Image Retrieval with Mixed Initiative and Multimodal Feedback

Nils Murrugarra-Llerena
nineil@cs.pitt.edu
Adriana Kovashka
kovashka@cs.pitt.edu

Department of Computer Science
University of Pittsburgh
Pittsburgh, PA, USA

Abstract

How would you search for a unique, fashionable shoe that a friend wore and you want to buy, but you didn't take a picture? Existing approaches propose interactive image search as a promising venue. However, they either entrust the user with taking the initiative to provide informative feedback, or give all control to the system which determines informative questions to ask. Instead, we propose a *mixed-initiative* framework where both the user and system can be active participants, depending on whose initiative will be more beneficial for obtaining high-quality search results. We develop a reinforcement learning approach which dynamically decides which of three interaction opportunities to give to the user: drawing a sketch, providing free-form attribute feedback, or answering attribute-based questions. By allowing these three options, our system optimizes both the informativeness and exploration capabilities allowing faster image retrieval. We outperform three baselines on three datasets and extensive experimental settings.

1 Introduction

Computer vision apps serve a variety of user needs: for example, they can automatically count calories [25], summarize vacation footage [50], "paint" [10], or help users find shoes they want to buy [19] via image search. While for calorie-counting or machine-painting the interaction between the user and the machine is limited to submitting a photograph, for image search the user needs to communicate with the system in a more fine-grained and unrestricted fashion, since success is defined by whether the system successfully "guessed" what the user wanted to find. A person can look for online shopping options on products they saw in a store, or even try to find a criminal they saw in an online database. The user's mental concept of what they wish to retrieve can be arbitrarily subtle hence difficult to capture, and in order to ensure that the system's model of the user's search concept is accurate, the user needs to be able to "explain" to the system how it should adjust its predictions.

Prior work has tackled this challenge in a number of ways. Some work has used semantic visual attributes (like "shiny" or "chubby") [7, 19, 23, 31] to allow the user to give precise language-based guidance to the system. Attributes provide an excellent channel for communication because humans naturally explain the world to each other with adjective-driven descriptions. Attributes have been shown promising as a tool for image search [12, 19, 22, 32, 38, 53]. For example, [19] show how a user can perform rich relevance feedback by specifying how the attributes of a results image should change to better match the user's

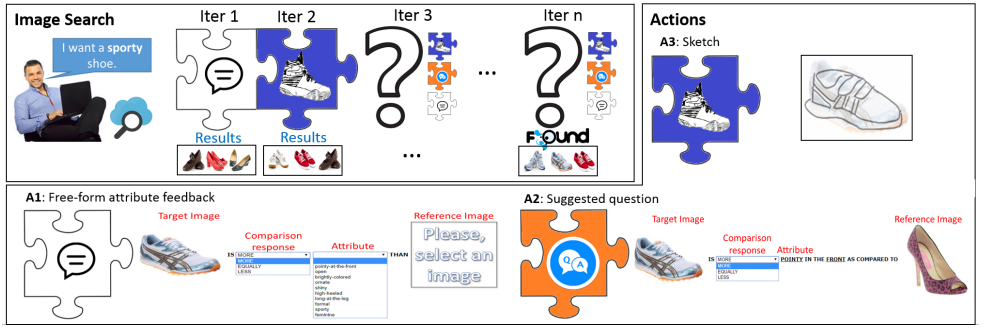

Figure 1: We learn how to intelligently combine different forms of user feedback for interactive image search, and find the user's desired content in fewer iterations. The *image search* section depicts our search agent that predicts an appropriate action at a certain iteration. For example, our agent selects free-form attribute feedback for iteration 1, and sketching for iteration 2. The *actions* section presents the three possible interactions (actions) of our agent.

target image. For example, the user might say "Show me people with longer hair than this one." Another approach has been to engage the user in question-answering with questions that the system estimated are most useful [8, 16]. Thus, in prior work, the initiative for what guidance to give to the system has been taken by either the user [18, 19, 22, 38, 53] or system [8, 16, 44] *but not both.* Another approach has been to allow the user to provide visual cues for what they are looking for, e.g. by drawing a sketch [6, 34, 54, 55]. The system can then retrieve visually similar results. Thus, the user can use either language or visuals to search, but it is not clear which modality is more informative.

