Human and Machine Detection of Stylistic Similarity in Art

Adriana Kovashka
Department of Computer Science
University of Texas at Austin
Austin, TX 78712
adriana@cs.utexas.edu

Matthew Lease
School of Information
University of Texas at Austin
Austin, TX 78701-1213
ml@ischool.utexas.edu

ABSTRACT

We describe methodology and evaluation for a new *find-similar* search task: the user specifies a source painting and seeks other *stylistically* similar paintings, regardless of the source painting's subject (i.e. the object, person, or scene depicted). We formulate this search as a content-based image retrieval task, modeling stylistic similarity via detected color, intensity in color changes, texture, and sharp points. Additional features from machine vision are used for local patches and the overall scene. To evaluate both the task difficulty and system effectiveness, 90 people with varying knowledge of art were asked to judge stylistic similarity between different pairings of 240 paintings. To obtain these judgments, we utilized Amazon Mechanical Turk, and we discuss design issues involved in working with the platform and controlling for quality in a crowdsourced setting. Results of 3128 judgments show both task difficulty, with approximately 50% to 76.5% agreement between judges, and a range of accuracies of system features vs. human judgments. Most promising, features based on Leung-Malik filters [10] achieve roughly 80% agreement with knowledgeable judges.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.5.2 [**User Interfaces**]: Theory and methods; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms

Algorithms, Experimentation, Human Factors

Keywords

search, image analysis, crowdsourcing, Mechanical Turk

1. INTRODUCTION

Finding similar items is a natural human activity and one for which automated systems are commonly employed in practice when the number of items being searched is large [17]. In computer vision, applications such as surveillance and content-based search of objects or faces require modeling similarity of photographs or video frames. In this paper, we consider a potentially more challenging task: finding similar paintings on the basis of stylistic similarity. One potential application of such automation would be as part of a content-based art recommendation search engine in which the user might identify a painting she liked in order to receive recommendations of other paintings she might enjoy. Note that we are interested in evaluating similarity without explicit knowledge regarding the author of the painting.

While machine vision techniques for analyzing photographs and video have received significant attention, relatively little work has investigated analysis and feature design of artistic imagery. There has been some work in developing descriptors for categorizing paintings according to their authorship, such as the fractals-based features for recognizing paintings by Jackson Pollock due to Irfan and Stork [8]. Vill and Sablatnig overview another type of feature for describing brush strokes [20]. Hertzmann et al. develop an algorithm for automatic learning of painting styles by example [7], and their approach uses features such as luminance and texture, but their task involves synthesis while we are concerned with analysis, so their feature mapping and search techniques are not necessarily appropriate for image retrieval. Another art-related project is Hany Farid's art forensics [5]. The most similar work we are aware of, by Li and Chen, predicted painting quality using human judgments [11].

While style is a broad concept, we model it in our system primarily via low-level features capturing color palette, intensity or smoothness of color changes across an image, appearance and granularity of brush strokes, density of sharp points and edges in a painting, and texture of the painting. We use a number of well-known descriptors and adapt others to the task at hand (§3). Once features are extracted, stylistic similarity is modeled via distance between image feature vectors under a χ^2 kernel (§4).

To evaluate both task difficulty and system effectiveness (§6), we asked people with varying knowledge of art to judge the stylistic similarity between various pairs of paintings. We crowdsourced this similarity judging task as a Human Intelligence Task (HIT) in Mechanical Turk (MTurk)[1], as further discussed in §5. While platforms like MTurk reduce many technological barriers to crowdsourcing, a variety of practical challenges remain which can limit the practical effectiveness of the crowdsourcing paradigm [4, 14, 19]. Following community recommendations for effective HIT design, 90 human judges with varying knowledge of art completed 3128 similarity judgments at a total cost of $12.09. Despite several measures taken to provide quality control, inter-annotator agreement was roughly 50% to 76.5%, suggesting the inherent difficulty of the task even for people.

CrowdConf 2010, October 4, 2010, San Francisco, CA.

[1] https://www.mturk.com

System features based on Leung-Malik filters [10] showed highest predictive accuracy, achieving roughly 80% agreement with knowledgeable human judges.

2. DATA

We collected a dataset of paintings by 6 authors and 40 images per author, for a total of 240 images. The authors and their artistic styles are listed in Table 1. Sample paintings from each author are shown in Figure 1. The images were downloaded from Wikimedia Commons [2].

Painter	Primary Style
Francisco Goya	Romanticism
Ernst Kirchner	Expressionism
Gustav Klimt	Symbolism, Secession
Franz Marc	Expressionism
Claude Monet	Impressionism
Vincent van Gogh	Post-Impressionism

Table 1: Authors in our dataset and their styles, according to Wikimedia's sister project Wikipedia.

