Syntharch: Interactive Image Search with Attribute-Conditioned Synthesis

Zac Yu* Adriana Kovashka
University of Pittsburgh
{zac.yu,kovashka}@pitt.edu

Abstract

The use of interactive systems has been found to be a promising approach for content-based image retrieval, the task of retrieving a specific image from a database based on its content. These systems allow the user to refine the set of results iteratively until the target is reached. In order to proceed with the search efficiently, conventional methods rely on some shared knowledge between the user and the system, such as semantic visual attributes of the images. Those approaches demand the images to be semantically labeled and introduce a semantic gap between the two parties' understanding. In this paper, we explore an alternative approach to interactive image search where feedback is elicited exclusively in visual forms, therefore eliminating the semantic gap and allowing for a generalized version of the method to operate on unlabeled databases. We present Syntharch, a novel interactive image search approach which uses synthesized images as options for feedback, instead of asking textual questions to gain information on the relative attribute values of the target image. We further demonstrate that by using synthesized images rather than real images retrieved from the database as feedback options, Syntharch causes less confusion to the user. Finally, we establish that our proposed search method performs similarly or better in comparison to the conventional approach.

1. Introduction

In recent years, the rapidly growing volume of searchable images calls for more and more efficient methods to retrieve one target image from a large pool of images. The task has been formalized as content-based image retrieval (CBIR) and the techniques have been implemented as applications across multiple domains, including web image search [19, 3], e-commerce [31, 43], health care [10, 42], etc. The search focuses on the visual content rather than textual metadata such as labels, description, or the context; however, the search query generated by the user usually takes a textual form. Therefore, the challenge is to establish a mapping between the user's high-level concept and the machine's low-level representation of the image; this is sometimes referred to as a "semantic gap."

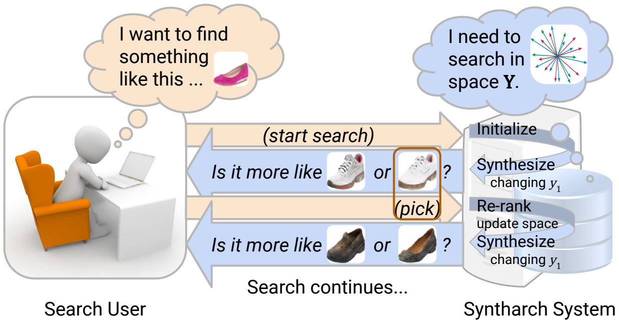

Figure 1. Syntharch elicits attribute feedback by synthesizing pairs of options and then employs user responses to re-rank images in the database and to refine the search space.

To complicate matters further, when a large number of similar images are present in the database, more fine-grained queries are needed in order to reach the target image. A classical approach for this refining process is to allow the user to interact with the retrieval system in order to provide additional information regarding the target image iteratively [35]. For each iteration of the interactive search, the system accepts some form of feedback, but its efficiency is still hindered by the semantic gap.

Relative attributes can help ameliorate this challenge by transforming the machine's low-level image representation to high-level semantic attributes. These can be expressed in textual form and understood easily by the user [23, 15, 17]. In particular, the user can provide feedback on how some attribute of an image differs from that of the target image. However, this approach introduces the burden for the system to understand high-level semantic attributes known to the user. This presents two drawbacks. First, the verbal representations of the attributes (*e.g.* how "ornamental" and "formal" a piece of apparel is) can be ambiguous and might vary among users and cause confusions [16]. Sec-

*Now at Google. This work was completed prior to joining Google.

ond, attributes may be correlated as discussed in [4], so when asked to provide feedback about a particular image, the user may have trouble decorrelating these attributes and hence may provide feedback that is confusing to the system.

To address these two challenges of interactive image search, we propose Syntharch, a new way to close the semantic gap using *visual-only* feedback on high-level properties of *synthetically generated* images. We first produce a multidimensional space $\mathbf{Y}$, and generate images that correspond to points in this space. Each dimension corresponds to an image property akin to an attribute, but not necessarily human-nameable. We show pairs of synthesized images that differ in attribute values are presented to the user as options, and ask her to choose one which more closely resembles the user's desired target (see Figure 1). Using the obtained feedback, we narrow down the search region within $\mathbf{Y}$. We re-rank the images, and use this narrower region to generate the next two images for feedback.

In contrast to prior attribute-based work [17, 22], our method does *not* rely on textual requests for feedback, *e.g.* "Is the image you are looking for more or less *pointy* than this one?" This means that the dimensions for feedback need not be nameable, which addresses the first challenge above. In addition, rather than using images from the dataset for feedback, we use synthetically generated images for which we have direct control over the properties they exhibit. By only changing one dimension at a time (and keeping others fixed) in the generated images on which we solicit feedback, we cope with the second challenge above.

In this project, we use a vocabulary of attributes as the space $\mathbf{Y}$. Each image is represented as a normalized attribute vector in this space, along with a latent vector from space $\mathbf{Z}$. We train a generator that interpolates smoothly along a spectrum of attribute values. Each attribute vector, along with a latent vector describing low-level characteristics of a particular image A, can be transformed into another image B using this generator. This new image B observes the low-level visual appearance of the original image A, except for attributes which may have been modified. Once the attribute space and the generator are learned, we proceed with the interactive search and use synthesized images produced by the generator as options to gain relevance feedback and to approach to the target image in the attribute space. During the search, we maintain a search space to determine the attribute vectors for the A-B image pairs. Each choice within a pair indicates that the target is closer to the chosen image than the other in $\mathbf{Y}$. This information, in turn, allows us to update the search space accordingly.