In our work, we propose a framework where **either** the user **or** system can drive the interaction, and the input modality can be **either** textual **or** visual, depending on what seems most beneficial at any point in time. For example, the user can kick off the search using a sketch, then refine the results by explaining how the top retrieved images at a certain iteration differ from her mental model. Then the system might ask attribute-based questions, and give control back to the user when it runs out of informative questions to ask, so the user can provide some more free-form attribute feedback of her choosing. Since it is the system that must rank the results, we propose to leave the choice of what is most informative to the system. In other words, the system can decide to let the user lead and *explore*, if it cannot *exploit* any relevant information in a certain iteration. The system can request that the user provides multimodal feedback, i.e. textual **or** visual feedback. To make all these decisions, we train a reinforcement learning agent (see Fig. 1).

In particular, the options that the reinforcement learning chooses between are: (1) sketch feedback, (2) free-form attribute feedback, or (3) system-chosen attribute questions. At each iteration, the system adaptively chooses one of these interactions and asks the user to provide the corresponding type of feedback (e.g. it asks the user to choose an image and attribute to comment on). Briefly, our method works as follows. Our agent receives a state composed of top result images, proxies for the target image, and history of taken actions, when available. It interacts with the environment trying different actions. Over time, it learns to pick the most meaningful action, given a certain state. We guide our agent with information about whether the target image is among our top results.

Note that while it is the system that decides what *type* of search interaction is most useful, both the user and system are *active* participants in the search. When the system

gives control back to the user, the user can freely choose what language-based feedback to provide or what imagery to sketch. This is in contrast to prior work involving human-machine search interactions where only one party, either the user [18, 19, 22, 38, 53] or system [16], drives the search. In contrast, in *human-to-human* interactions, the participants in a conversation trade off control: usually all participants at least have the possibility of both asking and answering questions. To combine the information-theoretic benefits of [16] and the explorative nature of interaction of [6, 19], we propose a framework that allows the machine and human to alternate, depending on who can initiate more informative feedback.

2 Related Work

Attribute-based search. Prior work has explored the value of the fine-grained detail that attribute descriptions provide, by using attributes to initiate a search [38, 46] or provide iterative feedback on the results of a search system [16, 19, 27]. [18] browses the current search results, and can then provide a feedback statement of the form "The image I am looking for is more/less [attribute] than [this image in the results]." The choice of an attribute on which to comment is left to the user. This is helpful if the user is perceptive, or there are images which obviously differ from the user's desired content for particular attributes. On the other hand, browsing a set of images and choosing attributes is time-consuming for the user, as we find in experiments. [16] shows that given a limited budget of interactions that the user is willing to perform, more accurate search results can be achieved if the system asks the user questions of the form "Is the image you are looking for more/less/equally [attribute] than [this image]?" The chosen questions are those with high information gain. The disadvantage of [16] is that it limits the ability of the user to browse and explore the dataset space.

Sketch-based search. While attribute-based feedback is appropriate when the user can concisely describe what content they wish to find using words, some searches involve concepts which are purely visual. In our setting, we assume the user does not have a photograph of what they wish to find, so cannot directly do similarity-based search with a query image. However, the user does have a clear visual idea of what content they wish to find. Sketch-based search approaches allow the user to convey this visual idea to the system, via a sketch or drawing, which provides a complementary way of communication. The system can then extract features from this sketch and compare to the features of the images in a database [6, 34, 37, 54, 55]. We use a similar approach, but also propose to convert the sketch to an image using generative models. Other authors use generative learning to find a representation appropriate for cross-domain (sketch-to-image [30, 39, 40] or text-to-image [39]) search. We use sketch-based retrieval in a larger reinforcement learning framework that chooses which search interaction to propose (sketch, attribute-based feedback, or question-answering). Note that our focus is *not* in how we perform sketch-based retrieval, but rather *how to decide when* to request a sketch.

Interactive search. Rather than ask the user to issue a query and return a single set of results, we engage the user in providing interactive relevance feedback and show results after each round. This is a popular idea [4, 8, 9, 33, 56] whose key benefit is that incorrect predictions by the system can be corrected. We also adopt interactive search, but combine the advantages of free-form feedback and exploration with the information-theoretic benefits of actively querying for feedback [8], via reinforcement learning.

Active learning. In order to minimize the cost of data labeling, active learning approaches estimate the potential benefit of labeling any particular image, using cues such as entropy, uncertainty reduction, and model disagreement [11, 14, 36, 44, 47]. [2, 5, 41, 48] have

explored mixed initiative between user and system as well as reinforcement learning, for improving active learning at training time, in contexts other than image search. In contrast, we use reinforcement learning to select interactions at test time (during online search).