3. FEATURES

To model painting style, we consider two broad classes of features: global and local. Global features represent the image holistically, while local features describe individual small patches of the image. Color, Phog and Gist are global features, while extracting corners and measuring the image's response to a set of filters are local approaches. However, since we wish to arrive at a representation of the painting as a whole, rather than recognize individual objects in it, we adapt the descriptors and achieve global representations. Below we describe both the method for computing each feature, as well as how it is expected to help distinguish between the artistic style in different paintings. Table 2 below lists the feature types used and their referring acronyms.

Acronym	Feature
CC	Color cells
CC-2	histogram version of Color cells
IP1	number of interest points at different scales
C	Color
P	Phog
G	Gist
IP2	strengths of interest point detections
IP2-2	histogram of strengths of interest points
F1	responses to LM filters
F1-2	histograms of responses to LM filters
F2	responses to RFS filters
F2-2	histograms of responses to RFS filters
F3	responses to S filters
F3-2	histograms of responses to S filters
All	combination of all feature types

Table 2: Feature types.

3.1 Color Features

One of the simplest descriptions of an image is the color distribution of the whole image. This description can be encoded in the form of a color histogram. To produce such a histogram, we examine the intensity of color along the different channels of some color space, such as RGB, HSV, or CIELAB, the last of which is used in our work. Next, we designate a set of "bins," each of which corresponds to some intensity range, and count how many pixels fall into each bin. The combination of the bin counts for each color channel constitutes our color histogram for the image. We call this feature type "Color."

A variation of this feature can be used to examine how color changes across an image. If color changes smoothly in the painting, it is likely that it is an impressionist painting, while if the color changes are abrupt, perhaps the image is an expressionist painting. We devise the feature type "Color cells" which consists in the following. We place an imaginary grid over the image with some user-specified size nxn, where n is the number of grid cells along each dimension. We then compute the color histogram of each grid cell individually. Next, for each pair of grid cells, we compute the Euclidean distance between their color histograms, and we concatenate all distance values. The string of values becomes a new representation of the image.

The "Color cells" feature is sensitive to orientation because it requires that regions of great color changes between two images happen at the same place. To remove this limitation, we also compute a histogram over the distance values string and use this as the new feature representation.

3.2 Phog Features

A common feature used in computer vision is Histogram of Oriented Gradients (HoG). We use this as a standard feature type, rather than one that is expected to be particularly suitable to paintings. As shown in Figure 2, a HoG feature describes the strength of the gradients in a fixed number of orientations (8 in this case), so if the histogram is computed on the whole image, the feature vector will be 8-dimensional. If histograms are computed on each cell in a 2x2 grid, the dimension of the feature vector will be 2x2x8=32.

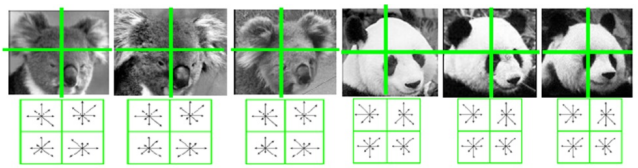

Figure 2: HoG feature computation. Image courtesy of Kristen Grauman.

We use the implementation of a version of HoG named Pyramid Histogram of Oriented Gradients (PHOG), which computes HoG at different granularities, due to Bosch and Zisserman [3]. At the base pyramid level, the HoG is computed for the whole image, but at the next level, the image is broken up into 2x2 regions and the histogram of oriented gradients is computed individually for each grid cell. At the next level, each grid cell is broken up further, and so forth. We use 4 pyramid levels in total (the default number of levels provided in the code by [3]) and concatenate the histograms for each level.