We show our approach improves search effectiveness compared to two baselines, which either use only real images for feedback, or use a different mechanism to select images for feedback. In the long term, Syntharch opens the potential for this interactive image search approach to operate on unlabeled databases, given a method to learn discriminative relative attributes in an unsupervised fashion.

2. Related Work

Interactive Image Search The image search system we propose is an interactive system, meaning that instead of providing one fixed set of results given a search query, the system allows for user interactions to refine the results, improving the accuracy of the search and correcting faulty assumptions over time. This idea has been studied for over two decades and is a popular approach for the content-based image retrieval task [34, 5, 35]. Early approaches, most notably those adopted by web image search engines, utilize low-level features such as color, dimension, and shape as image descriptors [34, 32, 19]. In recent years, relevance feedback has been shown to be more effective and accommodating to high-level concepts [29, 5, 6, 2, 17, 41]. By incorporating relevance feedback, search systems can iteratively gain information on the user's desired target results, and correct mistakes due to the semantic gap.

Attribute-Based Search Relative attributes are used to facilitate the interactive search by allowing comparative feedback on specific human-nameable properties [33]. For example, instead of providing binary feedback of whether some given reference is relevant or not, one can express their target image as being "more ornamental" or "less formal" than a reference image [17]. To improve efficiency, [15] proposes searching with a binary search tree for each relative attribute. While using relative attributes to elicit user feedback has been a popular approach, all previous work relying on semantic visual attributes uses some textual form of feedback, in closed-form, free-form, or natural language. Our Syntharch approach expresses relative attributes solely in visual forms, which helps alleviate the semantic gap. Recently, there have also been explorations with a larger variety of feedback forms in addition to attributes, including sketches [22] or natural-language dialog [8]. However, sketches are generally used once to initiate a search and do not enable interactive search, while dialog incurs additional overhead to learn the nuances of language.

Conditional Image Synthesis Recently, synthesis of realistic images has been greatly empowered by deep generative models including variational autoencoders (VAE) [14, 18, 36] and generative adversarial networks (GAN) [7, 27, 21, 9, 30]. Conditional generative networks (CGAN) enable modulation of the output image based on parameters including text [21], images [9, 11], and attribute values [36, 20, 12, 30, 40]. In particular, [36] suggests using a variational auto-encoder to estimate the posterior distributions of the disentangled foreground and background image to generate the composite full image, and [20] incorporating an attribute encoder at training time to allow generating variances of an image with controlled attribute values.

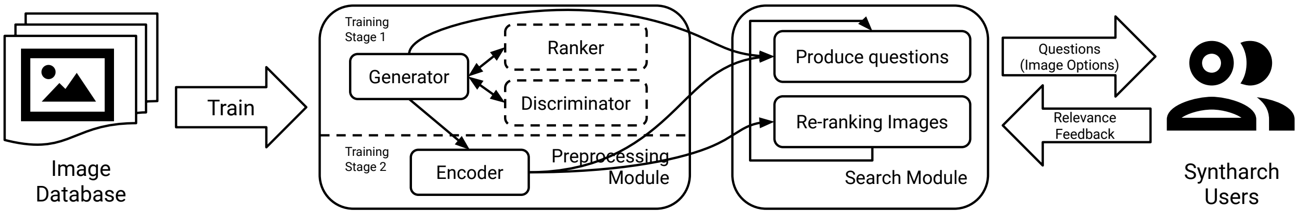

Figure 2. Syntharch preprocesses each image database with a two-stage training, which enables the interactive search.

Other approaches [12, 30] train adversarial networks and use a conditional vector as an additional input of the generator to control the attributes. In Syntharch, the conditional image synthesis module based on RankCGAN [30] is integrated into a preprocessing module to construct a multidimensional attribute space for image synthesis. Concurrent work [40] uses image generation to request labels for training attribute models, outside of any search context; in contrast, we solicit feedback on generated images specifically to improve search results, not attribute models.

Image Editing The task of image editing is an extension to conditional image synthesis with the capability to "invert" the synthesis process. VAE naturally comes with an encoder (a variational inference network) that can be used to estimate the noise vector in some latent representation space. To control the synthesis result, we can simply modulate the noise vector accordingly. GAN, as proposed originally [7] lacks the capability to project real images onto the latent space for reconstruction, thus researchers have built encoders on top of the GAN architecture for tasks such as disentangling latent factors of 3D view synthesis [37] and text to image synthesis [28]. [25] presented Invertible Conditional GAN (IcGAN), an in-depth analysis of using encoders to inverse the mapping of deep CGANs. Building on top of a conditional DCGAN, they introduced encoder networks that convert images to latent variables, trained by random datasets created with the generator. The encoders therefore allow reconstruction and modification of real images. In Syntharch, the encoder we built for recovering the latent noise vector and manipulating image attribute values is based on the network proposed in IcGAN.

3. Approach

We introduce Syntharch (**Synth**esis + Se**arch**), an interactive image search system which leverages conditional image synthesis for collecting more informative feedback. As shown in Figure 2, the system is comprised of two modules: a preprocessing module that performs two-stage training for every image database, and a search module that interacts with a user who wants to retrieve an image from a preprocessed database. The first stage of the preprocessing module trains the generator. The output of this generator is used during the second stage to train an encoder which maps an image to an estimated representation in the latent space. The search module produces the questions (feedback requests) to pose to the user using the learned generator and encoder. Then, using the collected responses, it ranks the images to produce the search results.