Reinforcement learning [15, 26, 45] has recently gained popularity for a variety of computer vision tasks, e.g. object [1, 24] and action detection [51]. The most related work to ours is [52] which also uses reinforcement learning to choose the type of feedback method for requesting feedback from the user. This approach considers query vector modification, feature relevance estimation, and Bayesian inference, as three possible feedback mechanisms. Neither of these allows the user to *comparatively* describe how the results should change (via attributes); instead, each image property is defined as desirable/undesirable. [19] show such binary feedback is inferior to comparative attribute feedback. Further, unlike [52], we consider both visual and textual feedback among the mechanisms presented to our users.

3 Approach

We develop an approach for interactive image retrieval, where the user can provide guidance to the system via two text-based and one sketch-based modalities, described below. The search scenario we envision is the following: The user has a clear idea of the exact target image they wish to find, but does not have that image in hand. Our system's goal is to determine which type of interaction to suggest to the user at any point in time.

3.1 Search setup and interactions

Interactions. The user can initiate a search with random images from the database, or ones that match a simple keyword query. Then the user can perform a combination of the following three types of feedback. First, the user can browse the returned images, and relate them to her desired target via attribute comparisons, e.g. "The person I am looking for is *younger* than this person," where "this person" is an image chosen from the returned results. Second, the system can ask the user a question, e.g. "Is the person you are looking for more or less chubby than this person?" Third, the user can draw a sketch to visually convey to the system their desired content. These search interactions are based on prior work [6, 16, 18, 19, 34, 54], and we learn how to combine them.

System interface. Our system is illustrated in Fig. 2, and it has three components: i) a target image, ii) user feedback using attributes or a sketch, and iii) current top images. User feedback is received in each iteration, and updates the top images.

Relevance models. After one of the three interactions is used and feedback from the user is received, the system must rank all database images by estimating their relevance using the feedback the user provided. For free-form attribute feedback and suggested question interactions, following [19], the relevance of a database image is proportional to the likelihood that it satisfies each attribute constraint, e.g. it is more shiny than a reference image. For sketch interaction, we "convert" the sketch to a photograph (i.e. we add color) using a conditional GAN [13]. An alternative is to directly learn a space whether sketches and images are aligned, and perform retrieval in this space; we show an experiment using this approach as well. Then, CNN features are extracted and we train a one-class SVM [35] whose output probabilities for each image are used to rank the images. The final relevance of an image is a product (multiplication) of all attribute-based and sketch-based relevance estimates.

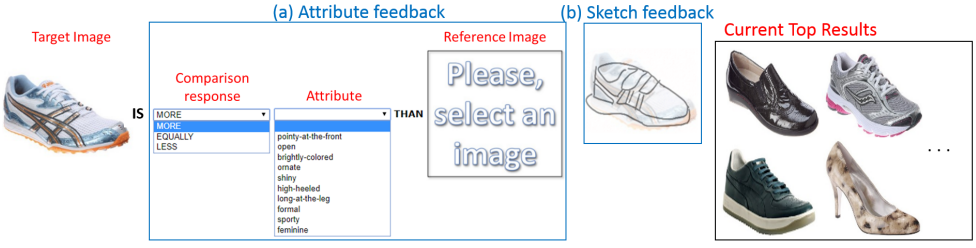

Figure 2: Image retrieval system setup. The system's goal is to find the target image. Users refine the image retrieval using an (attribute, reference image, comparison response) triplet or a sketch. User interactions are used to update the current top image results.

3.2 Reinforcement learning representation

We formulate the selection over search interactions as a Markov Decision Process composed of actions, states, and rewards, defined below.

Actions. We train a reinforcement learning algorithm to select one of three interactions for a given iteration. In order to train it, we require user selections of image-attribute pairs (the free-form feedback proposed in [18]), responses to attribute-based questions proposed in [16] (the more/less/equally value of a comparison between the target and reference image along a certain attribute dimension), and sketches (used for search in [6, 34, 54]). User selections are simulated by selecting an (image, attribute) pair that reduces the part of the multi-attribute space that needs to be searched in order to find the target image. In particular, our simulated users are given a subset of the attribute vocabulary[1], and a set of reference images. They are also given information about how many images in the database satisfy a given image-attribute constraint, e.g. how many images are "less chubby than [this person]," according to the system's model of "chubbiness." The simulated user then chooses the image-attribute pair that results in the smallest number of images satisfying the constraint. This simplifies search as only a few images remain relevant after each feedback constraint is given.