3.3 Gist Features

One popular global feature is Gist, due to Oliva and Tor-

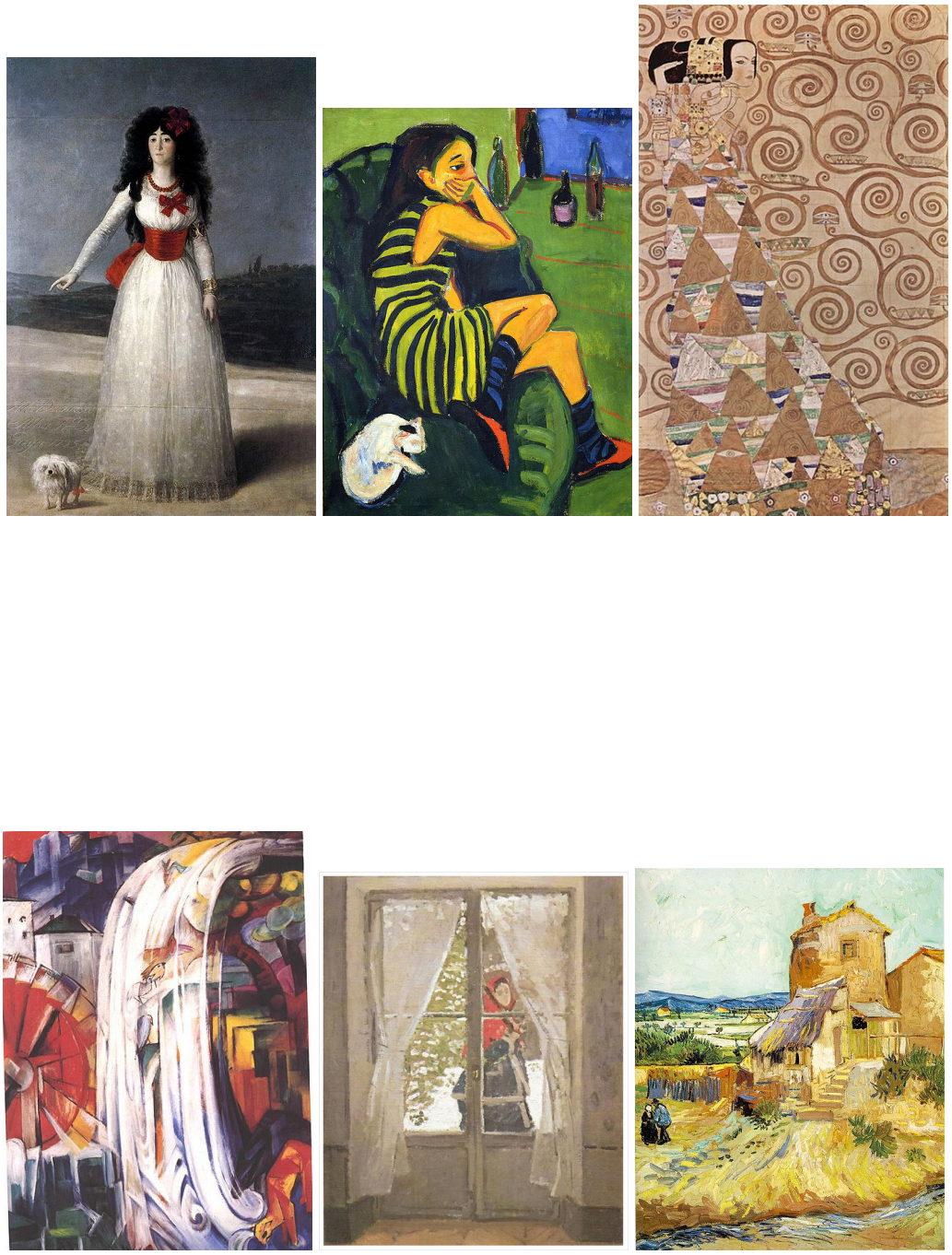

Figure 1: Sample paintings in the dataset.

ralba [13]. Gist features describe the energy spectra and the frequency of change in the images, using Fourier transforms. In [13], Gist features are used for categorizing scene types, and natural images are shown to have different spectral templates than urban images. We use this feature type because we expect that it can capture the global "feel" of a painting, which is a slightly higher-level (and more subjective) concept than the rest of the style markers we use. However, Gist was not designed specifically for paintings, so we consider it a generic feature type.

3.4 Interest Point Features

One mark of style is the smoothness of the appearance of the painting. If the strokes in the image are rough, there will be a large number of points where the edges and corners in the image appear, corresponding to locations of large intensity changes. To measure the frequency of such intensity changes, we develop interest point features.

There is a variety of interest point operators in computer vision. These operators seek parts of the image which are worth describing with local patches, if the patches were to be sparse as opposed to densely sampled. One type of interest points are corners in an image, which are locations of large intensity gradients in both the x and y dimensions, as shown in Figure 3. These corners can be detected at multiple scales. As a new representation of our image, we count how many corners were detected at each scale, and name this feature type "Interest Points 1."

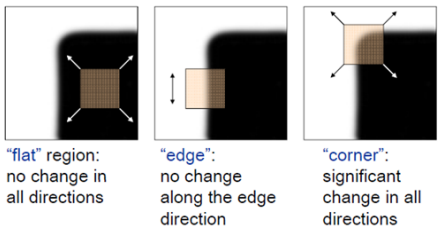

Figure 3: Corner interest points. Image courtesy of Kristen Grauman.

We can also use the strengths of the detected corners as a description of the style of the image. We concatenate the strength value for each detected corner, and this becomes the feature "Interest Points 2." Since two images can and likely will have a different number of detected corners, when comparing the feature vectors of two image, we pick a random sample of the values in each vector so that the two vectors are of the same size. Alternatively, we can compute a histogram by binning the strength values, and use the histogram for each image as its feature representation.

