Predicting User Annoyance Using Visual Attributes

Gordon Christie
Virginia Tech
gordonac@vt.edu

Amar Parkash
Goibibo
amar08007@iiitd.ac.in

Ujwal Krothapalli
Virginia Tech
ujjwal@vt.edu

Devi Parikh
Virginia Tech
parikh@vt.edu

Abstract

Computer Vision algorithms make mistakes. In human-centric applications, some mistakes are more annoying to users than others. In order to design algorithms that minimize the annoyance to users, we need access to an annoyance or cost matrix that holds the annoyance of each type of mistake. Such matrices are not readily available, especially for a wide gamut of human-centric applications where annoyance is tied closely to human perception. To avoid having to conduct extensive user studies to gather the annoyance matrix for all possible mistakes, we propose predicting *the annoyance of previously unseen mistakes by learning from example mistakes and their corresponding annoyance. We promote the use of attribute-based representations to* transfer *this knowledge of annoyance. Our experimental results with faces and scenes demonstrate that our approach can predict annoyance more accurately than baselines. We show that as a result, our approach makes less annoying mistakes in a real-world image retrieval application.*

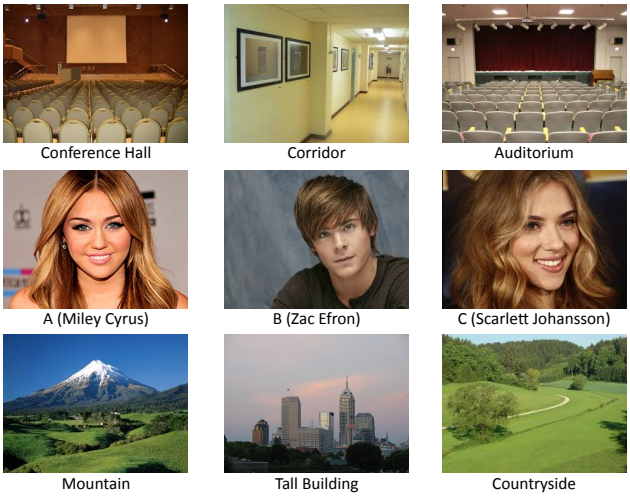

Figure 1: Some mistakes are more annoying for users than others. For each of the three examples (rows), if you were looking for the picture on the left, would you be more annoyed if the search engine returned the picture in the middle or the one on the right? We propose to use attribute-based representations to predict the annoyance of a novel previously unseen mistake.

1. Introduction

State of the art image understanding algorithms in computer vision today are far from being perfect. They make a lot of mistakes. But not all mistakes are equal. Some mistakes are worse or more costly than others. In order to train computer vision systems to minimize the overall cost of mistakes they make, and not just the number of mistakes, one needs access to the cost matrix that specifies the cost of every possible mistake.

Where do these cost matrices come from? For some applications like pedestrian detection, the relative costs of the different types of mistakes are driven by the domain. One would expect that false negatives are considered to be significantly worse than false positives when an autonomous vehicle is driving on the road. Industry standards or safety laws may dictate the relative cost.

However for human-centric applications like image search or retrieval, the costs of the different kinds of mistakes are tied to human perception. For common objects and scenes, resources like the WordNet can provide a meaningful distance between categories, which could be converted to a cost matrix. However, these need not necessarily translate well to how annoying users may find certain mistakes. For instance, according to the SUN hierarchy of scenes [35], conference halls and corridors are more similar (both fall under "workplace") than conference halls and auditoriums ("cultural"). However, when searching for images of conference halls, if the search engine were to make a mistake, we expect users to prefer that it return images of auditoriums rather than corridors. That is, from the user's perspective, returning images of corridors would be a more annoying mistake on the part of the search engine than returning auditoriums. See Figure 1 (top). Moreover, there are many visual concepts of interest such as identities of people (*e.g.* when searching for photographs of celebrities) that are not well organized in such ontologies. One could conduct user studies to identify the cost of each mistake *e.g.* given two images (or categories), ask subjects how annoyed they would be if the search engine returned the second image (or an image from the second category) when they were in fact looking for the first one. The size of such a cost matrix would be quadratic in the number of images (cat-

egories) involved, making this option impractical. This is especially the case when the image database may be dynamically growing. To alleviate this, we propose to *predict* the annoyance of a novel mistake from examples of known mistakes and their annoyances.

What representation should be used to facilitate this transfer? Let's consider the following question: What makes mistakes annoying? We conducted the following study. We showed human subjects on Amazon Mechanical Turk image triplets (A, B, C). We asked them "If you were looking for a picture of A, would you be more annoyed if the search engine returned a picture of B instead, or a picture of C? Why?". For the triplet shown in Figure 1 (middle), subjects consistently said for Miley Cyrus (A) that they would be less annoyed to get a picture of Scarlett Johansson (C) than Zac Efron (B). [1] More importantly, reasons stated by subjects include statements like "A and C are at least the same gender".