In terms of question responses, we also simulate users' feedback, similarly to [16], by adding Gaussian noise to the attribute model predictions, and choosing the more/less/equally response based on the difference in the attribute values predicted for the target image and the reference image which the system chose. The original method of [16] requires entropy computation, which is computationally expensive if it needs to be repeated many times, as we require for reinforcement learning. Hence, we use an ablation presented in [16] which performs similarly but is much faster. It uses the per-attribute binary search trees of [16] but alternates between attribute pivots in a round-robin fashion.

Sketches are simulated using edge maps [49] generated from the target image, similarly to [13]. We also show experiments using real human-drawn sketches. We then convert them to photographs using a GAN [13], and rank database images by their similarity to the photo, using the probabilities from a one-class SVM [35].

State. Let h_{+prox} and h_{-prox} be positive and negative proxy sets for the target image, defined as the five neighbors closest to the target (excluding the target itself), and five neighbors furthest from the target. We represent our state as (h_{top_ims}, h_{+prox}, h_{-prox}, $h_{actions}$), where h_{top_ims} is the history of top images (i.e. those ranked at the top in previous iterations),

[1]Since our simulated users receive system-level information as described next, allowing them to use the full vocabulary results in unrealistic alignment between the user's mental model and the system's predictions.

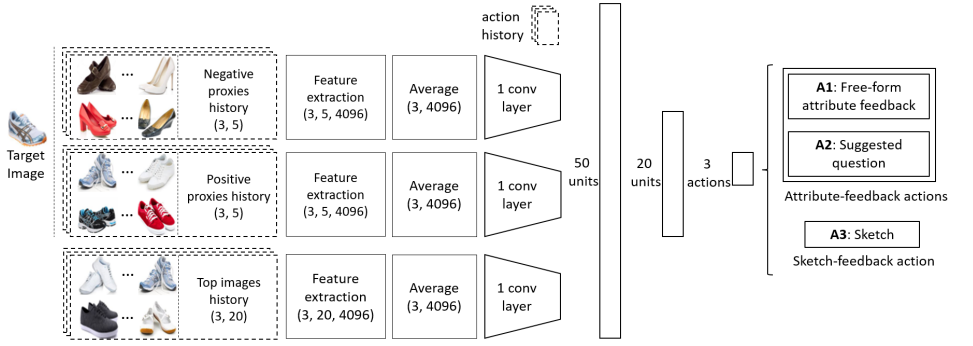

Figure 3: Architecture of our proposed Q-network. It receives histories of top-ranked images, positive and negative proxy images, and taken actions. It predicts the best action given a specific state. Inputs are denoted with dotted lines. Please see text for further explanation.

and $h_{actions}$ are the actions taken in previous iterations. Images are represented by features extracted from AlexNet [20], and actions by a 3-dimensional binary vector, where all values are zero, except the one corresponding to the taken action. We use a history size of 3.

Rewards. We would like that in each iteration, our top images become more and more similar to the target image (which is unknown to the system). We can measure this using two cues: distance to positive proxy images, and distance to negative proxy images. We encourage a decrement of the first distance, and an increment of the later distance. We do this using a reward function $r(s,s')$ which is evaluated when an action is performed and causes a transition from state s to state s'. Each state has associated top images (top_ims) and proxies ($+prox$ and $-prox$). We calculate the Euclidean distance d between (1) the average features of all top images and (2) the average features of the positive/negative proxy images. Then the function r is defined as:

$$r(s,s') = sign[d(top\_ims,+prox) - d(top\_ims',+prox)] + sign[d(top\_ims',-prox) - d(top\_ims,-prox)] \quad (1)$$

In other words, we want the distance of the top images to positive proxies to decrease, and distance to negative proxies to increase. One might think that using positive proxies is enough, however we prefer a more fine-grained representation. Both sets of proxies are helpful, especially at the beginning when the search space is large and could be misleading. For example, imagine a two-dimensional search space where $+prox = (4,1)$, $-prox = (1,4)$, $top\_ims = (3,3)$ and $top\_ims' = (2,2)$. Thus, $r(s,s') = sign(2-1.4) + sign(2-2.8) = 1-1 = 0$. We observe that decreasing the distance to $+prox$ does not necessarily enforce an increment on the distance to $-prox$, so we need to explicitly encourage this.

We also want to encourage that the sketch action is used only once. Hence, we assign a penalty of -1 if the sketch interaction is requested more than once.

