SYNTHARCH: INTERACTIVE IMAGE SEARCH WITH ATTRIBUTE-CONDITIONED SYNTHESIS

by

Zac Yu

University of Pittsburgh, 2019

Submitted to the Graduate Faculty of

the Department of Computer Science in partial fulfillment

of the requirements for the degree of

Bachelor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

SCHOOL OF COMPUTING AND INFORMATION

This thesis was presented

by

Zac Yu

It was defended on

May 28, 2019

and approved by

Adriana Kovashka, Assistant Professor, Department of Computer Science

Daqing He, Professor, Department of Informatics and Networked Systems

Erin Walker, Associate Professor, Department of Computer Science

Mayank Goel, Assistant Professor, Human-Computer Interaction Institute,

School of Computer Science, Carnegie Mellon University

Thesis Advisor: Adriana Kovashka, Assistant Professor, Department of Computer Science

SYNTHARCH: INTERACTIVE IMAGE SEARCH WITH ATTRIBUTE-CONDITIONED SYNTHESIS

Zac Yu, BPhil

University of Pittsburgh, 2019

The use of interactive systems has been proposed and 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 new semantic gap between the two parties' understanding. In my thesis, I 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.

Thanks to the recent advancements in generative adversarial networks, we can now generate realistic images of certain controlled characteristics and use a multidimensional attribute space learned from an image database to condition image synthesis. I present Syntharch, a novel interactive image search approach which uses synthesized images as options instead of textual questions to gain information on the relative attribute values of the target image. For each iteration of the search, rather than asking the user to make an attribute-value comparison in words, Syntharch generates a pair of options (synthesized images) which varies only in one attribute and let the user select the option that is more visually similar to the target.

I then demonstrate that using synthesized images rather than real images retrieved from

the database as feedback options, Syntharch causes less confusion to the user. Further, I establish that the specific search method I propose performs similarly or better in comparison to the conventional approach.

Overall, my thesis presents a new approach of interactive image search, proposes a specific implementation following that approach, and validates the hypotheses that guided the search approach as well as the implementation choices.

Keywords: Content-Based Image Retrieval (CBIR), Interactive Image Search, Generative Adversarial Network (GAN), Image Synthesis, Image Editing, Computer Vision, Human-Computer Interaction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I wish to thank various people for their help and support throughout my undergraduate career. By the chronological order, I would first like to thank Dr. Arthur Kosowsky for his enlightening teaching of Honors Physics during my freshman year and his effort to knowing his students personally; his enthusiasm in sharing about his research work inspired me with the beauty of research and motivated me to start working on research as an undergraduate student. I want to thank Dr. Jingtao Wang for accepting me into his lab during my first year and offering me my first job after I started college, to collaborate on CourseMIRROR, an interesting multidisciplinary project. I am grateful of Dr. Bruce McLaren for offering the opportunity to collaborate with his student Ken Holstein on an independent study at the Human-Computer Interaction Institute (HCII) at Carnegie Mellon University; both of them provided me with tremendous help in guiding my academic career. I also wish to acknowledge Dr. Mary McGlohon, my internship host at Google, for sharing her experience in graduate school, encouraging the pursuit of my research interests, and helping me to navigate between the industry and the academia.

I would like to express my very great appreciation to everyone who contributed to this project. In particular, I would like to thank Dr. Mayank Goel for allowing me to join his lab and working with his students on various projects at HCII. During the collaboration of the GymCam project with his student Rushil Khurana, I discovered interests in computer vision and applied machine learning, which prompted me to explore more about those branches of study and eventually led to the start of my thesis project. I want to thank both of them for providing helpful early feedback when I was starting this project and Dr. Goel for joining my defense committee as the external examiner. The advice given by other thesis committee members including Dr. Daqing He and Dr. Erin Walker has also been a great

help in discovering new aspects of the problem and improving my thesis drafts.

I would like to offer my special thanks to Dr. Adriana Kovashka. This thesis project would not have been made possible without her useful pointers, valuable critiques, patient guidance, and firm support. Her inspirational teaching in the introductory courses to computer vision and machine learning helped me learn the fundamentals systematically and encouraged me to start my exploration for this thesis work. I am especially thankful for her willingness to dedicate her time from a packed schedule to work with me on this project and to guide me toward the completion of my degree.