This clearly suggests that what makes mistakes annoying is differences in semantic attributes between images that are mistakenly treated to be the same (or similar). Attributes are mid-level visual concepts that are shareable across related categories such as "furry", "young" and "natural". Their influence on annoyance of mistakes is not surprising. The vocabulary of attributes is generated by humans, and by definition captures properties that humans care about. Hence, we argue that attributes provide the appropriate representation to deduce the cost of novel mistakes from example mistakes and their associated costs. While attributes have been explored for a variety of problems in the past few years (more about this in Section 2), we find that the use of attributes to get a handle on annoyance of mistakes made by our existing systems in human-centric applications is under-explored.

Some attribute violations are more annoying than others. For instance, when searching for a picture of a mountain scene (a natural scene with large objects [20]), a search engine returning a tall building scene (a manmade scene with large objects) is more annoying for a typical user than it returning a country side scene (a natural scene without large objects). See Figure 1 (bottom). Clearly, the attribute "is natural" is more critical to maintain than the attribute "has large objects". Our annoyance prediction model learns a weighting of attributes given example mistakes and their associated annoyances.

Note that the notion of annoyance provides us with a continuous spectrum between "not-mistakes" and mistakes. Non-mistakes, by definition, are the least annoying "mistakes". An effort towards minimizing overall annoyance thus involves minimizing the number of mistakes as well as minimizing the annoyance of individual mistakes. While the community has traditionally focused on the former, we argue that the latter is equally important, especially for user-centric applications.

A mistake consists of a pair of images (categories) that are mistakenly considered to be similar when they are not. We represent each mistake by the difference and similarity in the attribute signatures of the two images (categories) involved (*i.e.* which attributes are present/absent in both or present in one but not the other). Given a training set of pairs of images (categories) and the associated cost of mistakenly considering them to be similar, we learn a mapping from this attribute representation to the cost. Given a novel pair of images (categories), we predict the presence/absence of attributes in the pair. We compute differences and similarities between the attribute signatures of the two and use our trained annoyance prediction model to estimate the likely annoyance a user would experience if these two images (categories) were confused with each other. We experiment with two domains (faces and scenes) and show that attributes can be used to infer the annoyance of previously unseen mistakes. Attributes allow this transfer of knowledge from previously seen mistakes to new mistakes better than low-level features (e.g. gist, color), even when the attribute predictors are trained on the *same* low-level features. Finally, we show that our proposed approach outperforms two baselines at an image retrieval task.

2. Related Work

We now relate our work to existing works that explore the use of attributes for a variety of tasks, reason about image similarity and predict semantics concepts in images for improved image retrieval.

Attributes: Attributes have been used extensively, especially in the past few years, for a variety of applications [2, 3, 7–9, 13, 15, 17, 21–23, 31–33]. Attributes have been used to learn and evaluate models of deeper scene understanding [7] that reason about properties of objects as opposed to just the object categories. They have also been used for alleviating annotation efforts via zero-shot learning [17, 22] where a supervisor can teach a machine a novel concept simply by describing its properties (*e.g.* "a zebra is striped and has four legs" or "a zebra has a shorter neck than a giraffe"). Attributes being both machine detectable and human understandable provide a mode of communication between the two. This has been exploited for improved image search by using attributes as keywords [15] or as interactive feedback [13], or for more effective active learning by allowing the supervisor to provide attribute-based feedback to a classifier [23], or even at test time with a human-in-the-loop answering relevant questions about the test image [3]. Attributes have been used for generating automatic textual descriptions of images [14, 22] that can potentially point out anomalies in objects [8]. Attributes have also been explored to improve object categorization [8] or face verification performance [16]. However, in spite of attributes

[1] In this paper, we are concerned with visual search, and not search based on other metadata such as which celebrities (in this case) are co-stars, or are dating, etc.

inherently being properties humans care about, they have not been explored for reasoning about human perceived annoyance of mistakes made by a vision system.

Learning similarity: The notion of annoyance or user satisfaction is closely related to image similarity. Many works in machine learning in general and computer vision in particular propose novel formulations to learn similarity functions [1, 10, 11, 26] that better respect ground truth image similarity constraints. Since these similarity constraints are often elicited from semantic labels of images, they can be viewed as being a proxy for human perception. In fact, efforts have been made at building kernels that explicitly store only human perceived similarity between images [28]. These advancements are orthogonal to our work. We do not propose a novel similarity learning approach. Instead, we argue that attributes are a better *representation* to reason about human perceived image similarity than commonly used features. As we will demonstrate in our results, learning a similarity (or mistake annoyance) measure on *predicted* attributes significantly outperforms learning one on the *same* low-level features used to predict the attributes.

Retrieval: Semantic concepts have been used for multimedia retrieval [6, 19, 25, 27, 34, 36] to reduce the semantic gap. The semantic concepts are often well aligned with what the users are expected to search for. Attributes are a mid-level representation between low-level features and the high level semantic concepts. They provide an intermediate representation that we argue mimics human perception of image similarity better than low-level features and are thus better aligned to reason about annoyance of mistakes in human-centric applications. These attributes need not themselves be the target high level concepts of interest in the task at hand. Existing work has looked at other intermediate representation of images. For instance an image can be represented by a signature that captures the classification or detection score of various object categories in the image [18, 29]. These have been shown to provide improved classification performance in terms of accuracy (and efficiency) *i.e.* fewer mistakes. To the best of our knowledge, ours is the first to study the ability of such intermediate representations to reason about the annoyance (and not number) of mistakes. We focus our efforts on attributes as the intermediate representation since they directly capture properties of concepts that humans care about are are thus intuitively likely to affect annoyance of mistakes. While applicable to personalized search, capturing user *preferences* (e.g. when looking for red shiny high-heel shoes, a particular user may be willing to compromise on the shininess but not on the height of the heels) is not the focus of this work.