3.3 Learning

The goal of our agent is to update the search results by selecting actions. There are many possible states, so using a transition matrix with all states and actions is not recommended. Also, our reward function is data-dependent (i.e. we use image ranking to calculate it). Q-learning [42], which receives a state and predicts the best action, is a good fit for our task. Our Q-learning agent aims to maximize the future discounted reward $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$ at each timestep t, where $r_{t'}$ is the reward at time t', T is the time when the search episode ends and γ is the discount factor. We maximize R_t learning a policy to select an action by $\pi(s) = argmax_a Q(s,a)$ at state s.

We approximate the Q function with a neural network, which is based on [1] and is depicted in Fig. 3. Our top images and proxies data uses the same convolution architecture composed of a convolutional layer with 8 filters of size 3x3 and a max-pooling layer. The outputs of the top images and proxies branches are concatenated with the history of actions, and projected using 3 fully-connected layers to generate action scores. We employ RELU activation for the convolutional and fully-connected layers. We employ convolutional layers in the top result image and proxies branches, because they capture information about image features and ordering. Our Q-network learning requires data in the form of $[s, s', a, r]$, which denotes current state, next state, action and reward; and aims to maximize the following loss, where V represents the true future discounted reward using r and s'.

$$L = \frac{1}{2} * [V - Q(s,a)]^2 \qquad V = r + \gamma * max_{a'} Q(s', a') \tag{2}$$

Our approach also considers replay-memory to collect many data instances as it is running. Each instance follows our previous format $[s, s', a, r]$. This information enriches our training data, and in each iteration, a random subset of this data is used for training. This procedure also removes short-term correlation between subsequent states, and makes our algorithm more robust and stable.

At initial stages of learning, random actions are beneficial so the agent can *explore* [42] and get information about the problem. Later this information is *exploited* to select actions. We generate random actions with probability decreasing from 1 to 0.1 as training progresses.

Implementation. We implemented the described network using the Theano [43], Keras [3] and DEER[2] frameworks. We use the RMSProp optimizer, a discount factor of 0.9, a learning rate of 1e-5, and 30 epochs. At the end of each epoch, the network was evaluated on a validation set, and the network that successfully completed more searches (i.e. found the target image in at most 10 iterations) over a validation set was selected for testing.[3]

4 Experimental Validation

Datasets. We use three datasets which have frequently been used for image search: Pubfig [21] with 11 attributes (e.g. smiling, rounded-face, masculine) and 769 images (after de-duplication); Scenes [29, 31] with 6 attributes (open, in perspective, etc.), and 2668 images; and Shoes [19] with 10 attributes (formal, high-heeled) and 12,807 images. We extracted fc6 deep features for Pubfig and Shoes; and fc7 features for Scenes as in [27]. To speed up the interaction of our reinforcement learning agent and the image retrieval system, we reduce the number of images to 1000 by clustering in the predicted attribute strengths space.

Evaluation protocol. For each dataset, we split the data in 70% for training, 10% for validation and 20% for testing. Our reinforcement learning approach uses the train and validation splits to learn to predict actions. To compare the methods more precisely, we tell the user which image to search for (*target image*). In each iteration, the user provides a comparison of the target and pivot/reference image, or a sketch of the target. We report percentile rank of the target, defined as the fraction of database images ranked lower than the target (in the range [0, 1], higher is better).

Baselines. We compare our reinforcement learning agent (RL) with three baselines:

- Whittle Search [19] (*WS*): In each iteration, users select a (reference image, attribute) and compare target and reference for the chosen attribute dimension ("more / less / equally"). The relevance of database images which satisfy this feedback increases.

[2] https://github.com/VinF/deer/

[3] For our Scenes dataset, the best model is acquired using percentile rank.

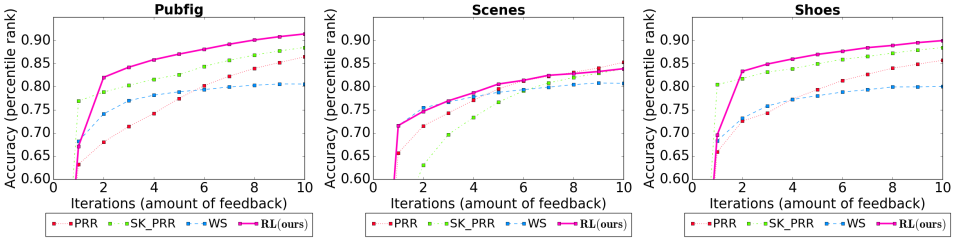

Figure 4: Percentile rank plots for Pubfig, Scenes, and Shoes. Our mixed-initiative RL agent outperforms the other baselines on Pubfig and Shoes, and performs competitively for Scenes.