3.1. Generator Networks

The generator network is adapted from RankCGAN [30], which is a combination of a conditional GAN (CGAN) and a RankNet [1]. The generator, the ranker, and the discriminator are trained simultaneously. The CGAN is composed of two neural networks, generator and discriminator. The generator $G(z, y)$ takes in a latent noise vector $z \backsim \mathcal{N}(0, I)$ of length 100 and an attribute vector $y \backsim \mathcal{U}(-1, 1)$ whose length equals the number of attributes in the database. The concatenation of the two vectors serves as the input to the network. The output tensor (synthesized image) has dimension of 3x64x64. The discriminator network $D(x)$ takes in an image tensor x and makes a prediction indicating if the input image is real. The output layer of D is a single scalar in $(0, 1)$ capturing the likelihood of input image x being real.

The second component of our design is RankNet [1] which uses gradient descent methods for learning ranking functions. These are then used to estimate pairwise comparisons of image attributes and to help in decorrelating the attributes by splitting them into different dimensions in the attribute space $\mathbf{Y}$. In particular, we use binary labels (greater than v.s. less than[1]) to represent pairwise attribute comparisons of images in the database. We benefit from using pairwise comparisons as opposed to exact values when regulating the ranker because it allows us to formulate the RankNet loss as a binary cross-entropy loss, similar to that of the discriminator:

$$\mathcal{L}_R(x_i^{(1)}, x_i^{(2)}, c_i) = -c_i \log{(p_i)} - (1 - y_i) \log{(1 - p_i)}\,, \tag{1}$$

where p_i is the posterior based on the estimated rank score

[1] The equal case, which rarely occurs, can be combined with either inequality of the dichotomy.

$$p_i = \text{sigmoid}\left(R\left(x_i^{(1)}\right) - R\left(x_i^{(2)}\right)\right), \qquad (2)$$

and c_i is the binary comparison result either given as a sample label (for real images) or inferred from the y values used for synthesis (for images produced by the generator).

The particular RankNet we use shares the same structure as the discriminator, except for the sigmoid function not being applied to the output layer as we only care about the pairwise ranking orders. A dedicated ranking layer is appended to the last hidden layer in parallel for each attribute.

In summary, the RankCGAN networks give us the capability of learning a multidimensional attribute space for image synthesis. With a fixed z, by modifying the y in the attribute space, we can generate images with different degrees of attribute expression. However, the real images from the search database are not yet mapped onto this space. In particular, although the ranker does provide a form of attribute-value representation, because the training only optimizes the ranking order of the prediction in pairs, its output cannot be directly mapped onto distribution $\mathcal{U}(-1, 1)$ of the attribute vector space. This can be addressed by building an encoder, as presented in Section 3.2.

3.2. Encoder for Image Editing

Given an image, the encoder proposed in IcGAN [25] can be used to approximate the latent noise vector z and the attribute vector y. Specifically, we learn two encoders E_z and E_y, for estimating z and y respectively. At training time, we use the generator to create a dataset of synthesized images with uniformly distributed z and y labels. Then, to optimize the encoder approximations, we learn the weights which minimize the mean squared error (MSE) loss of

$$\mathcal{L}_{E_z}(x) = \mathbb{E}_{z \backsim p_z, y' \backsim p_y} ||z - E_z(G(z, y'))||_2^2 \qquad (3)$$

$$\mathcal{L}_{E_y}(x) = \mathbb{E}_{z' \backsim p_z, y \backsim p_y} ||y - E_y(G(z', y))||_2^2. \qquad (4)$$

The encoder has four hidden convolutional layers, each applied with batch normalization and ReLU. The last convolutional layer is flattened, followed by two linear transformations which outputs an estimated vector of y or z.

The learned encoders, together with their generator counterparts, allow us to reconstruct images and further modify their attributes, as shown in Figure 3. We observe that while not all the details (*e.g.* colors and fine patterns) are fully preserved in the reconstructed images, the general shape and the style, as dictated by the labeled attributes of the dataset, are similar in every pair. To edit the attribute values of an image x, we first obtain the estimated vectors $y = E_y(x), z = E_z(x)$ from the encoder. Then, while fixing z, we modify y to y' as described below, then obtain the edited image $x' = G(z, y')$.

Figure 3 shows some of the editing results. Each row is a group that shows the original image x_{ori}, the reconstructed image $x_{rec} = G(E_z(x_{ori}), E_y(x_{ori}))$, followed by four edited images, each of which has the attribute value $y'^{(m)}$ (dimension m of y') incremented by 0.5 from $E_y(x_{ori})^{(m)}$. As an example, in the first group, E_y approximated the attribute value $y = [0.1536, -0.0565, 0.7329, 0.0863]$, where each dimension corresponds to an attribute: for the UT-Zap50K dataset, they are *open*, *pointy*, *sporty*, and *comfort*. The attribute vector used for generating the first edited image labeled "open $+0.5$" is therefore $y' = [\mathbf{0.6536}, -0.0565, 0.7329, 0.0863]$.