3.5 Filter Bank Features

Our last feature type seeks to record the type of brush strokes which appear in each painting. To do that, we compute the response of an image to a number of filters from a filter bank, using code from [1]. The response values are obtained by computing a 2-d convolution between the image and the filters. The three filter banks we use are the Leung-Malik (LM) filter bank [10], the Schmid filter bank [15], and the Maximum Response (MR, also denoted RFS) filter bank [6]. Some of these filters resemble brush strokes. The filters from the LM bank are shown in Figure 4.

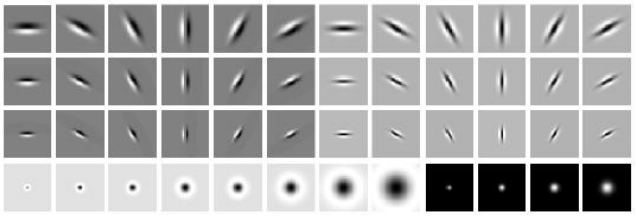

Figure 4: The Leung-Malik filter bank. Image courtesy of `http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html`.

Once a response has been computed, we can combine response values into a vector or histogram the responses. Thus for each type of filter bank we have two feature types.

4. COMPUTING SIMILARITY

For each feature type, we have a vector for each image corresponding to this image's representation according to the given feature descriptor. To compare the similarity between images, we need a way to compute a distance between them using one or more of their feature vectors. For this purpose, we compute a χ^2 *kernel* K as defined in [9]. $K(i,j)$ computes the similarity between two images i and j, with 1 corresponding to identical images and 0 to completely different images. For each feature type, there is one kernel. To combine features, we simply average their corresponding kernels; future work will investigate weighting and integrating these kernels into a combined ensemble via supervised *learning to rank* [12]. H_i and H_j below denote the histograms for images i and j, and k is the dimension of their feature vectors. m is the mean χ^2 distance across all dataset images.

$$\chi^2(H_i, H_j) = \frac{1}{2}\sum_{c=1}^{k}\left(\frac{(H_i(c) - H_j(c))^2}{H_i(c) + H_j(c)}\right) \quad (1)$$

$$K(i,j) = \exp\left(-\frac{1}{m}\chi^2(H_i, H_j)\right) \quad (2)$$

The χ^2 kernel helps us compute the distance between two images. Now we map this distance to a similarity rank between 1 and 5, with 1 corresponding to very similar paintings and 5 to very dissimilar paintings. Assume v is a vector of the distance between image I and all other images in the dataset. A high kernel value corresponds to a small distance, but we want a low similarity rank to correspond to similar images, so we map high values to low values by setting $v = 1 - v$. Now we divide v by the highest value in v, to ensure that all values in v are between 0 and 1 (which they should already be). Next, we multiply v by 5 and round to the next highest integer, to ensure that v ranges between 0 and 5. Now we map all values below 1 to 1 to obtain values between 1 and 5 for all images. This is namely the vector of similarity rank scores we will use for evaluation.

5. COLLECTING HUMAN JUDGMENTS

To evaluate both task difficulty and system effectiveness, we asked people with varying knowledge of art to judge the stylistic similarity between different pairs of paintings. We defined this similarity judging task as an MTurk HIT and crowdsourced it to distributed workers.

To determine how humans perceive the similarity between paintings, we generated sets of image pairs for a subset of all possible pairs in our dataset. For each pair, we asked the worker to rate the stylistic similarity of each pair on a 5-point scale: "very similar," "somewhat similar," "neither similar nor dissimilar," "somewhat dissimilar," or "very dissimilar". Our instructions explained what we meant by "style," as well as illustrated a dissimilar pair and a similar pair (see Table 3). An example HIT is shown in Figure 5.

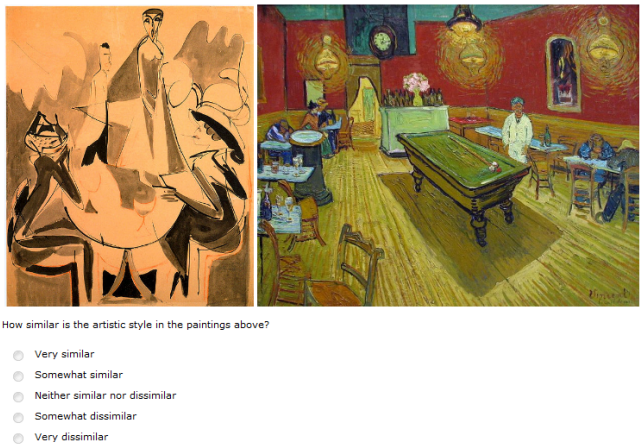

Figure 5: Interface of the judgment request for one image pair. The guidelines, feedback box, and self-reported expertise box are not shown.