3. Approach

Our goal is to predict the annoyance of a mistake. We consider mistakes of the following form: an image from one category (say i) is classified as a different category (say j). The annoyance of this mistake i.e. the cost of this mistake is c_{ij}. As training data, we are given a set of N triplets: $D = \{(i, j, c_{ij})\}$ consisting of pairs of categories i and j along with their associated costs c_{ij}.

We are also given a set of M pre-trained attribute predictors along with example images from each of the K categories. We evaluate the attribute predictors on these images and take a majority vote across images from the same category to determine the attribute memberships of the categories. $a_i^m = \{0, 1\}$ indicates whether attribute m is present or absent in category i. If from domain knowledge (or from a supervisor), these attributes memberships are known a priori, those can be used as well instead of attribute predictors.

We use a $2M$-dimensional feature vector $\boldsymbol{d}_{ij}$ to represent each category pair (i, j). Intuitively, the annoyance of a mistake depends on which attributes are different between the two categories. But notice that the presence of common attributes in the two categories can contribute towards *reducing* the annoyance. Our descriptor thus captures both differences and commonalities in the attribute signatures of the two categories. $\boldsymbol{d}_{ij} = [\boldsymbol{d}_{ij}^m \; \boldsymbol{d}_{ij}^{2m}]$ where $\boldsymbol{d}_{ij}^m = a_i^m \oplus a_j^m$ and $\oplus$ is the exclusive OR (differences), and $\boldsymbol{d}_{ij}^{2m} = |a_i^m \wedge a_j^m|$, where $\wedge$ is the logical AND (similarities).

We learn a mapping from this descriptor to the annoyance of a mistake. That is

$$\hat{c}_{ij} = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{d}_{ij}) + b \tag{1}$$

where $\boldsymbol{\varphi}$ is a potentially high dimensional mapping of our feature vector $\boldsymbol{d}_{ij}$, and $\boldsymbol{w}$ and b are the parameters to be learnt. We learn these in two ways, which we describe next.

3.1. Preserving Annoyance Values

The first is applicable to scenarios where one may want to train a classification system that minimizes the overall cost of mistakes made. In this case, it is important for the predicted cost to be accurate. We convert the provided training data to a set $\{(\boldsymbol{d}_{ij}, c_{ij})\}$ consisting of the attribute-based representation of a pair of categories and its associated annoyance. We use the ϵ-insensitive loss function to penalize a prediction *i.e.* the loss is 0 if the difference between the predicted annoyance value $\hat{c}_{ij}$ and the true value c_{ij} is less than ϵ, otherwise it is the amount by which the difference $|\hat{c}_{ij} - c_{ij}|$ overshoots ϵ.

$$L(\hat{c}_{ij}, c_{ij}) = \max\{0, |\hat{c}_{ij} - c_{ij}| - \epsilon\} \tag{2}$$

ϵ is set via cross validation. We wish to minimize

$$\min \quad \sum_{i=1}^{K} \sum_{j=1}^{K} L(\hat{c}_{ij}, c_{ij}) \tag{3}$$

The training data need not involve all possible pairs of K categories but for ease of discussion, we assume that is

the case. We solve a regularized and relaxed version of this problem:

$$\min_{\boldsymbol{w},\xi_{ij},\xi^*_{ij},b} \quad \frac{1}{2}||\boldsymbol{w}||_2^2 + C\left(\sum_{i=1}^{K}\sum_{j=1}^{K}\xi_{ij} + \sum_{i=1}^{K}\sum_{j=1}^{K}\xi^*_{ij}\right) \tag{4}$$

$$s.t. \quad \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{d}_{ij}) + b - \hat{c_{ij}} \leq \epsilon + \xi_{ij}, \tag{5}$$

$$\hat{c_{ij}} - \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{d}_{ij}) - b \leq \epsilon + \xi^*_{ij}, \tag{6}$$

$$\xi_{ij}, \xi^*_{ij} \geq 0, i, j \in \{1, \ldots, K\} \tag{7}$$

where $||\boldsymbol{w}^T||_2^2$ is the large-margin regularization term, ξ_{ij} and ξ^*_{ij} are the slack variables, and C modulates the regularization vs. training error trade-off determined via cross validation. In our experiments we use the ϵ-Support Vector Regressor implementation of LIBSVM [4] with the RBF kernel to learn $\boldsymbol{w}$. Having learnt $\boldsymbol{w}$, given a new pair of categories p and q, we compute their representation $\boldsymbol{d}_{pq}$ and estimate their cost $\hat{c}_{pq} = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{d}_{pq}) + b$. We show this formulation for simplicity. In practice, we optimize the SVR dual expressed with RBF kernels rather an an explicit feature map $\boldsymbol{\varphi}$.