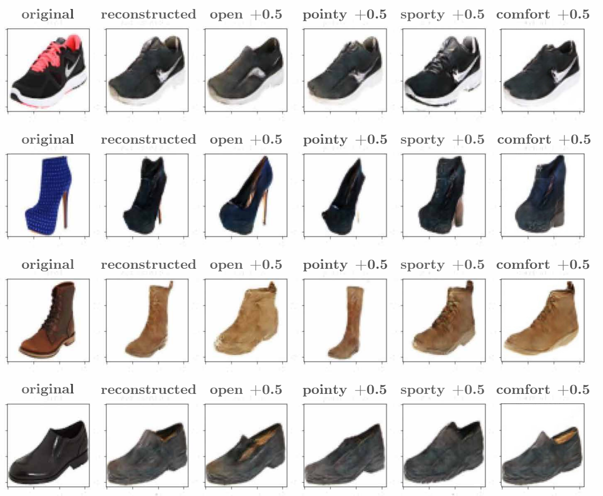

Figure 3. Image editing using the encoders and the generator.

3.3. Range-Based Search

During the search, we maintain an active search range $r_m \subseteq [-1, 1]$ for each dimension m of the attribute vector y. Initially, all ranges are set to $[-1, 1]$ since $y \backsim \mathcal{U}(-1, 1)$. As the user provides relevance feedback, the ranges are updated and they are used to determine the y' vector for synthesizing the feedback options. As shown in Figure 1, each question asked by Syntharch comprises two generated images. For each question, we want to elicit information regarding a specific attribute, an idea inspired by Attribute Pivots [15]. In each pair of images, every attribute m except for the attribute n we are querying is set to the center value of its corresponding range, i.e.

$$y_1'^{(m)} = y_2'^{(m)} = r_m^{(1)} + (r_m^{(2)} - r_m^{(1)})/2. \qquad (5)$$

For the attribute n, we divide the range to four equally-spaced segments and set the attribute value of one of the images to be at the $1/4$ of the range, while that of the other to be at the $3/4$ of the range, i.e.

$$y_1'^{(n)} = r_n^{(1)} + (r_n^{(2)} - r_n^{(1)})/4 \qquad (6)$$

$$y_2'^{(n)} = r_n^{(1)} + (r_n^{(2)} - r_n^{(1)}) \cdot 3/4. \qquad (7)$$

We pick $1/4$ and $3/4$ here because they are the centers of the two evenly divided portions of the original range. For example, at some stage of the search of a database with three attributes, if the ranges are $r_1 = [-0.5, 0.7], r_2 = [0.2, 0.8]$, and $r_3 = [0.2, 0.6]$, and that if we want to collect relevance feedback regarding the second attribute, then the attribute vectors are $y'_1 = (0.1, \mathbf{0.35}, 0.4), y'_2 = (0.1, \mathbf{0.65}, 0.4)$. Notice that the attribute values for the first and the third dimensions are intentionally kept the same so that the degree of attribute expression for those two are controlled; at the same time, for the second dimension, the values are precisely at $1/4$ and $3/4$ of the range. By controlling attribute values in all dimensions except for the one we are querying, when comparing the image options, the user will be more likely to provide informative feedback regarding to that particular attribute. Importantly, this type of fine-grained control is empowered by our image synthesis idea.

Meanwhile, to generate an image, we also need the latent vector z. To ensure that the synthesized results are realistic, we search for the real image x_{ref} from our database whose estimated attribute vector $y_{ref} = E_y(x_{ref})$ is the closest (with the least Euclidean distance) to the center of the current attribute search space, which also happens to be the average of the attribute vectors, i.e. $(y'_1 + y'_2)/2$ We then use $z_{\text{ref}} = E_z(x_{ref})$ as the latent noise vector to synthesize both images. An alternative approach would be to find x_{ref1} and x_{ref2} where their estimated attribute vectors are the closest to y'_1 and y'_2 respectively, and then use their estimated z-vectors for the synthesis. However, because we want to control the options visually so that they only differ in the attribute expression, we need to fix the noise vector. Therefore the two image feedback options are $x_1 = G(z_{\text{ref}}, y'_1)$ and $x_2 = G(z_{\text{ref}}, y'_2)$.

Whenever the user answers a question, we can infer from their choice the possible range of some attribute of the target image. Using the y'_1 and y'_2 values from the example above, if the user chooses x_1 over x_2, then we know that for the second attribute, the target value is likely closer to $r_2^{(1)} = 0.15$ than $r_2^{(2)} = 0.45$, and therefore the range can be reduced from $[0, 0.6]$ to $[0, 0.3]$. If we reduce the search range in this fashion, we are essentially performing a binary search on the attribute value range, similar to [15]. However, the performance would then be heavily affected by any mistakes (noises) in the selection process. To remedy that, we need to add in some tolerance: in the example above, instead of lowering the upper bound to the middle point (i.e. center of the range, $r_m^{(1)}/2 + r_m^{(2)}/2$), we want to pick a value between $1/2$ (middle point) and $3/4$ (selected option) of the range to balance range reduction speed and noise-induced variance. For Syntharch, we decided to use the value $2/3$ (i.e. $2r_m^{(1)}/3 + r_m^{(2)}/3$). Conversely, in the event of choosing x_2 over x_1, we raise the lower bound to be at $1/3$ of the range (i.e. $r_m^{(1)}/3 + 2r_m^{(2)}/3$). In the example, we would lower the upper bound to 0.4, i.e. reducing the search range for attribute 2 from $[0, 0.6]$ to $[0, 0.4]$.

In order to elicit feedback from different attributes, we use the round-robin approach, suggested by [15], to request feedback responses for each attribute one-by-one and to reduce the search range in all the dimensions iteratively.