Finally, I would like to acknowledge the support provided by my family and close friends. It is their enthusiastic encouragement and selfless assistance that made me stay confident in following my interests and pursuing my ambitious dreams.

1.0 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 [18, 3], e-commerce [31, 44], health care [10, 43], 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 of the task is to establish a mapping between the user's high-level concept and the machine's low-level representation of the image.

On the other hand, 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 gain additional information regarding the target image iteratively [35]. For each iteration of the interactive search, the system needs to accept some form of feedback, and its efficiency is still hindered by the challenge of the CBIR task discussed in the previous paragraph.

Relative attributes can help solve this challenge by transforming the machine's low-level image representation to high-level semantic attributes which can be expressed in textual form and understood easily by the user [23, 15, 16]. 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. The two immediate drawbacks are that (1) in order for the system to distinguish between semantic attributes, we will need to manually pick the attributes and let the system learn to classify them under some form of supervision ahead of time, and that

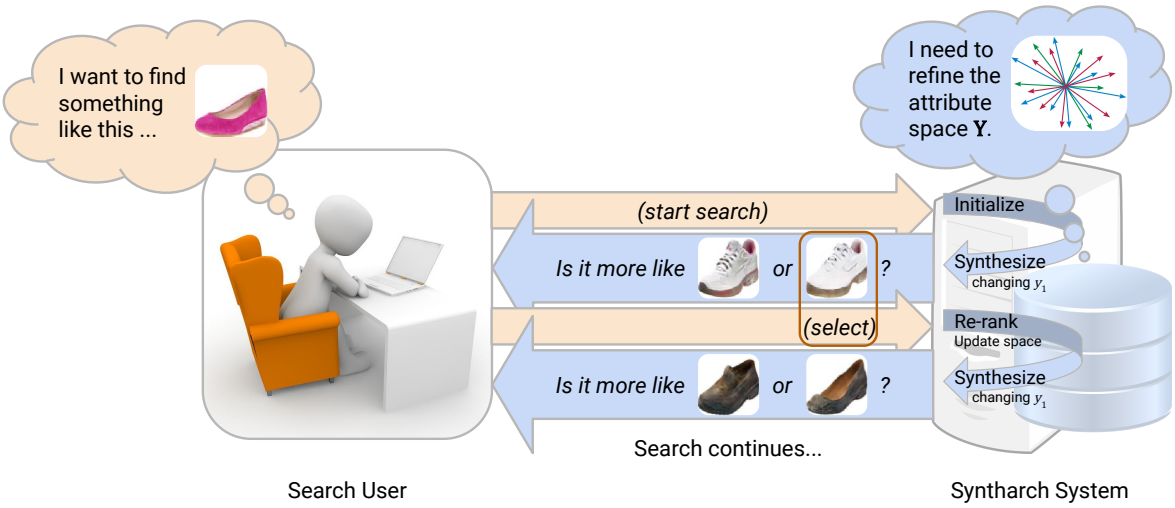

Figure 1: Syntharch elicits attribute feedback by synthesizing pairs of options and then uses user responses to re-rank images in the database and to refine the search space.

(2) 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.

1.1 PROPOSED SYSTEM

To address these challenges of interactive image search, I propose Syntharch, a new way to close the semantic gap using visual-only feedback on high-level attributes. In Syntharch, given a database of images with relative attributes, we can learn a ranker to produce decorrelated and normalized attribute vectors in a multidimensional space $\mathbf{Y}$. Then, using a generator learned during the same session, each attribute vector, along with a latent noise vector, can be transformed into an image which preserves all the visual features of the original image. Once the attribute space and the generator are learned, we will be able to 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. As

shown in Figure 1, synthesized images that differ in attribute values are presented to the user as options. When the user makes a selection that one option is closer to their target than another, the system can then incorporate that feedback, re-rank the images, and possibly continue with more questions. Syntharch allows universal image search independent of the user's understanding of the attributes. It further 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.

1.2 THESIS STATEMENT

This thesis explores the usage of visual feedback options for interactive image search and evaluates the hypothesis that image synthesis and range searching can be used to improve the search accuracy.

1.3 OUTLINE

This chapter discusses the status quo of interactive search approach for the CBIR task, highlights the major challenges, and offers an overview of the thesis.

Chapter 2 provides a more comprehensive review of various tasks and methods relevant to this work and previous literature that contribute to the proposed approach.