3.2. Preserving Relative Ordering of Annoyance

In applications such as image search, it is desirable to rank the images in increasing order of their likely annoyance. In this scenario, it is important to maintain the relative ordering of mistakes according to their annoyance, but the actual annoyance value is less critical.

At training time, we are given a set of ordered pairs of mistakes $O = \{((i,j),(i,'j'))\}$ such that $((i,j),(i',j')) \in O \implies c_{ij} > c_{i'j'}$, *i.e.* confusing classes i and j is more annoying to the user than confusing classes i' and j'. Note that if i and i' are the same, users may provide more consistent responses when gathering training data ("Would you be more annoyed if an image from class i was classified as j or as j'?"). But the approach is general and applicable even when i and i' are different.

Our goal is to learn a ranking function for the model shown in Equation 1. We use a linear feature map, and since the bias is irrelevant to a ranking function, we wish to learn $\boldsymbol{w}$ in $\hat{c}_{ij} = \boldsymbol{w}^T\boldsymbol{d}_{ij}$ such that the maximum number of the following constraints is satisfied:

$$\forall((i,j),(i',j')) \in O : \boldsymbol{w}^T\boldsymbol{d}_{ij} > \boldsymbol{w}^T\boldsymbol{d}_{i'j'} \tag{8}$$

While this is an NP hard problem [12], it is possible to approximate the solution with the introduction of non-negative slack variables, similar to an SVM formulation. We directly adapt the formulation proposed in [12], which was originally applied to web page ranking, except we use a quadratic loss function leading to the following optimization problem:

$$\min_{\boldsymbol{w},\xi_{iji'j'}} \quad \frac{1}{2}||\boldsymbol{w}^T||_2^2 + C\left(\sum \xi_{ij}^2\right) \tag{9}$$

$$s.t. \quad \boldsymbol{w}^T\boldsymbol{d}_{ij} \geq \boldsymbol{w}^T\boldsymbol{d}_{i'j'} + 1 - \xi_{iji'j'} \tag{10}$$

$$\xi_{iji'j'} \geq 0, \forall((i,j),(i',j')) \in O \tag{11}$$

Rearranging the constraints reveals that the above formulation is quite similar to the SVM classification formulation, but on pairwise difference vectors:

$$\min_{\boldsymbol{w},\xi_{iji'j'}} \quad \frac{1}{2}||\boldsymbol{w}^T||_2^2 + C\left(\sum \xi_{ij}^2\right) \tag{12}$$

$$s.t. \quad \boldsymbol{w}^T(\boldsymbol{d}_{ij} - \boldsymbol{d}_{i'j'}) \geq 1 - \xi_{iji'j'} \tag{13}$$

$$\xi_{ij} \geq 0, \forall((i,j),(i',j')) \in O \tag{14}$$

where C is the trade-off constant between maximizing the margin and satisfying the pairwise relative constraints. Notice that b from Equation 1 is not a variable in this formulation, since adding an offset to all costs would result in the same relative ordering. We solve the above primal problem using Newton's method [5]. We note that this learning-to-rank formulation learns a function that explicitly enforces a desired ordering on the training images; the margin is the distance between the closest two projections within all desired (training) rankings. Having learnt $\boldsymbol{w}$, given a new pair of categories p and q, we compute their representation $\boldsymbol{d}_{pq}$ and estimate their cost $\hat{c}_{pq} = \boldsymbol{w}^T\boldsymbol{d}_{pq}$. This predicted cost is not meaningful by itself, but allows us to meaningfully compare categories (p,q) to a different pair of categories (r,s). If $\hat{c}_{pq} > \hat{c}_{rs}$ we can conclude that confusing categories p and q is more annoying than confusing categories r and s. For an image search application, given the query category q, we are interested in sorting all other categories in increasing order of their annoyance. We would compare $c_{pq}, c_{rq}, c_{sq}, \ldots$ and sort categories $p, r, s, \ldots$ accordingly.

4. Experimental Setup

Datasets: We experiment with two domains: faces and scenes. For faces, we use a subset of the Public Figures Face Database [16] containing 8523 face images from 60 different public figures (categories) in the development set. We use a vocabulary of 63 out of 73 attributes [16] describing various category-level properties of faces such as race, age, gender, shape of nose, eye brows, etc. We discard the 10 attributes that make sense only at an image-level. We collect category-level ground truth annotations of these attributes (GTA) on Amazon Mechanical Turk which we have made publicly available on the last author's webpage. We use two attribute predictors. The first (K) uses the service provided by Kumar *et al.* [16] that uses state-of-the-art features[2] to predict facial attribute scores for any face image. For a fair

[2]not publicly available

comparison to low-level features, we also train our own attribute predictors PRA on low-level features using an SVM with a RBF kernel. We use 512-d gist features concatenated with 30-d color histogram features as our low-level features (LLF). Note that any LLF can be used to train the attribute predictors. More sophisticated LLFs would lead to improved PRA. We are interested in comparing PRA to LLF to evaluate the role of attributes as an effective representation for annoyance prediction. Evaluating the impact of different choices of LLF on annoyance prediction is orthogonal to our claims.