3.4. Relevance Prediction

The objective of Syntharch is to retrieve the most relevant results based on the query. For the set of relevance feedback $\mathcal{F} = \{(r_m, f)_k\}_{k=1}^T$ collected during T search iterations, where r_m is the search range of attribute m and $f \in \{1, 2\}$ denotes either x_1 or x_2 was selected to be more similar to the target image, we want to produce a ranking of the database images x_i according to their relevance.

For that, we use a probabilistic-based relevance prediction model derived from [15] which further allows for some mistakes in relevance feedback in addition to our relaxed binary search constraint. The model can be formulated as the following: given the set $\mathcal{F}$, for each image x_i, we want to compute its probability of relevance $P(\text{relevant} \mid x_i, \mathcal{F})$. Let $S_{k,i} \in \{0, 1\}$ represent whether image x_i satisfies the binary search constraint in the k-th iteration of feedback. Specifically, if $f = 1$, then $S_{k,i} = 1$ if and only if $y_i^{(m)} < 2r_{m,k}^{(1)}/3 + r_{m,k}^{(2)}/3$. Similarly, if $f = 2$ then $S_{k,i} = 1$ if and only if $y_i^{(m)} > r_{m,k}^{(1)}/3 + 2r_{m,k}^{(2)}/3$. We can now express the probability of relevance for each image x_i as a sum of log probabilities,

$$P(\text{relevant} \mid x_i, \mathcal{F}) = \sum_{k=1}^{T} \log P(S_{k,i} = 1 \mid x_i). \tag{8}$$

Then we can use Platt's method [26] to estimate the probabilities with the following transform, that the log probability $\log P(S_{k,i} = 1 \mid x_i)$ equals

$$1 - \frac{1}{\exp(\alpha_m)\left(y_i^{(m)} - \left(2r_{m,k}^{(1)}/3 + r_{m,k}^{(2)}/3\right) + \beta_m\right)} \tag{9}$$

if $f = 1$, and

$$\frac{1}{\exp(\alpha_m)\left(y_i^{(m)} - \left(r_{m,k}^{(1)}/3 + 2r_{m,k}^{(2)}/3\right) + \beta_m\right)} \tag{10}$$

if $f = 2$, where α_m and β_m are learned from the pairwise comparison labels as well as the output of E_y on all images.

We run the relevance prediction model on all images after each iteration and sort the images by their probability of relevance to get our search results.

4. Experimental Validation

To evaluate how Syntharch's contribution advances prior art for interactive image search, we set up a user study.

4.1. Dataset

We evaluate Syntharch with the UT-Zap50K dataset [38, 39] consisting of 50,025 catalog images of shoes with 4 relative attributes labels: open, pointy, sporty, and comfortable. The attribute labels are provided in the form of 6,751 ordered pairs. Each pair label contains two image indices i, j for an attribute dimension m, indicating that x_i has a stronger strength in attribute m compared to x_j.

4.2. Metric

Similar to previous work [15, 17, 22, 8], we quantify the search performance (accuracy) by the percentile rank of the target image's probability of relevance, as given by the method described in Section 3.4, over time (iterations). The percentile rank is defined as the percentage of images in the search database that are ranked lower than the target image in the search results. Therefore, the higher the percentile rank, the closer the target image is to the top of the search results, and the more accurate the search results are. After each iteration of every search session, the current target percentile rank is recorded. By the end of all search sessions, the average percentile rank is aggregated for each search iteration for each search method.

4.3. Setup

Each experiment session contains 10 random search targets, and we run 30 search sessions in total. 10 of the search sessions use the Syntharch method, 10 of them using the alternative (baseline) method described in Section 4.5 and the remaining 10 using the method in Section 4.6. The order of the 30 search sessions (and consequently, that of the targets) are randomized. All experiment participants are instructed to perform the same task: for each search iteration, given a target image and two option images x_1 and x_2, select the option between the two that is closer to the target. We tell users which image to search for, in order to be able to precisely measure percentile rank, but no search system is given this information. For each search session, the search system asks 12 questions and collects 12 relevance feedback responses. Each question and answer count as one search iteration. If at some stage of the search, the target image is ranked within the top 20 results of the dataset (i.e. with a percentile rank of 99.96% for UT-Zap50K), the search session will terminate early to move to the next session (if any remaining). If so, we consider the missing iterations as having a percentile rank of 100% when computing the average. We asked 10 people (mostly undergraduate and graduate students) to complete live experiment sessions for our user study. From these, we collected search interactions in 300 search sessions (100 sessions per method), with a total of 3,596 relevance feedback responses.

4.4. Implementation

The preprocessing module for learning of the RankCGAN networks and the encoders is implemented in Python with the PyTorch [24] deep learning framework. As discussed in Section 3.1, the CGAN (G, D) and RankNet (R) architecture are built upon RankCGAN [30] and are largely modified from their open-source repository[2]. In particular, we modified their RankCGAN implementation to support more than two attributes. The encoders (E_y and E_z) are implemented according to the IcGAN [25] architecture based on the original Torch implementation[3]. For training, we used the recommended configuration with a minibatch size of 64 and trained the Adam optimizer [13] with $\beta_1 = 0.5, \beta_2 = 0.999$, and a learning rate $\eta = 0.0002$. We trained the RankCGAN networks for 200 epochs, picked the checkpoint that produced the best synthesis and ranking results, then used the learned generator to synthesize 100,000 (x, y, z) tuples and trained the encoder networks for 500 epochs. The search module is implemented in Python.