Chapter 3 presents Syntharch, a novel interactive image search approach, expounds on the design of its system, and illustrates its changes over previous systems.

Chapter 4 details the experiments to evaluate the changes introduced in Syntharch and to validate if they indeed facilitate the search process.

Chapter 5 reviews the limitations of the system, offers potential future improvements, and summarizes the thesis work.

2.0 RELATED WORK

2.1 INTERACTIVE IMAGE SEARCH

The image search system I 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, 4, 35]. Early approaches, most notably those adopted by web image search engines, utilize low-level features such as color, dimension, and shape as cues for the image content [34, 32, 18]. In recent years, relevance feedback has been shown to be more effective and accommodating to high-level concepts [29, 4, 5, 2, 16, 42]. By incorporating relevance feedback, search systems can iteratively gain information on the target image and approach to it.

2.2 ATTRIBUTE-BASED SEARCH

Relative attributes are used to initialize the search with a random distribution and to facilitate the interactive search by allowing comparative feedback [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 that reference [16]. In Attribute Pivots [15], Kovashka and Grauman proposed searching with a binary search tree for each relative attribute. Recently, there have also been explorations with a larger variety of feedback forms. Murrugarra-Llerena and Kovashka explored free-form

attribute feedback which allows the user to pick both the attribute and the reference image when making a comparison [21]. They also experimented with a visual form of feedback, specifically, requesting the user to draw a sketch of the target. However, this visual feedback form does not use the image attribute data, and can only be used as a supplemental method. Guo, et al. proposed converting the search interactions into a dialog between the user and the search system, accepting feedback in natural language [7]. 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. My approach with Syntharch expresses relative attributes solely in visual forms.

2.3 CONDITIONAL IMAGE SYNTHESIS

Recently, synthesis of realistic images has been greatly empowered by deep generative models including variational autoencoders (VAE) [17, 36], introduced by Kingma and Welling in 2013 [13], and generative adversarial networks (GAN) [20, 9, 30], proposed by Goodfellow, et al. in 2014 [6]. In 2015, Radford et al. introduced the class of deep convolutional generative adversarial networks (DCGAN) [27], which are capable of synthesizing highly compelling and detailed images. Meanwhile, conditional generative networks (CGAN) enable modulation of the output image based on parameters including text [20], images [9, 11], and attribute values [36, 19, 12, 30, 40]. In 2016, Yan et al. presented Attribute2Image [36] which formulates the conditional image generation problem and suggests the approach of using a variational auto-encoder to estimate the posterior distributions of the disentangled foreground and background image to generate the composite full image. Then in 2017, Lample et al. presented the Fader network [19], incorporating an attribute encoder at training time to allow generating variances of an image with controlled attribute values. Other approaches such as CFGAN [12] and RankCGAN [30] rely on training adversarial networks and using a conditional vector as an additional input of the generator to control the attributes. In Syntharch, the conditional image synthesis module based on RankCGAN is integrated into a preprocessing module to construct a multidimensional attribute space for image synthesis.

2.4 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 [6] by Goodfellow et al. lacks the capability to project real images onto the latent space for reconstruction despite having advantages in generating clearer and more realistic images. To enable GAN for image editing, 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]. In 2016, Perarnau et al. presented Invertible Conditional GAN (IcGAN) [25], 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 I built for recovering the latent noise vector and manipulating image attribute values is based on the network proposed in IcGAN.

3.0 APPROACH

I 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 comprises two modules: a preprocessing module that performs a 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. In Section 3.1, I explain my method to train the generator as the first stage of the preprocessing module. In Section 3.2, I present the second stage of the module, training an encoder which maps every given image to an estimated representation in the latent space. Section 3.3 introduces the search module and explains how it produces the questions. And finally, I talk about how the search module ranks the images and outputs the search results in Section 3.4.

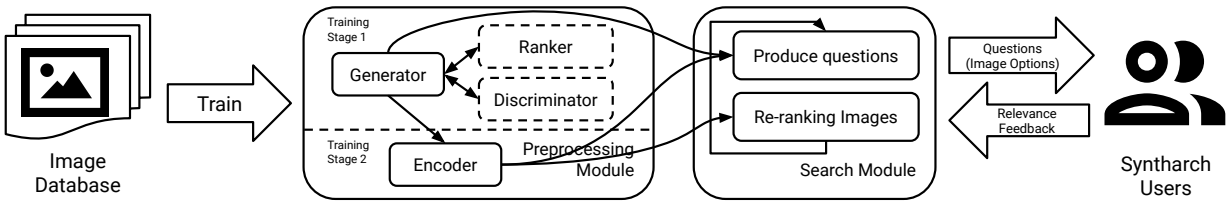

Figure 2: Syntharch preprocesses each image database with a two-stage training, which enables the interactive search.