For scenes, we use a subset of the SUN scenes dataset [35] containing 1600 images from 80 categories. The categories were selected by picking 5 random categories from each of the 16 nodes at the second level of the scenes hierarchy which include shopping and dining, workplace, home or hotel, transportation, sports and leisure, mountains, forests, manmade elements, sports fields, construction, commercial buildings, historical buildings, etc. We use 62 out of the 102 attributes from Patterson and Hays [24] that are well represented in our 80 categories. These include material properties like foliage, surface properties like rusty, functions or affordances like camping and studying, spatial envelope properties like enclosed and open spaces, and object presences like cars and chairs. We use the ground truth attribute annotations GTA made available with the SUN Attribute Database [24]. We train PRA to predict 62 of these attributes, and use 512-d gist features as our low-level features LLF. As with PubFig, for a fair comparison to low-level features we train our own attribute predictors PRA using an SVM with a RBF kernel on the same low-level features.

Setup: To convert image-level responses (be it output of attribute classifiers or low-level features) to category-level signatures, we select 20 random images from each category, and average their responses. When learning weights for differences in low-level features (wLLF), $\boldsymbol{d}_{ij}$ is computed as $|f_i - f_j|$, where f_i is the low-level feature signature of category i. Learning our models on this descriptor mimics distance metric learning approaches. We use 40 random categories from PubFig for training, 10 random categories for validation and the remaining 10 categories for test. We use 60, 10 and 10 categories from SUN for training, validation and testing. Given a 40×40 and 60×60 cost matrix for PubFig and SUN at training time, our goal is to predict a 10×10 cost matrix at test time given 20 random images from each of these 10 categories. Notice that none of the test categories were available at training time. Our approach estimates the annoyance of a mistake that confuses two categories – *none* of which were seen before. We report average results across 200 or more random train/val/test splits.

Ground truth annoyance: We collect the ground truth cost or annoyance matrices via real user studies on Amazon Mechanical Turk. Subjects were asked how annoyed they would be if they were searching for images from class A and the search engine instead returned images from class B. They were given five options ranging from 1: "Not annoyed at all" to 5: "Very annoyed". We asked multiple workers (10 for PubFig, 5 for SUN) to respond to each pair of categories, and averaged their responses.[3] This provides us with the ground truth annoyance matrices used to train our annoyance predictor at training time, and to evaluate our results at test time. These values are used as is to train the regressor (Section 3.1). We select 5000 random pairs of category-pairs, determine their pairwise ordering using the ground truth annoyance values, and used the ordered pairs to train the ranking function (Section 3.2).

Metrics: We evaluate our regressor-based annoyance predictor using the Mean Squared Error (MSE) between the predicted cost matrix and the ground truth matrix.

$$\text{MSE} = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}(\hat{c}_{ij} - c_{ij})^2, i \neq j \tag{15}$$

We assume $\hat{c}_{ii} = c_{ii} = 0$ and do not include these terms during training or evaluation. Lower MSE is better.

We evaluate our ranker-based annoyance predictor using Spearman Rank Correlation (RC) coefficient between the true annoyance values and the predictions.

$$\text{RC} = \frac{\sum_{i=1}^{K}\sum_{j=1}^{K}(r_{ij} - \bar{r})(\hat{r}_{ij} - \bar{\hat{r}})}{\sqrt{\sum_{i=1}^{K}\sum_{j=1}^{K}(r_{ij} - \bar{r})^2 \sum_{i=1}^{K}\sum_{j=1}^{K}(\hat{r}_{ij} - \bar{\hat{r}})^2}} \tag{16}$$

where r_{ij} is the rank of c_{ij} when all categories pairs are sorted according to their annoyance and $\bar{r}$ is the mean value of r_{ij} *i.e.* $\bar{r} = \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K} r_{ij}$. This corresponds to the Pearson's correlation coefficient but on ranks of data points instead of the data points themselves. Higher RC is better.

5. Results

We evaluate the performance of our approach on predicting the annoyance of previously unseen mistakes, as well as its resultant ability to provide search results that are less annoying for users.

5.1. Annoyance Prediction

Our proposed approach represents categories using their attributes memberships, as provided by a supervisor (GTA) or predicted automatically using classifiers (PRA and K for PubFig). The performance of our approach as compared to the baseline approach of learning a weighted difference of low-level features (wLLF) can be seen in Figure 2. Our approach (gPRA) performs significantly better in terms of

[3] Standard error of the mean was 0.38 and 0.57 for PubFig and SUN respectively.