4.5. Hypothesis 1: Benefits of Image Editing

One major difference between our proposed search approach and recent interactive image search approaches [33, 17, 22, 8] is the use of exclusive visual feedback questions. This change itself does not warrant the use of image editing and synthesis. In particular, one can design a similar search method that uses retrieved images instead of generated ones as feedback options. The modified search method can still use the range-based search and relevance prediction for ranking, and the only difference would be that rather than generating $x_1 = G(z_{\text{ref}}, y_1')$ and $x_2 = G(z_{\text{ref}}, y_2')$, we simply find x_1 and x_2 from the database such that $E_y(x_1)$ is the nearest to y_1' and $E_y(x_2)$ is the nearest to y_2'. The main disadvantage of this approach is that the images of each pair might differ in not only their attribute expression, but also other details that might confuse the user and the system. Additionally, the distribution of the images might be sparse in certain regions of the attribute space, such that the images with the nearest attribute vectors to y_1' and y_2' might be the same, rendering the question ineffective. In our experiment, we want to validate the benefits of image editing by comparing the Syntharch method to the method using retrieved images as options, referred to as **retrieved + range** in Figure 5.

4.6. Hypothesis 2: Benefits of Range-Based Search

In Section 3.3, we described the method of performing a ranged-based binary search with relaxed constraints. In our range-based search, we formulate the problem as range searching in a multidimensional space of visual attributes. In contrast, [15] considers each attribute independently: a binary search tree is formed for each individual attribute, and every tree contains all images in the database. The main reason we proceed differently is that without textual labels of the semantic attribute to pay attention to for each search

[2] https://github.com/saquil/RankCGAN

[3] https://github.com/Guim3/IcGAN

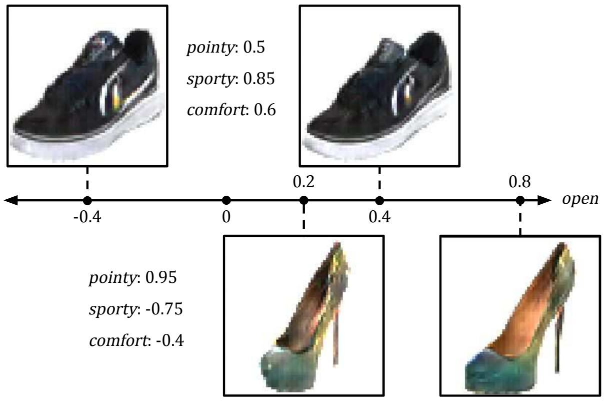

Figure 4. Manipulation of the "open" attribute exhibited differently on the synthesized images in different regions (semantically, "sports shoes" and "high heels") of the attribute space.

iteration, the user receives no guidance on the exact detail to compare against their target image. Therefore, we need the expression of the attribute that is being queried/modified to be as close to that in the target image as possible, because some attribute manipulation might lead to different manifestation in different regions of the attribute space. For example, as shown in Figure 4, in the UT-Zap50K dataset, the expression of *openness* in sports shoes is different from that in high heels.

To verify the benefit of our search where the ranges are updated from feedback on all attributes, we implemented the Attribute Pivots method [15]. Specifically, for each pivot image x_p, we take its estimated vectors $y_p = E_y(x_p)$ and $z_p = E_z(x_p)$. z_p is used as-is for generating both options. For y_p, we add or subtract 0.15 (chosen as optimizing the performance of this method qualitatively) in the attribute dimension we are modifying; this gives us y'_1 and y'_2. This method is labeled **synthesis + pivot** in Figure 5.

4.7. Main Results

Figure 5 visualizes the mean percentile rank over search iterations (higher is better). We observe that by the end of 12 search iterations, the average percentile rank for the Syntharch is the highest at 72.36%, compared to 70.89% for the baseline method of using retrieved images as options (Section 4.5) and 57.13% for using synthesized pivot images for the search (Section 4.6). We can conclude that the Syntharch method is more likely to perform better than both alternative methods by the end of the 12-question search. Further, across all search iterations, the average percentile ranks across all search iterations for the Syntharch, retrieved, and pivot methods are 69.71%, 67.85%, and 54.11% respectively. Therefore, we also establish that the Syntharch method is more likely to perform better than the alternative methods for most iterations. This indicates that both of our hypotheses hold.

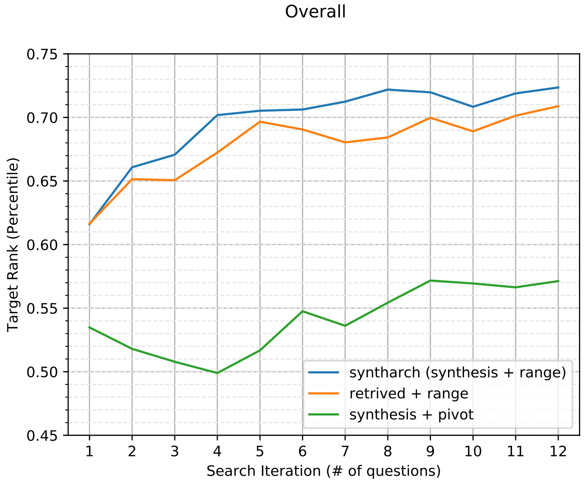

Figure 5. Average percentile rank over iterations by method.