3.1 GENERATOR NETWORKS

The generator networks used in this thesis is adopted from RankCGAN [30], which is a combination of a conditional GAN (CGAN) and a RankNet [1]. In this design, the generator, the ranker, and the discriminator are trained simultaneously, and they become crucial components of the Syntharch system.

3.1.1 Conditional GAN

The CGAN of the preprocessing module is composed of two neural networks, a generator $G(z, y)$ which takes in a latent noise vector z with an attribute vector y and outputs a synthesized image, and a discriminator $D(x)$ which takes in an image x and outputs the likelihood of the image x being real (i.e. not artificially synthesized).

At training time, the goal of the generator is to approximate the distribution of the image dataset $p_{dataset}$ in order to pass the discriminator whereas the goal of the discriminator is to distinguish between images from the dataset distribution (i.e. real images) and those that are not (i.e. fake, or synthesized images). The objective of the training of a conventional CGAN is to minimize the binary cross-entropy (BCE) loss, and can be formulated as

$$\min_G \max_D \mathbb{E}_{x,y\backsim p_{dataset}}[\log D(x, y)] + \mathbb{E}_{z\backsim p_z, y'\backsim p_y}[\log(1 - D(G(z, y')), y')]. \tag{3.1}$$

Note that in the RankCGAN network, the attribute vector y is not an input of the discriminator. Instead, the ordering of y is regulated by a separate ranker network, which will be expounded in Section 3.1.2.

As shown in Figure 3, for the generator network G, the latent vector $z \backsim \mathcal{N}(0, I)$ has a length of 100 and the length of the attribute vector $y \backsim \mathcal{U}(-1, 1)$ equals to the number of attributes in the image database. In practice, the concatenation of the two vectors serves as the input of the network. G has four hidden layers in total, each formed by a full convolution (transposed convolution) followed by batch normalization and the ReLU. The output layer, a three-channel image of dimension 64 by 64, is formed by tanh after a full convolution.

Figure 4 shows the discriminator network which instead takes in an image and makes a prediction indicating if the input image is real. Similar to the generator, D also has four

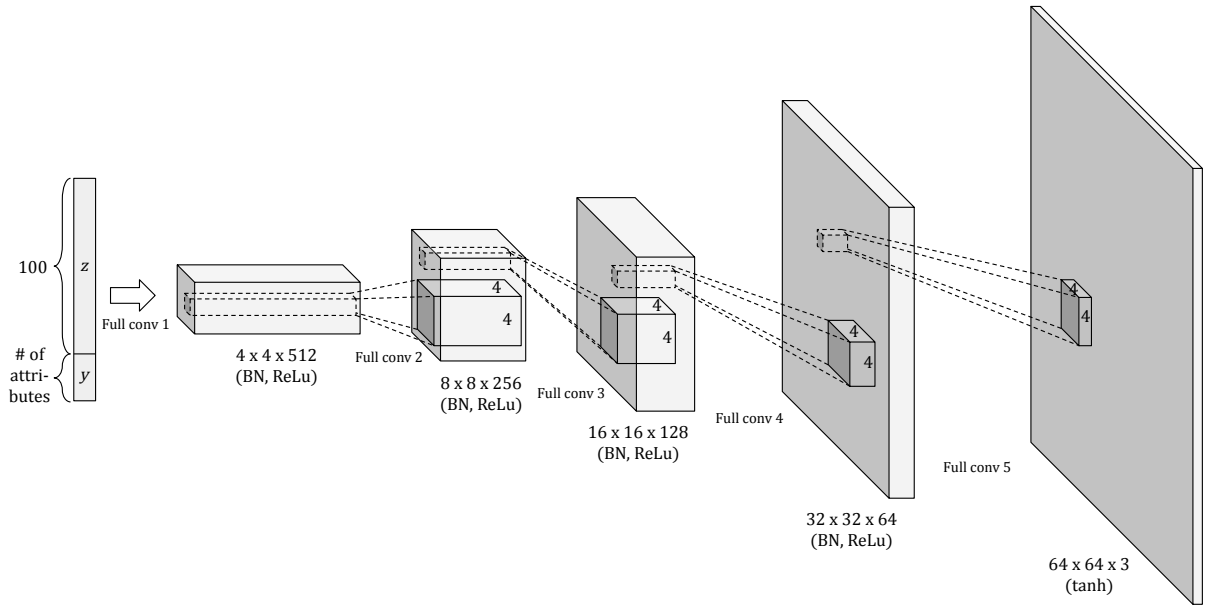

Figure 3: The architecture of the generator network.