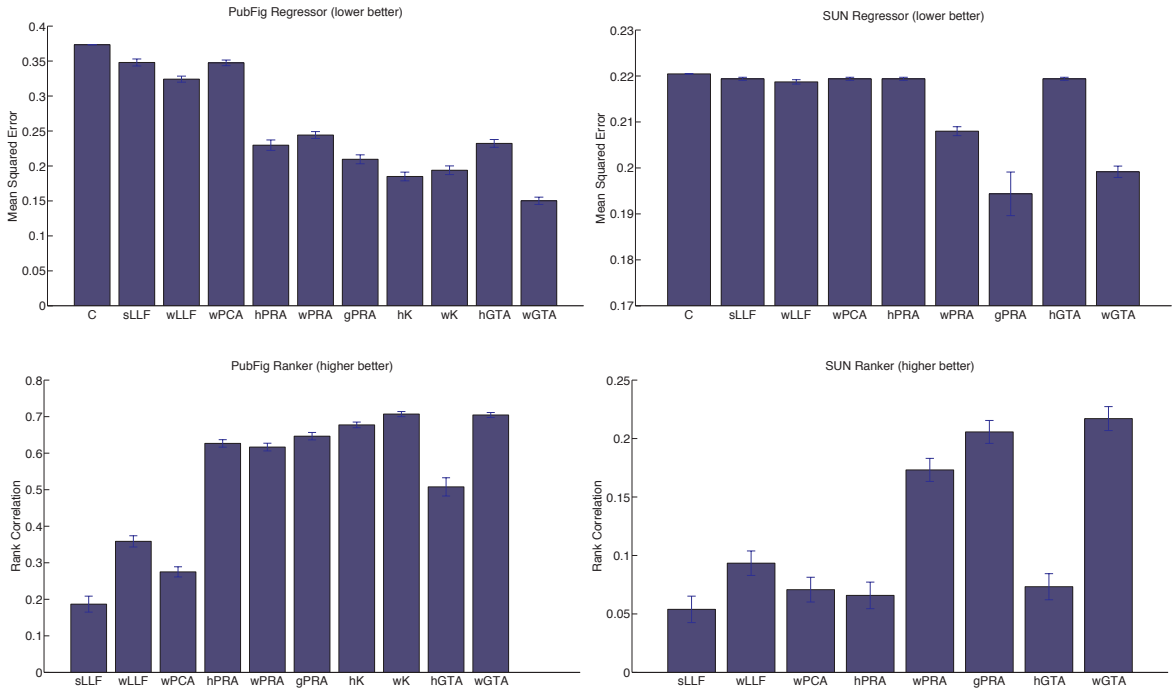

Figure 2: Annoyance prediction accuracy of our approach compared to various baselines. The lowercase letters preceding the method acronyms are as follows. w: weighted feature differences g: gate logic, where the binarized attribute scores are used to concatenate the outputs of an exclusive OR and AND operations, s: similarity, where the euclidean distance between the two feature vectors is used, h: hamming distance. The proposed method is PRA, which is the predicted attributes using low-level features. GTA: ground truth attributes and K: predicted attributes of Kumar *et al*. [16] are shown for completeness, but are not comparable to our approach because GTA requires annotated attributes at test time and K uses different low-level features than us that are not publicly available. Baselines are LLF: low-level features, PCA: principal component analysis on low-level features, C: constant prediction.

MSE using the regressor as well as RC using the ranker. Note that PRA uses attributes predictors trained on the same low-level features. This shows that the improvement in performance can be directly attributed to the use of semantic mid-level concepts (as opposed to better low-level features).

We show the performance of GTA and K for sake of completeness. But those cannot be directly compared to PRA since K involves the use of better low-level features than LLF, and GTA involves access to ground truth attribute memberships of test categories. Note that even with GTA, annoyance may not be predicted perfectly because the mapping from attributes to annoyance is *learnt*.

Need for differences *and similarities*: To evaluate the improvement in performance by modeling both commonalities and differences in attribute presences, we train our models using just differences in attribute scores[4] rather than outputs of the two logic gates (AND and XOR) on the binary attribute predictions as described earlier. This baseline (wPRA) performs significantly worse than reasoning about both commonalities and differences (gPRA).

Other mid-level representations: One might wonder if the improvement in performance of our approach is simply because of its reduced dimensionality and not so much the semantic nature of attributes. To address this, we compare our proposed approach to an approach that uses Principal Component Analysis on the low-level features to obtain a representation that is the same dimensionality as the number of attributes, and then learns a weighted difference for each dimension (wPCA). We used the implementation of [30]. For a fair comparison, we compare it to learning a weighted difference in attributes and ignoring similarities (wPRA). In Figure 2 we find that wPRA significantly outperforms wPCA. This demonstrates the benefit of using attributes – *semantic* mid-level concepts – for reasoning about annoyance of mistakes as perceived by users.

Need for weighing attributes: To evaluate the need to weigh the attributes (as learnt by $\boldsymbol{w}$ in our approach), we compare to a baseline that simply counts the number of attribute differences (similarities are redundant in this case), instead of weighing each one differently. It then learns a scalar weight and a bias for the regressor. A ranker can not change the order of points given a scalar feature and hence need not be learnt. In Figure 2, we see that our approach (gPRA) significantly outperforms this hamming-distance based approach (hPRA). This shows that some attributes are indeed more important to users, and modeling this importance improves our approach's ability to predict annoyance of novel mistakes. Compared to wPRA that also learns weights but only reasons about differences, hPRA performs significantly worse on the SUN dataset that has a larger variety of attributes and images. Learning the

[4]Using soft scores performed better than the binary decisions.