Table 1: AUC for percentile rank curves from Fig. 4. Best scores are highlighted per dataset.

	PRR [16]	WS [19]	SK_PRR [16, 54]	**RL (ours)**
Pubfig	0.729	0.737	0.789	**0.810**
Scenes	0.741	0.741	0.699	**0.754**
Shoes	0.745	0.731	0.806	**0.810**
avg	0.738	0.736	0.764	**0.791**

- Pivot round robin [16] (*PRR*): In contrast to *WS*, *PRR* provides an (image, attribute) pair, and users only need to provide a more/less/equally response.
- Sketch retrieval [54] + pivot round robin [16] (*SK_PRR*): In the first iteration, we ask the user for a sketch of the target image, then attribute questions follow.

4.1 Simulated experiments

We simulate ten users as described in Sec. 3.2. Fig. 4 shows percentile rank curves for our proposed method and the three baselines. For the Pubfig and Shoes datasets, our reinforcement agent outperforms the baselines with a large margin. However, for Scenes, the improvement is reduced. Hence, we also inspect AUC for the percentile rank curves in Table 1. We observe that our approach outperforms all baselines for all datasets.

We observe that *WS* achieves high accuracy at the very first iterations and outperforms the *PRR* method. This follows the intuition that with *WS*, which allows *exploration*, the user can provide more meaningful feedback that reduces the search space, in contrast to earlier stages of the *PRR* method. However, in later iterations, *PRR* improves accuracy because it follows a binary-search strategy iterating over all attributes. Hence, *PRR* ensures diversity of feedback, in contrast to *WS* which can be repetitive. *SK_PRR* outperforms *WS* and *PRR* in two of the three datasets. Incorporating sketch feedback enhances the informativeness of attribute-based feedback, except for Scenes. A possible explanation is that scenes are more complex than faces and shoes, as they contain more than one object. This prevents our GAN from being able to generate good photo versions of our scene edge maps (see Fig. 6).

4.2 Live experiments

In order to run a user study, we develop a web interface that implements our three baselines, and our approach. Our approach queries the next action using a REST API[4], that connects to our web interface. For this experiment, we replace sketch-to-photo coloring with sketch retrieval [54] directly comparing features of the sketches to images, as an alternative to get diverse and realistic images. This helps avoid GPU memory problems due to multiple queries for the GAN conversion. We only conduct an experiment for the Shoes dataset because we

[4] https://blog.keras.io/building-a-simple-keras-deep-learning-rest-api.html

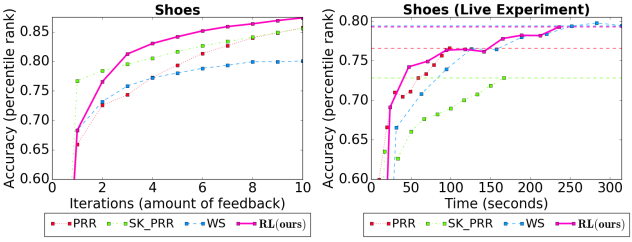

Figure 5: Percentile rank plots for Shoes dataset with simulated (left) and live users (right). Both experiments use sketch retrieval. Live user experiment results are plotted over time.

Figure 6: Sample sketch-to-photo colored images for Pubfig (columns 1-3), Shoes (columns 4-6), and Scenes (columns 7-9). Each column denotes a different category.

did not find any appropriate sketch annotations for training, for Faces[5] and Scenes. The result for simulated users (Fig. 5 left) in this setting is similar to our previous findings: our approach outperforms all baselines.

We recruit workers on Amazon Mechanical Turk and university students to search for 100 images. Each participant searches for one image, which is the same for the four methods. We request Turkers with location in the US, HIT approval rate greater or equal to 98%, and at least 1000 approved HITs. We remove blank and careless sketches (i.e. just straight lines), which results in 88 searches. The results are shown in Fig. 5 (right). Because different interactions require very different amount of user time (*PRR*: 9s, *SK_PRR*: 16s, *WS*: 31s, and *RL*: 23s), we plot time on the x-axis, multiplying each iteration by the number of seconds it requires. We show horizontal lines with the final (highest) percentile rank a method achieves. Our *RL* method and *WS* achieve similar peak performance (79.2% for *RL* and 79.4% for *WS*) while *PRR* only achieves 76.6% at the end of 10 iterations. However, our method achieves higher performance early on; the curve for *RL* is higher than that for *WS* until about 230s of user time spent, then performance is similar. Thus, our approach achieves higher performance in a smaller amount of time, compared to the strongest baseline *WS*.