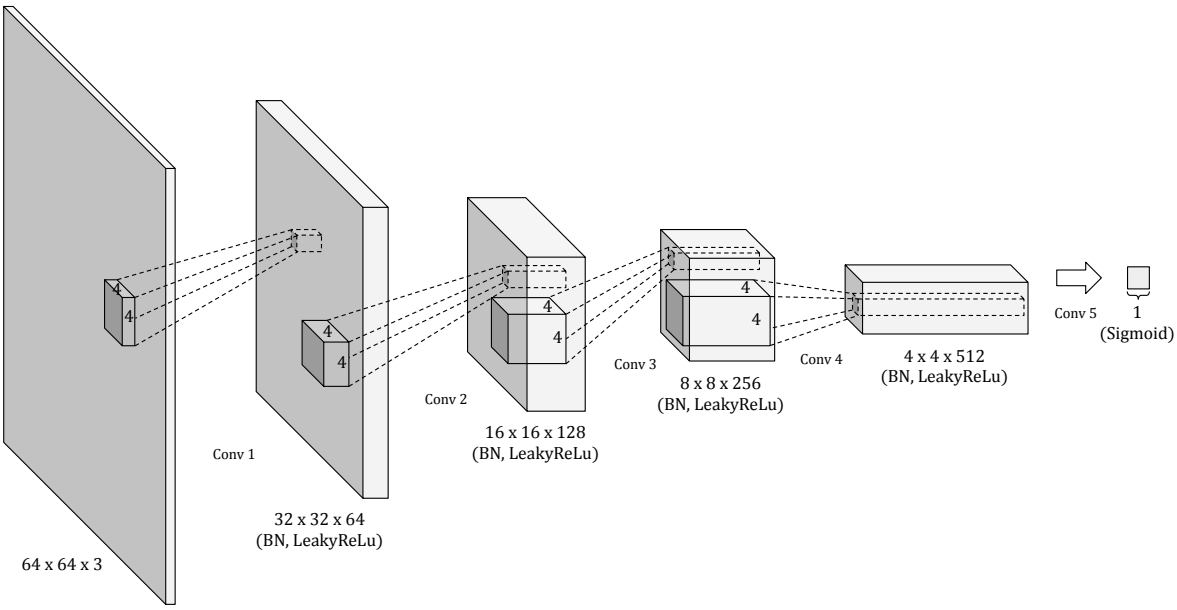

Figure 4: The architecture of the discriminator network.

hidden layers. However, they are results of regular convolutions applied with leaky ReLU. The output layer of D is a single scalar in $(0, 1)$ produced by a convolution followed by a sigmoid function. It captures the likelihood of input image x being real. As noted above, y is not an input of the discriminator in the RankCGAN architecture. Consequently, this network differs from the discriminator in a conventional CGAN in that the first hidden convolutional layer is not concatenated with the attribute vector y.

3.1.2 RankNet

RankNet [1] uses gradient descent methods for learning ranking functions, which can be then used to estimate pairwise comparisons of image attributes and to help in decorrelating the attributes. In order to achieve that, we need binary labels representing pairwise attribute comparisons of images in the database. The requirement of these labels introduces a limitation which will be discussed in details in Section 5.1. On the other hand, we benefit from using pairwise comparisons as opposed to exact values when regulating the ranker because it allows us to formulate the combined optimization problem more easily. This is because with binary (greater than v.s. less than[1]) pairwise comparison results, we can formulate the RankNet loss as a BCE loss, similar to that of the discriminator. Specifically,

$$\mathcal{L}_R(x_i^{(1)}, x_i^{(2)}, c_i) = -c_i \log(p_i) - (1 - y_i) \log(1 - p_i), \tag{3.2}$$

where p_i is the posterior probability based on the estimated ranking score

$$p_i = \text{sigmoid}\left(R\left(x_i^{(1)}\right) - R\left(x_i^{(2)}\right)\right), \tag{3.3}$$

and c_i is the binary comparison result either given as a sample label or inferred from the y values used for synthesis.