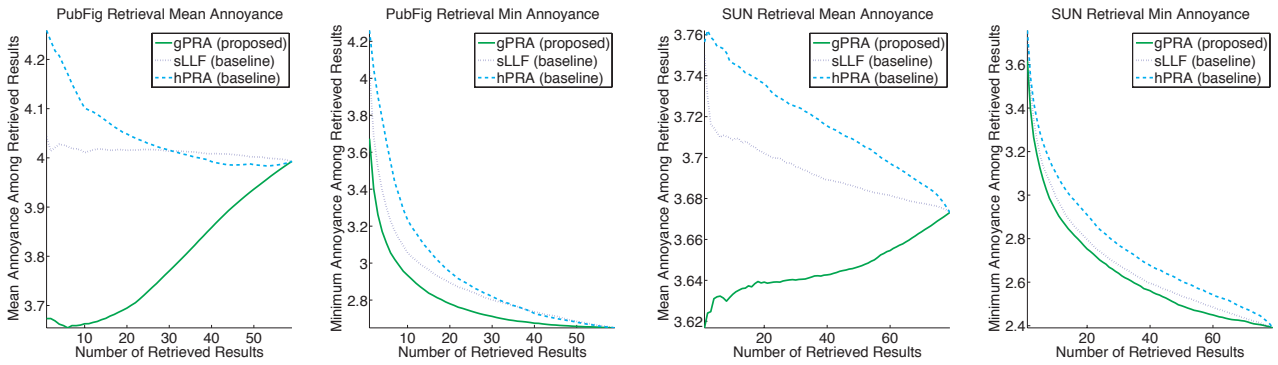

Figure 3: Image retrieval results. Our approach returns less annoying results than the baseline approaches. Note that a difference of 0.5 on the y-axis corresponds to a difference of 12.5% on our scale (1 to 5). Lower is better.

weights for GTA seems to be especially crucial – wGTA performs significantly better than hGTA, and in fact, hGTA performs worse than several automatic attribute prediction approaches. Perhaps automatic attribute predictors tend to predict the visually obvious attributes well, and those may be the more important ones to human perception. GTA does not benefit from this bias, unless we learn to explicitly down weigh a subset of attributes (via wGTA). In image-level search it is common to compare images based on their similarity in lower level feature spaces. We therefore compare our approach to the euclidean distance between LLF feature vectors (sLLF). Our approach significantly outperforms this method.

Sanity check: Finally, as a sanity check for the regressor, we report the MSE of a baseline C that predicts a constant (the average annoyance value for all categories pairs) for all mistakes. Most baselines outperform this approach.

5.2. Image Search Results

We now consider image retrieval, a common real-world human-centric computer vision application. This also gives us an opportunity to evaluate our approach at the image level rather than the category level. We choose a random image from each category 25 times and use it as a query. Each time, we use cross-validation to predict the cost of returning images from all other categories. We use 6/8 folds for PubFig/SUN respectively. We then return the R least annoying images. We record the mean and min annoyance across the R returned images from the ground truth cost matrix. This captures how annoying the returned results are on average, and how annoying the least annoying result among the R results is. We average these results at each value of R over different queries.

We compare our approach (gPRA) to two baselines. The first counts how many attributes a database image has in common with the query image (similar in spirit to Kumar *et al*. [15]). It computes the hamming distance between predicted attributes on the query and database image (hPRA). The second baseline computes similarity between the query image and the database image using the Euclidean distance between low-level features (sLLF). Results are shown in Figure 3. Clearly, our approach returns less annoying results to the user than either of these baseline approaches. As evidenced by the increasing mean annoyance values for our approach with more retrieved results, our approach ranks the truly least annoying results first. Note that the true annoyance values were measured using user studies. Hence, they capture true user experience / satisfaction.

We show some qualitative results in Figure 4. Consider the second row from the top. Our approach has learnt that attributes such as "has bangs" and "is not wearing eye glasses" are not sufficient, and weighs gender more, leading to a less annoying retrieval result. A common reason for inaccurate prediction of annoyance is inaccurate predictions of attributes that are given a high weight by our approach.

6. Discussion

Attributes are shareable across related categories, but not across drastically different domains. For instance, face attributes are not relevant to describe furniture. As with any learning-based approach, the training annoyance data must be collected on categories related to test categories such that the attribute vocabulary can be expected to generalize.

To collect annoyance ground truth, we asked subjects how annoying a mistake would be. If we also ask subjects the reasons for the annoyance, the resultant responses can be mined to discovery a vocabulary of attributes relevant to the task at hand. Moreover, the form of the response – e.g. "This mistake is very annoying because image A is natural but image B is manmade" – can also provide us with annotations of the attributes on the images. This can be used to train the attribute predictors. Hence, this annoyance annotation interface can be a powerful tool for discovering the vocabulary of relevant attributes as well as gathering the annotations for this vocabulary.

The annoyance of a mistake, and in fact the definition of "mistake" itself, depends on the task at hand. Hence, even within the same domain (e.g. faces), different models of annoyance may be required to reason about say appearance- vs. meta-data-based image-retrieval.