While MTurk reduces many technological barriers to crowdsourcing, a variety of practical challenges remain which can limit the practical effectiveness of the crowdsourcing paradigm [4, 14, 19]. For example, we followed a principle of iterative refinement: we incrementally designed our MTurk HIT based on feedback from friends, co-workers, and small pilot runs. This let us identify and fix problems as early as possible to reduce cost and maintain a positive reputation with workers. For example, one early tester reported not knowing what was meant by "style" thus leading us to add example similar and dissimilar pairs to elucidate the desired distinction.

Amazon currently charges 10% overhead on HIT cost, with a minimum charge of $0.005 per HIT, providing some incentive to perform multiple judgments per HIT. Further incentive comes from wanting to ensure each worker performs some minimum number of HITs such that their accuracy can be assessed with some minimal confidence. We included 5 image pairs to judge per HIT.

An open question in general with crowdsourcing is how to determine appropriate pay. Issues include: difficulty of work (how long it will take), nature of the work (how fun it will be), desire to attract workers while avoiding spam workers, etc. We did not investigate this issue here; we tried the minimum rate of $0.01 per HIT and had no problem attracting workers. It is certainly possible that higher quality workers might have been attracted by greater pay.

While we did not use either a qualification test or trap questions for quality control, we did try requiring workers to provide feedback justifying their judgments. The argument for such feedback is that besides identifying problems with HIT design and providing useful feedback on the specific HIT performed, it can be a simple way to gauge user effort and seriousness via the degree and nature of the feedback provided. The concern of requiring such feedback rather than having it be optional is that it may discourage some workers who are competent to perform the task but not comfortable or willing to provide written feedback in English. Our subsequent analysis divides judgments into HIT design groups *D1* (feedback required for at least one judgment) and *D2* (no option for feedback in the HIT).

To improve quality, we collected three judgments per image pair and resolved disagreements via simple majority vote. More sophisticated strategies for label selection [16] and label aggregation [18, 21] have been left for future work.

As part of the HIT design, we asked workers to self-assess their own knowledge of art as a basis for interpreting their judgments. To encourage honesty, no suggestion was made of greater pay to more knowledgeable workers. Knowledge of art was rated on a 3-point scale: "a lot," "a little bit," or "none". Our analysis thus partitions the three judgments into three expertise categories, meaning some image pairs will have less than three judgments for a given category. When this leaves two disagreeing judgments, or when three judgments all pick different categories, we randomly pick one of the judgments. While we wanted our evaluation to include such "close-calls" (system scores should reflect these boundary cases), the inclusion of this random tie-breaking data effectively added an unhelpful white-noise signal to our system evaluation, and in hindsight it would have been better to omit it entirely. We could have also reduced cases of two-way ties by iteratively resubmitting each image pair until we had collected at least three judgments for it for each expertise category.

The number of workers who completed HITs and the number of judgments are presented in Tables 4 and 5. The number of workers overall is less than the sum of the workers for designs *D1* and *D2* since some workers completed tasks for both designs. At a cost of $0.015 per HIT, the 630 HITs cost a total of $9.45. An additional $2.64 was spent in iterating the HIT design, for a total cost of $12.09.

HIT Design	Unique Workers	HITs	Judgments
D1	74	450	2237
D2	47	180	891
Total	90	630	3128

Table 4: Worker statistics.

6. EVALUATION

We begin our evaluation by measuring task difficulty as a function of inter-annotator agreement. In particular, we report what fraction of judgments for a given image pair are equal to the majority vote for that image pair (± 1, i.e. allowing scores to be 1 off and still match). While reporting of Fleiss' kappa would have been more standard, this simple statistic was sufficient to show the agreement of ap-

Guidelines:

The goal of this task is to determine how similar the artistic style of two paintings is.

Please only examine the style of the paintings, NOT their content.

Style is a broad concept, but some examples of aspects of style involve the intensity of color changes, the appearance of brush strokes, subtlety versus visual "loudness", color palettes, realism of the images versus abstractness, etc. You also have some freedom in choosing what "style" means to you.

For example, the following two images (one of which is an impressionist painting and the other expressionist) differ in the intensity of color changes.

[displayed pair of stylistically dissimilar images]

The types of brush strokes in the images below are also different.

[displayed pair of stylistically dissimilar images]

On the other hand, the following two images are similar in style (and are in fact both expressionist paintings).

[displayed pair of stylistically similar images]

Please provide a very brief (a few words) explanation of your choice for one of the image pairs in the field at the bottom. Also please indicate how knowledgeable you are about art.

If you cannot see the images, please return the HIT. Thank you!

Table 3: HIT instructions to workers in design *D1* (explained below).