The particular RankNet we use for the ranker shares the same structure as the discriminator as illustrated in Figure 4, except for the output layer. For the ranker, the sigmoid function is not applied to the output (ranking layer) because we only care about the pairwise

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

ranking orders. A dedicated ranking layer is appended to the last hidden layer in parallel for each attribute (affecting only the output layer; all prior layers are shared).

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 be 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 the distribution of $y \backsim \mathcal{U}(-1, 1)$.

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, we learn a network to minimize the mean squared error (MSE) loss of

$$\mathcal{L}_{E_z}(x) = \mathbb{E}_{z \backsim p_z, y' \backsim p_y} ||z - E_z(G(z, y'))||_2^2 \tag{3.4}$$

and

$$\mathcal{L}_{E_y}(x) = \mathbb{E}_{z' \backsim p_z, y \backsim p_y} ||y - E_y(G(z', y))||_2^2. \tag{3.5}$$

The architecture of the encoder network is shown in Figure 5. It has four hidden convolutional layers, each applied with batch normalization and ReLU. The last convolutional layer is then flattened, followed by two linear transformations which finally outputs an estimated vector of either y or z. The hidden linear layer before the output is applied with one-dimensional batch normalization and ReLU.

The learned encoders, together with its generator counterpart, allow us to reconstruct images. Figure 6 showcases some of the reconstruction results trained on the UT-Zap50K dataset [38, 39]. I observe that while not all the details (e.g. colors and fine patterns) are

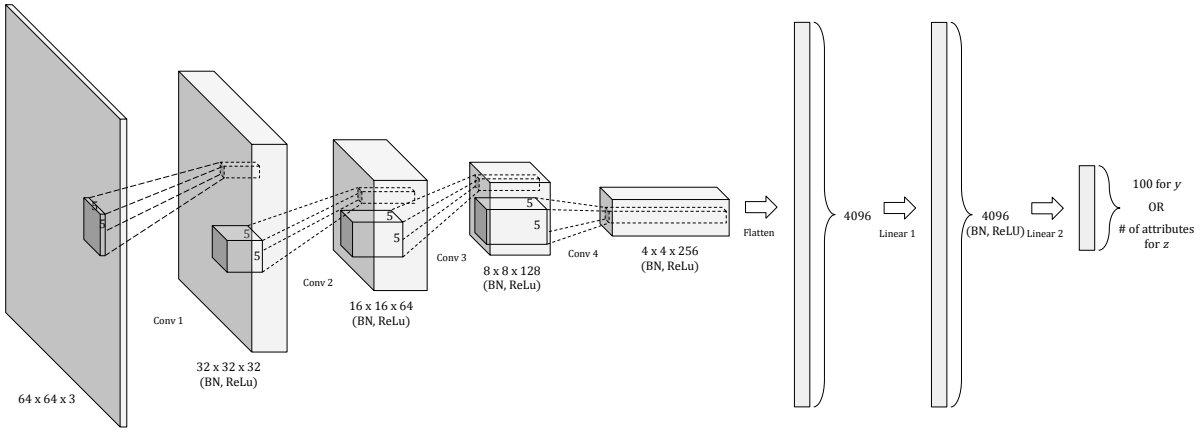

Figure 5: The architecture of the encoder network.

Figure 6: Reconstruction results: in each pair of two, the image on the left is a real image from the database and the one on the right is reconstructed by the generator with inputs estimated by the encoders.

fully preserved in the reconstructed image, the general shape and the style, as dictated by the labeled attributes of the dataset, are similar in every pair.

In practice, to edit the attribute values of an image x, we can first obtain the estimated vectors $y = E_y(x), z = E_z(x)$ from the encoder. Then, while fixing z, we can modify y to y'[2] of some desired attribute and obtain the edited image $x' = G(z, y')$. Figure 7 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 $y = [0.1536, -0.0565, 0.7329, 0.0863]$. Recall that each dimension corresponds to an attribute: for UT-Zap50K, they are *open*, *pointy*, *sporty*, and *comfort*. The attribute vector used for generating the first edited image labeled "open +0.5" is therefore $y' = [0.6536, -0.0565, 0.7329, 0.0863]$.

