[5] Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009, June). Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 1597-1604). IEEE.

[6] van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... & Yu, T. the scikit-image contributors,(2014) Scikit-image: image processing in Python. Peer J, 2(6).

[7] Bradski, G. (2000). The opencv library. *Doctor Dobbs Journal*, 25(11), 120-126.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825-2830.

[9] Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer, Berlin: Springer series in statistics.