Knowledge	D1	D1 labels	D2	D2 labels
Most	522	448	175	160
Middle	1650	735	641	290
Least	65	65	75	70
Total	2237	1248	891	520

Table 5: Statistics of collected judgments and labeled image pairs as a function of HIT design vs. worker knowledge level. A "ground-truth" label for each image pair is inferred from judgments via majority vote (with random selection in case of ties).

proximately 50% to 76.5% between judges. Table 6 provides specific results of inter-annotator agreement for each of the three knowledge level worker categories.

Knowledge	D1	D2
Most	0.7655	0.6667
Middle	0.7637	0.7582
Least	N/A	0.5000

Table 6: Inter-annotator agreement statistics.

We next evaluate agreement of our features in comparison with human judgments of style similarity. We expect generic feature types (such as Phog) to perform less well than feature types more suitable for discriminating between art styles (such as the filter bank features) or specifically designed to capture style markers (such as "Color cells"). Our evaluation procedure and results are described below.

We evaluate the performance of our algorithm by comparing the human or user similarity score (U) to the system or machine score (S). As noted above, if the two scores are $\leq$ 1 ranks apart (e.g. $U = 1, S = 2$), we consider the scores to be in agreement. There are $5x5 = 25$ possible combinations of U and S scores, and 13/25 scores signify agreement within ± 1. Assuming similarity judgments are uniformly distributed across the five possible categories, chance performance is 52%. However, the ± 1 tolerance means predicting only categories 2-4 raises chance performance to 60%.

The performance score for each feature type is computed as the fraction of image pairs for which the users and the feature agree (within ± 1 categories). We use the competence scores which the workers provided to compute separate performance scores for each feature type in each competence category (Figures 6, 7 and 8). Perhaps most promising, "F1" (LM filters) achieved nearly 80% agreement with the most knowledgeable workers (Figure 6).

6.1 Worker Feedback Examples

In HIT design *D1* which required written justification of judgments made, the following examples provide a flavor for the quality of feedback provided:

Workers with significant knowledge of art:

- "good", "very nice", "not bad"
- "... they are very dissimilar because the brush strokes and use of line and color are quite different."
- "the image pair is of a painting by Goya (?) and an image by Sir Laurence Alma-Tadema. The former is probably early c.19th and is a romantic portrait,with somewhat free brushstrokes. The latter, a late c.19th historical painting, with erotic overtones. The brushwork is very smooth and facile."
- "The pair of images consists of a very freely handled, art deco-influenced , 20th century image of deer frolicking against a semi abstract background. The second image is a late c19th portrait of a lady, rather formal, Japanese-iinfluenced, much tighter and studied brushstrokes. The former painter is not interested in texture, whereas the second is."

Workers with medium knowledge of art:

- "first picture", "last pair", "HARD"
- "I said they were very disssimilar although they both have human figures, the one on the left is very realistic, and the one on the right is extremely skewed."
- "one is just a modern art, other is nature's beauty"
- "the pairs is expressing tne happyness in similarity"
- "the first painting is very harsh with definate strong lines and context. Whereas the second picture is soft and rounded both in style and colour."
- "Both Paintings are by van Gogh"

Workers with little knowledge of art:

- "good", "HAPPY"
- "number 1 - realism in the image"
- "The image 5 is somewhat dissimilar because one in of a lady and one of a scenery."
- "... the first pic is more flowing and blended; the second has harder lines and more patterns."
- "palette three proves very disimilar due to the visual loudness, the monet is very subtle in comparison to the other painting in reguards to brushstroke and colour"

6.2 Discussion

In comparison to the knowledgable workers, the filter bank features performed very well, with the LM filter bank features most closely matching human judgments. In contrast, the generic Gist and Phog features did not work well, as expected. This suggests that our special features agreed quite well with knowledgeable judges. Overall, the histogram versions of most features work as well as the non-histogram versions, except for the LM filters features and the color cells features, which is intriguing and indicates that the spatial placement of features cannot be ignored.

With the somewhat knowledgable users, all features performed roughly comparably, with the feature types designed specifically for discriminating between art styles (the filter bank and color cells features) performing slightly better than the generic features. Interestingly, we observe that very and somewhat knowledgeable workers both had similar agreement for design *D1* yet compare differently with the machine output. This merits further investigation. With workers

who have very little knowledge about art, generic features like Gist and Phog work very well. This suggests that non-experts might themselves miss important markers of style.

The agreement scores obtained from the *D1* and *D2* HIT interfaces are comparable, with a slight tendency of *D1* judgments to agree with our features more. This indicates that when users are asked to provide justification for their choices, they are more likely to agree with our idea of what style is. Finally, we note that most workers considered themselves to be at least somewhat knowledgeable about art, while very few admitted they knew very little about it.

7. CONCLUSION AND FUTURE WORK

We described methodology and evaluation for a new *find-similar* search task in which a user seeks *stylistically* similar paintings. We approached this as a content-based image retrieval task, modeling stylistic similarity via various machine vision features designed specifically to capture differences between art styles. Results of system prediction using these features are promising and suggest significant potential for approximating expert judgments via automation.