4.8. In-Depth Analysis

As shown in Figure 5, while the rank of the target over time has the general upward trend for all three methods, that of the Syntharch method exhibits a more consistent increasing pattern. We argue that in each search session, the percentile rank *decreases* if and only if the user makes a comparison contrary to the model, which we refer to as "confusing feedback", in the sense that the search system will be confused. On average, we expect users to make more informative feedback than confusing feedback. However, when a large quantity of confusing feedback is made across search sessions, we will observe a decline of average percentile rank (*e.g.* from iteration 8 to 10 in Syntharch, from iteration 4 to 6 in **retrieved + range** and from iteration 9 to 11 in **synthesis + pivot**). To understand the conditions of receiving confusing feedback, we look at specific search sessions with aggressive percentile rank declines.

During the search session shown in Figure 6, there is a sharp declining trend of percentage rank for **retrieved + range** starting after iteration 2. Moreover, the Syntharch method also experiences rank regression starting after iteration 3. When examining the feedback options and the user's choices from these three search sessions reconstructed from the saved log, we observed that during the session with **retrieved + range**, the feedback options remain the same since iteration 4. This is due to the fact that if the previous choices lead a region of the attribute space that is sparse with images, regardless of variations in the attribute vector y, the nearest image will always be the same. Specifically, in iteration 4, $y_1 = (-1/3, -1/3, 1/3, \mathbf{0.5}), y_2 = (-1/3, -1/3, 1/3, -\mathbf{0.5})$, where the last attribute differs by $|-0.5 - 0.5| = 1.0$, yet we still retrieved the same image due to the aforementioned reason. When the two images in the pair are too close to distinguish, users are in-

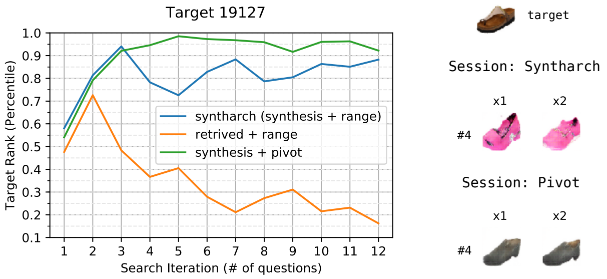

Figure 6. Percentile rank over iterations by method with iteration 4 of the "syntharch" and the "pivot" for one specific search target, showing issues with the "retrieved" method.

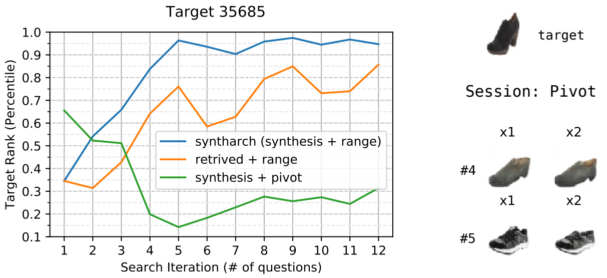

Figure 7. Percentile rank over iterations by method with iteration 4 and 5 of the "pivot" method for one specific search target, showing issues with the "pivot" method.

structed to make an arbitrary selection, meaning that they will provide confusing feedback half of the time. On the other hand, while both Syntharch and **synthesis + pivot** exhibit more diversity thanks to the use of the image generator, when crossing over the sparse region, the synthesized images are at times less realistic. For example, during iteration 4, the quality of the options are too bad for the user to provide any informative feedback. In fact, the user indeed responded with "confusing feedback" during that iteration. Nonetheless, methods using image synthesis are still better at eliciting relevance feedback than plain retrieval, and the percentile rank eventually recovered in this search session as shown in Figure 6. We think the reason that **synthesis + pivot** is less severely affect by the sparse region in this example is that its attribute values in each pair differ by a smaller fixed value of $2 \times 0.15 = 0.3$, as opposed to 1.0 in the case of range searching during the first few iterations.

However, **synthesis + pivot** has its own shortcomings which make it less effective in more general cases for image search with visual feedback questions. In particular, during the search session illustrated in Figure 7, this method suffers harshly at iteration 4 and 5. The feedback questions and the user's choices for the two iterations are shown to the right of the percentile rank plot. At iteration 4, the attribute being considered is "comfort" with $y_1^{(4)} = -0.1345$ and $y_2^{(4)} = 0.1655$ whereas for the target image, $E_y(x_{35685})^{(4)} = -0.6354$. Similar to how the openness of shoes are expressed differently among sports shoes and high heels (as shown previously in Figure 4), the expressions of "comfort" of the shoes for the neighborhood around the target image and that around the pivot image chosen for the iteration are quite different. In this case, increasing the attribute value actually changes the overall shape of x_2, making it more visually similar to the target despite $y_2^{(4)}$ being far greater than $E_y(x_{35685})^{(4)}$, misleading the user, judging purely by visual similarity, to select the option that is farther in the attribute space from the target. A similar pattern occurred at iteration 5, possibly accounting for the confusing feedback from the user and inferior search performance.

To summarize, the study shows that the Syntharch method outperforms both alternative methods on average. In particular, using synthesized as opposed to retrieved images allow Syntharch to have fine control over attributes in regions of the attribute space that do not have sufficient image samples. At the same time, range searching leads to attribute expressions that are more likely to be consistent with those near the target image. Consequently, the combined approach in Syntharch leveraging both image synthesis and range-based search has the best performance of all three methods tested in the user study.