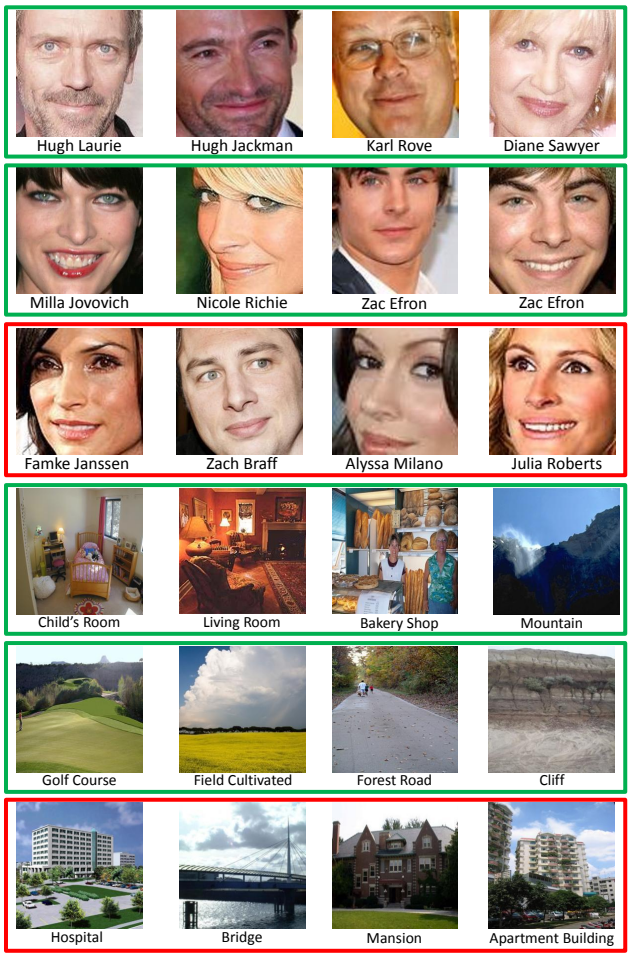

Figure 4: Qualitative image-level search results for the PubFig (top 3 rows) and SUN (bottom 3 rows) datasets. Images in the first column are the query image, and all other images are the predicted least annoying images returned by our approach (gPRA, second column), hamming distance on attributes (hPRA, third column) and similarity of low level features (sLLF, fourth column). Green boxes show success cases, and red boxes show failure cases.

The notion of annoyance may be user-specific. By collecting annoyance information from individual users to train our model, our approach can be leveraged for predicting user-specific annoyance of mistakes, for improved personalized image search. Actively choosing pairs of categories for gathering annoyance annotations and exploring matrix completion algorithms to predict the annoyance matrix from sparse annotations is part of future work.

Conclusion: In this work we focus on the novel problem of predicting the annoyance of previously unseen mistakes. We promote the use of attribute-based representations for this task. We argue that differences and similarities in attribute signatures – as opposed to low-level feature representations – contribute to the annoyance of mistakes. We collect ground truth annoyance for faces and scenes, which we have made publicly available, to learn our models and evaluate them. We show that our approach can predict annoyance more effectively than several baselines. This allows us to make less annoying mistakes in an image retrieval task, resulting in improved user experience.

Acknowledgements: This work is supported in part by a Google Faculty Research Award (DP) and NSF IIS-1115719.

References

[1] D. Batra, R. Sukthankar, and T. Chen. Semi-supervised clustering via learnt codeword distances. In *BMVC*, 2008. 3
[2] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010. 2
[3] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010. 2
[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011. 4
[5] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 2007. 4
[6] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011. 3
[7] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 2
[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2
[9] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 2
[10] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007. 3
[11] P. Jain, B. Kulis, I. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *NIPS*, 2008. 3
[12] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002. 4
[13] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, pages 2973–2980. IEEE, 2012. 2
[14] G. Kulkarni, V. Premraj, S. L. Sagnik Dhar and, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR*, 2011. 2
[15] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2010. 2, 7
[16] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 2, 4, 6
[17] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2
[18] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010. 3
[19] M. Naphade, J. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 2006. 3
[20] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 2
[21] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011. 2
[22] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 2
[23] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012. 2
[24] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012. 5
[25] N. Rasiwasia, P. Moreno, and N. Vasconcelos. Bridging the gap: Query by semantic example. *IEEE Transactions on Multimedia*, 2007. 3
[26] G. Shakhnarovich. Learning task-specific similarity. In *Ph.D. Thesis, MIT*, 2006. 3
[27] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *ICME*, 2003. 3
[28] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. Adaptively learning the crowd kernel. In *ICML*, 2011. 3
[29] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 3
[30] L. van der Maaten. Matlab toolbox for dimensionality reduction (v0.8.1 - march 2013). 6
[31] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009. 2
[32] G. Wang, D. Forsyth, and D. Hoiem. Comparative object similarity for improved recognition with few or no examples. In *CVPR*, 2010.
[33] J. Wang, K. Markert, and M. Everingham. Learning models for object recognition from natural language descriptions. In *BMVC*, 2009. 2
[34] X. Wang, K. Liu, and X. Tang. Query-specific visual semantic spaces for web image re-ranking. In *CVPR*, 2011. 3
[35] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 5
[36] E. Zavesky and S.-F. Chang. Cuzero: Embracing the frontier of interactive visual search for informed users. In *Proceedings of ACM Multimedia Information Retrieval*, 2008. 3