We include sketches provided from our live users in our supplementary material.

4.3 Qualitative Results

In order to understand the success of our approach, we visualize some of the generated colored pictures (Fig. 6), and we also show the predicted actions on our test split (Fig. 7).

For our sketch-to-photo generated images, we observe that the most realistic ones correspond to Pubfig, then Shoes, and finally Scenes. This order also corresponds with the performance of our method in terms of percentile rank, where Pubfig and Shoes achieve the best performance. Scenes did not benefit from the generated images as much because they are not realistic and present poor quality. However, our GAN intuitively associates brown color to coast (Fig. 6, column 7). Similarly, it learns green color for forest (Fig. 6, column 9). We add more visualizations in our supplementary material including edge maps.

We also want to understand our mixed-initiative RL agent, so we count its predicted actions per iteration in Fig. 7. Note that the action at each iteration is chosen by our agent.

[5]Fine-grained sketches are available but most real users cannot provide such high-quality sketches.

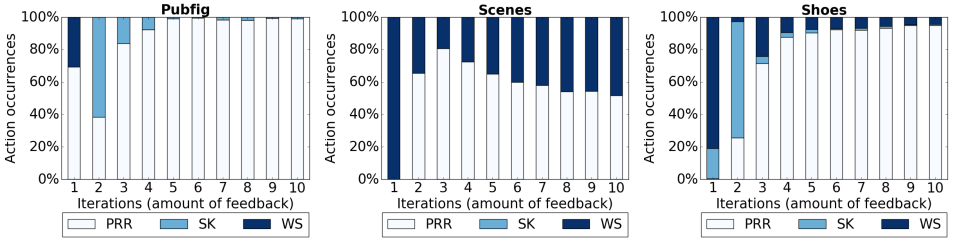

Figure 7: Percentage of actions predicted by our approach in the test set.

Partial not available history information is filled with 0s. For Pubfig and Shoes, we observe that *SK* (sketch) and *WS* actions are mainly performed in iterations 1 and 2, because these are the *exploration*-like actions. Then, after iteration 3, the *PRR* is the most common one. Once the most beneficial human knowledge is acquired, having a computer suggest feedback (in the form of questions) helps reduce the search space the fastest. Hence, our agent learned to prioritize human-initiated feedback early on, and complement it with machine-initiated feedback in later iterations. For Scenes, our method prioritizes *WS* early on and *PRR* later, and ignores *SK* because it does not provide much benefit.

5 Conclusion

We explored the problem of selecting interactions in a mixed-initiative image retrieval system. Our approach selects the most appropriate interaction per iteration using reinforcement learning. We find that our model prefers human-initiated feedback in former iterations, and complements it with machine-based feedback requests (e.g. questions) in later iterations. We outperform standard image retrieval approaches with simulated and real users. For future work, we plan to learn personalized reinforcement agents that follow the individual attribute interpretations [17, 28], visual perception and sketching style of users.

Acknowledgment. This research was funded by a University of Pittsburgh Central Research Development Fund (CRDF) grant and an NVIDIA hardware grant. This research also used the Extreme Science and Engineering Discovery Environment (XSEDE) and the Data Exacell at the Pittsburgh Supercomputing Center (PSC), supported by National Science Foundation grants ACI-1053575 and ACI-1261721. Finally, we thank Ray Mooney for the original idea of developing a mixed-initiative framework for attribute-based search, and we are grateful to our search experiment participants for their time and effort.

References

[1] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[2] Maya Cakmak and Andrea L Thomaz. Mixed-initiative active learning. In *International Conference on Machine Learning (ICML)*. PMLR, 2017.

[3] François Chollet. Keras, 2015.

[4] Ingemar J Cox, Matthew L Miller, Stephen M Omohundro, and Peter N Yianilos. Pichunter: Bayesian relevance feedback for image retrieval. In *International Conference on Pattern Recognition (ICPR)*. IEEE, 1996.

[5] Sandra Ebert, Mario Fritz, and Bernt Schiele. Ralf: A reinforced active learning formulation for object class recognition. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[6] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *Transactions on visualization and computer graphics (TVCG)*, 2011.

[7] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing Objects by Their Attributes. In *Computer Vision and Pattern Recongition (CVPR)*. IEEE, 2009.

[8] Marin Ferecatu and Donald Geman. A statistical framework for image category search from a mental picture. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2009.