Nevertheless, quality control of crowdsourced judgments remains an important issue going forward. While we filtered workers by approval rating, solicited feedback, and collected multiple judgments, a subjective task such as ours still requires stricter safeguards for quality. Workers naturally want to optimize their hourly pay, and some spam workers will try the fastest approach of all: random clicking (manually or via a bot). Strategies like qualification tests and trap questions can provide valuable quality assurance against such spam and are critical for ensuring that subsequent analyses are based on a solid foundation of quality data. Human-computer interaction (HCI) is also important for effective design and interaction with people in order to engage them, clearly convey instructions, and support effective collaboration between man and machine.

Another clear direction for future work is using the collection human judgments we have collected to train a supervised *learning to rank* model [12] which integrates weighted features into a unified model for predicting stylistic similarity. As mentioned earlier, use of more sophisticated strategies for label selection [16] *a la* active learning and label aggregation [18, 21] is also expected to help by collecting more labels where they are most needed and modeling the varying accuracy of different human judges.

In terms of computer vision features, interesting possibilities abound. We can obtain a segmentation over each image (resulting in distinct regions such as, for instance, beach, boat, flowers, building, sky) and compute our histograms over those regions rather than over fixed-size grid cells. We might also incorporate higher-level style markers.

Acknowledgments

Kristen Grauman directed us to some valuable related work and provided several useful images. Unmil P. Karadkar provided helpful discussions of the distinctions between art styles. Finally, we thank our anonymous reviewers and those who provided feedback on our initial MTurk HIT design.

8. REFERENCES

[1] Texture Classification. `http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html`.

[2] Wikimedia Commons. `http://commons.wikimedia.org/wiki/Main_Page`.

[3] A. Bosch and A. Zisserman. Pyramid Histogram of Oriented Gradients (PHOG). `http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html`.

[4] V. Carvalho, M. Lease, and E. Yilmaz, editors. *Proceedings of the ACM SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010)*. `http://ir.ischool.utexas.edu/cse2010/CSE2010-Proceedings.pdf`.

[5] H. Farid. Digital Art Forensics. `http://www.cs.dartmouth.edu/farid/research/artforensics.html`.

[6] J. Geusebroek, A. Smeulders, and J. van de Weijer. Fast Anisotropic Gauss Filtering. *IEEE Transactions on Image Processing*, 12(8):938–943, 2003.

[7] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *International Conference on Computer Graphics and Interactive Techniques: Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, 2001.

[8] M. Irfan and D. Stork. Multiple visual features for the computer authentication of Jackson Pollocks drip paintings: Beyond box-counting and fractals. *SPIE Electronic Imaging: Machine vision applications II*, 7251:1–11, 2009.

[9] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[10] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[11] C. Li and T. Chen. Aesthetic visual quality assessment of paintings. In *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Visual Media Quality Assessment*, April 2009.

[12] T. Liu. Learning to Rank for Information Retrieval. *Information Retrieval*, 3(3):225–331, 2009.

[13] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Intl. Journal of Computer Vision*, 42(3):145–175, 2001.

[14] D. Parikh, A. Gallagher, and T. Chen, editors. *Proceedings of the IEEE CVPR Workshop on Advancing Computer Vision with Humans in the Loop (ACVHL)*. 2010.

[15] C. Schmid. Constructing models for content-based image retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 39–45, 2001.

[16] V. Sheng, F. Provost, and P. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622. ACM, 2008.

[17] M. Smucker and J. Allan. Find-similar: similarity browsing as a search tool. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, page 468, 2006.

[18] R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

[19] A. Sorokin and L. Fei-Fei. Mechanical turk for computer vision. In *Tutorial at the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[20] M. Vill and R. Sablatnig. Stroke ending shape features for stroke classification. In *Proceedings of the Computer Vision Winter Workshop*, pages 91–98, 2008.

[21] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems (NIPS)*, 22:2035–2043, 2009.