5. Conclusions

We explored a novel approach for interactive image search using only visual feedback. Our Syntharch method incorporates image synthesis and range searching to achieve better accuracy, as a proof of concept for the new approach. The user study results confirmed that (1) using image editing is beneficial over retrieving real images for feedback options and (2) performing a range-based search in a multidimensional attribute space over searching in separate binary search trees lead to better search accuracy. In future work, we aim to improve image quality results. We will also explore an alternative search strategy that uses ranges based on density along an attribute dimension, and experiment with non-nameable, latent attributes for feedback.

References

[1] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM.

[2] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The bayesian image retrieval system, pichunter: theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, Jan 2000.

[3] J. Cui, F. Wen, and X. Tang. Real time google and live image search re-ranking. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 729–732, New York, NY, USA, 2008. ACM.

[4] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, June 2009.

[5] M. Ferecatu and D. Geman. A statistical framework for image category search from a mental picture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1087–1101, June 2009.

[6] J. Fogarty, D. Tan, A. Kapoor, and S. Winder. Cueflik: Interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 29–38, New York, NY, USA, 2008. ACM.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[8] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Feris. Dialog-based interactive image retrieval. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 678–688. Curran Associates, Inc., 2018.

[9] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, July 2017.

[10] C. C. Jaffe, H. D. Tagare, and J. Duncan. Medical Image Databases: A Content-based Retrieval Approach. *Journal of the American Medical Informatics Association*, 4(3):184–198, 05 1997.

[11] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[12] T. Kaneko, K. Hiramatsu, and K. Kashino. Generative attribute controller with conditional filtered generative adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7006–7015, July 2017.

[13] D. Kingma and L. Ba. Adam: A method for stochastic optimization. In F. Amsterdam Machine Learning lab (IVI, editor, *International Conference on Learning Representations (ICLR)*. arXiv.org, 2015.

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] A. Kovashka and K. Grauman. Attribute pivots for guiding relevance feedback in image search. In *2013 IEEE International Conference on Computer Vision*, pages 297–304, Dec 2013.

[16] A. Kovashka and K. Grauman. Discovering attribute shades of meaning with the crowd. *International Journal of Computer Vision*, 114(1):56–73, 2015.

[17] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Interactive image search with relative attribute feedback. *International Journal of Computer Vision*, 115(2):185–210, Nov 2015.

[18] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2539–2547. Curran Associates, Inc., 2015.

[19] M. La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Proceedings. IEEE Workshop on Content-Based Access of Image and Video Libraries (Cat. No.98EX173)*, pages 24–28, June 1998.

[20] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. DENOYER, and M. A. Ranzato. Fader networks:manipulating images by sliding attributes. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5967–5976. Curran Associates, Inc., 2017.

[21] E. Mansimov, E. Parisotto, J. Ba, and R. Salakhutdinov. Generating images from captions with attention. In *ICLR*, 2016.

[22] N. Murrugarra-Llerena and A. Kovashka. Image retrieval with mixed initiative and multimodal feedback. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 310, 2018.

[23] D. Parikh and K. Grauman. Relative attributes. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 503–510, Washington, DC, USA, 2011. IEEE Computer Society.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[25] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez. Invertible Conditional GANs for image editing. In *NIPS Workshop on Adversarial Training*, 2016.

[26] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[27] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[28] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1060–1069, New York, New York, USA, 20–22 Jun 2016. PMLR.

[29] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, Sep. 1998.

[30] Y. Saquil, K. I. Kim, and P. Hall. Ranking cgans: Subjective control over semantic image attributes. In *Proc. of British Machine Vision Conference (BMVC)*, 7 2018.

[31] P. C. Saraiva, J. M. Cavalcanti, E. S. de Moura, M. A. Gonalves, and R. da S. Torres. A multimodal query expansion based on genetic programming for visually-oriented e-commerce applications. *Information Processing & Management*, 52(5):783 – 800, 2016.

[32] S. Sclaroff, L. Taycher, and M. La Cascia. Imagerover: a content-based image browser for the world wide web. In *1997 Proceedings IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 2–9, June 1997.

[33] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR 2011*, pages 801–808, June 2011.

[34] M. J. Swain. Interactive indexing into image databases. In *Storage and Retrieval for Image and Video Databases*, volume 1908, pages 95–104. International Society for Optics and Photonics, 1993.

[35] B. Thomee and M. S. Lew. Interactive search in image retrieval: a survey. *International Journal of Multimedia Information Retrieval*, 1(2):71–86, Jul 2012.

[36] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2image: Conditional image generation from visual attributes. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 776–791, Cham, 2016. Springer International Publishing.

[37] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1099–1107. Curran Associates, Inc., 2015.

[38] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *Computer Vision and Pattern Recognition (CVPR)*, Jun 2014.

[39] A. Yu and K. Grauman. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *International Conference on Computer Vision (ICCV)*, Oct 2017.

[40] A. Yu and K. Grauman. Thinking outside the pool: Active training image creation for relative attributes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, Apr 2003.

[42] X. S. Zhou, S. Zillner, M. Moeller, M. Sintek, Y. Zhan, A. Krishnan, and A. Gupta. Semantics and cbir: A medical imaging perspective. In *Proceedings of the 2008 International Conference on Content-based Image and Video Retrieval*, CIVR '08, pages 571–580, New York, NY, USA, 2008. ACM.

[43] Z. Zhou, Y. Xu, J. Zhou, and L. Zhang. Interactive image search for clothing recommendation. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 754–756, New York, NY, USA, 2016. ACM.