[9] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. Cueflik: interactive concept learning in image search. In *SIGCHI Conference on Human Factors in Computing Systems (SIGCHI)*. ACM, 2008.

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[11] Yuhong Guo and Russell Greiner. Optimistic active-learning using mutual information. In *International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann Publishers Inc., 2007.

[12] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[14] Christoph Kading, Alexander Freytag, Erik Rodner, Paul Bodesheim, and Joachim Denzler. Active learning and discovery of object categories in the presence of unnameable instances. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[15] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research (JAIR)*, 1996.

[16] Adriana Kovashka and Kristen Grauman. Attribute pivots for guiding relevance feedback in image search. In *International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[17] Adriana Kovashka and Kristen Grauman. Discovering attribute shades of meaning with the crowd. *International Journal of Computer Vision (IJCV)*, 2015.

[18] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Image search with relative attribute feedback. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[19] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Interactive image search with relative attribute feedback. *International Journal of Computer Vision (IJCV)*, 2015.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., 2012.

[21] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. In *International Conference of Computer Vision (ICCV)*. IEEE, 2009.

[22] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Describable visual attributes for face verification and image search. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2011.

[23] Christoph Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to Detect Unseen Object Classes By Between-Class Attribute Transfer. In *Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2009.

[24] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. Reinforcement learning for visual object detection. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[25] Austin Meyers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, and Kevin P. Murphy. Im2calories: Towards an automated mobile vision food diary. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[27] Bhavin Modi and Adriana Kovashka. Confidence and diversity for active selection of feedback in image retrieval. In *British Machine Vision Conference (BMVC)*, 2017.

[28] Nils Murrugarra-Llerena and Adriana Kovashka. Learning attributes from human gaze. In *Winter Conference of Computer Vision (WACV)*, 2017.

[29] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*, 2001.

[30] Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Cross-domain generative learning for fine-grained sketch-based image retrieval. British Machine Vision Conference (BMVC), 2017.

[31] Devi Parikh and Kristen Grauman. Relative attributes. In *International Conference of Computer Vision (ICCV)*. IEEE, 2011.

[32] Nikita Prabhu and R. Venkatesh Babu. Attribute-graph: A graph based approach to image ranking. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[33] Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *Transactions on Circuits and Systems for Video Technology (TCSVT)*, 1998.

[34] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *Transactions on Graphics (TOG)*, 2016.

[35] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computing (NC)*, 2001.

[36] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Workshop on Computational Learning Theory (WCLT)*. ACM, 1992.

[37] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Data-driven visual similarity for cross-domain image matching. *Transactions on Graphics (TOG)*, 2011.

[38] Behjat Siddiquie, Rogerio S Feris, and Larry S Davis. Image ranking and retrieval based on multi-attribute queries. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011.

[39] Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy Hospedales. Fine-grained image retrieval: the text/sketch input dilemma. In *British Machine Vision Conference (BMVC)*, 2017.

[40] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketchbased image retrieval. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[41] Jina Suh, Xiaojin Zhu, and Saleema Amershi. The label complexity of mixed-initiative classifier training. In *International Conference on Machine Learning (ICML)*. IEEE, 2016.

[42] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 2011.

[43] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, 2016.

[44] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research (JMLR)*, 2001.

[45] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Conference on Artificial Intelligence (AAAI)*. AAAI, 2016.

[46] Daniel A Vaquero, Rogerio S Feris, Duan Tran, Lisa Brown, Arun Hampapur, and Matthew Turk. Attribute-based people search in surveillance environments. In *Winter Conference on Computer Vision (WACV)*. IEEE, 2009.

[47] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision (IJCV)*, 2014.

[48] Steven A Wolfman, Tessa Lau, Pedro Domingos, and Daniel S Weld. Mixed initiative interfaces for learning tasks: Smartedit talks back. In *International Conference on Intelligent User Interfaces (IUI)*. ACM, 2001.

[49] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[50] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[51] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[52] Peng-Yeng Yin, Bir Bhanu, Kuang-Cheng Chang, and Anlei Dong. Integrating relevance feedback techniques for image retrieval using reinforcement learning. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2005.

[53] Aron Yu and Kristen Grauman. Just noticeable differences in visual attributes. In *International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[54] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen Change Loy. Sketch me that shoe. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[55] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural network that beats humans. *International Journal of Computer Vision (IJCV)*, 2017.

[56] Xiang Sean Zhou and Thomas S Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems (MS)*, 2003.