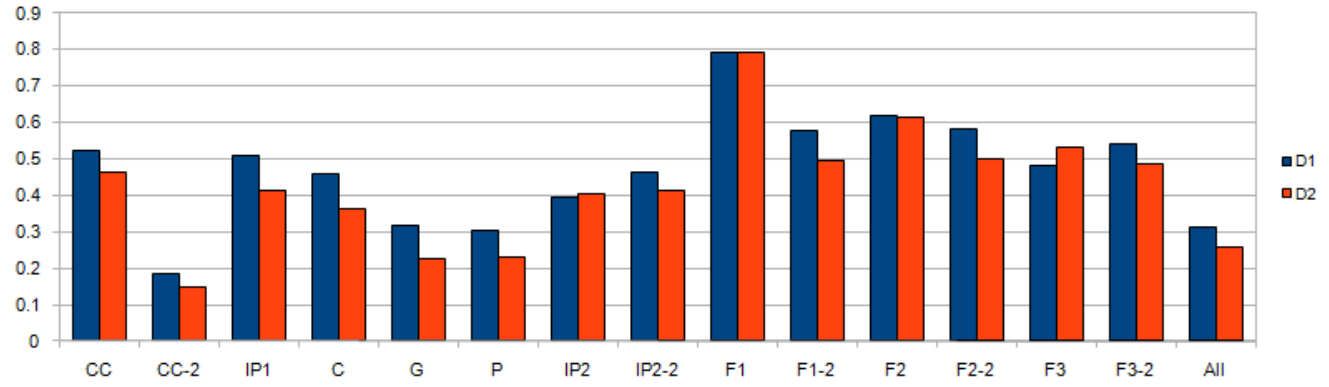

Figure 6: Feature agreement with most knowledgeable workers. Left and right bars correspond to *D1* and *D2* conditions, respectively.

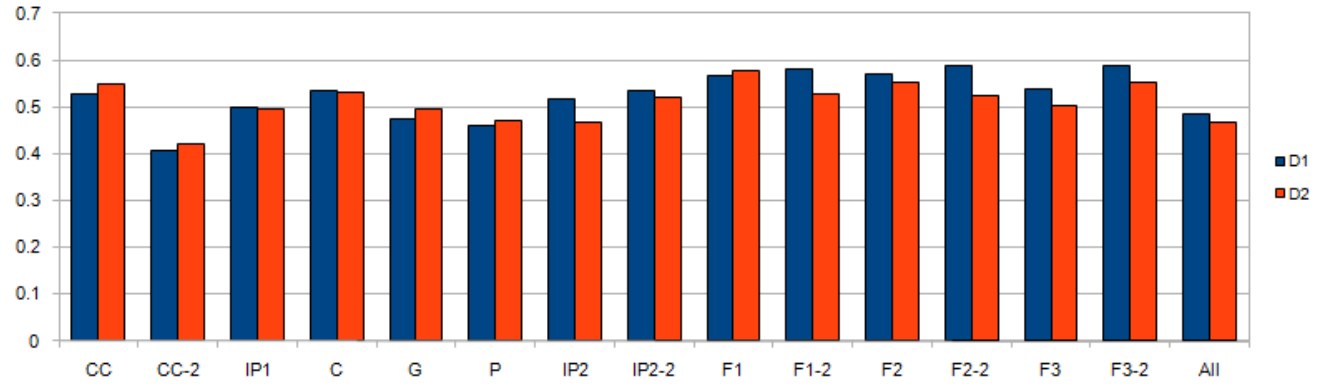

Figure 7: Feature agreement with somewhat knowledgeable workers. Left and right bars correspond to *D1* and *D2* conditions, respectively.

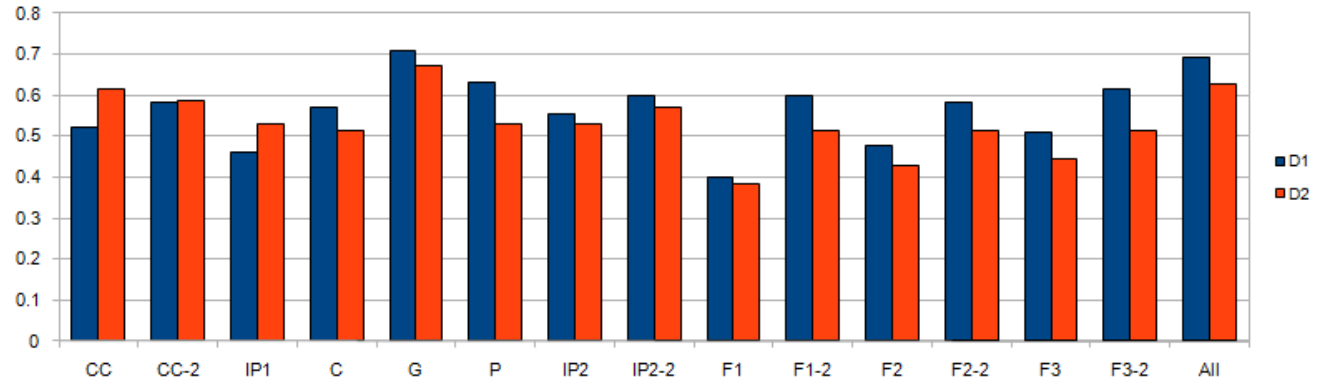

Figure 8: Feature agreement with least knowledgeable workers. Left and right bars correspond to *D1* and *D2* conditions, respectively.