3.3 RANGE-BASED SEARCH

During the search, we maintain a 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, all attributes except for the attribute n we are querying are set to the center values of their corresponding ranges, i.e. $y_1'^{(m)} = y_2'^{(m)} r_m^{(1)}/2 + r_m^{(2)}/2$. 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, i.e. $y_1'^{(n)} = 3r_n^{(1)}/4 + r_n^{(2)}/4$, while that of the other to be at the 3/4 of the range, i.e. $y_2'^{(n)} = r_n^{(1)}/4 + 3r_n^{(2)}/4$. I 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

[2]We use the prime symbol ($'$) to denote modified values.

Figure 7: Image editing using the encoders and the generator.

ranges are $r_1 = [-0.5, 0.7]$, $r_2 = [0, 0.6]$, 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 $y'_1 = (0.1, 0.15, 0.4), y'_2 = (0.1, 0.45, 0.4)$. Notice that the attribute values for the first and the third dimensions are intentionally kept the same; at the same time, for the second dimension, the values are precisely at 1/4 and 3/4 of the range.

To generate an image, we also need the noise vector z. To ensure that the synthesized results are realistic, we search for the image x_{ref} from our database whose estimated attribute vector $y_{\text{ref}} = E_y(x_{ref})$ is the closest (with the least Euclidean distance) to $(y'_1 + y'_2)/2$. 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. However, the performance would then be heavily affected by any mistakes 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 and 3/4 of the range. For Syntharch, I 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, an image search system, 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.

Since we modeled the iterative search problem as a range search problem, as discussed in the previous section, a naive solution would be to rank each image x_i by its Euclidean distance to the center of the search range. This ranking approach, while plausible, does not account for the specific choices. During early-stage experiments, I found its performance inferior to the probabilistic approach used in the [15] and decided to opt for the latter.

The probabilistic-based relevance prediction model, modified from that in [15], can be formulated as the following. Given the set $\mathcal{F}$, for each image x_i, we want to compute its probability of relevance $P(\text{relevant} \mid x_i, \mathcal{F})$.

Now, consider the choices, let $S_{k,i} \in \{0, 1\}$ represent whether image x_i satisfies the binary search constraint (we use the constraint with tolerance, i.e. reducing $1/3$ of the range each time) in the k-th 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,

$$P(\text{relevant} \mid x_i, \mathcal{F}) = \sum_{k=1}^{T} \log P(S_{k,i} = 1 \mid x_i). \tag{3.6}$$

Then we can use Platt's method [26] to estimate the probabilities with the following transform,

$$\log P(S_{k,i} = 1 \mid x_i) = \begin{cases} 1 - \frac{1}{\exp(\alpha_m)\left(y_i^{(m)} - \left(2r_{m,k}^{(1)}/3 + r_{m,k}^{(2)}/3\right) + \beta_m\right)} & \text{if } f = 1 \\ \frac{1}{\exp(\alpha_m)\left(y_i^{(m)} - \left(r_{m,k}^{(1)}/3 + 2r_{m,k}^{(2)}/3\right) + \beta_m\right)} & \text{if } f = 2 \end{cases}, \tag{3.7}$$

where α_m and β_m are learned from the pairwise comparison labels as well as the output of E_y on all images of the database.

As pointed out in [15], the probabilistic model further allows for some mistakes in relevance feedback in addition to our relaxed binary search constraint.

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.

4.0 EXPERIMENTAL VALIDATION

To evaluate to what extent Syntharch's addition to the conventional approach contributes to the accuracy of the task of interactive image search, I set up a quantitative user study. In Section 4.1, I introduce the overall design of the experiment. Section 4.2 and Section 4.3 detail two hypotheses regarding Syntharch's contributions and how they are validated through the experiment. Finally, I present and analyze the experiment results in Section 4.4 and Section 4.5.

4.1 DESIGN

4.1.1 Dataset

I evaluate Syntharch with the UT-Zap50K dataset [38, 39], a public image dataset consisting of 50,025 catalog images of shoes with 4 relative attributes labels: open, pointy, sporty, and comfort. The attribute labels are provided in the form of 6,751 fine-grained ordered pairs. Each 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.

4.1.2 Metric

Similar to previous work [15, 16, 21, 7], I 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. 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

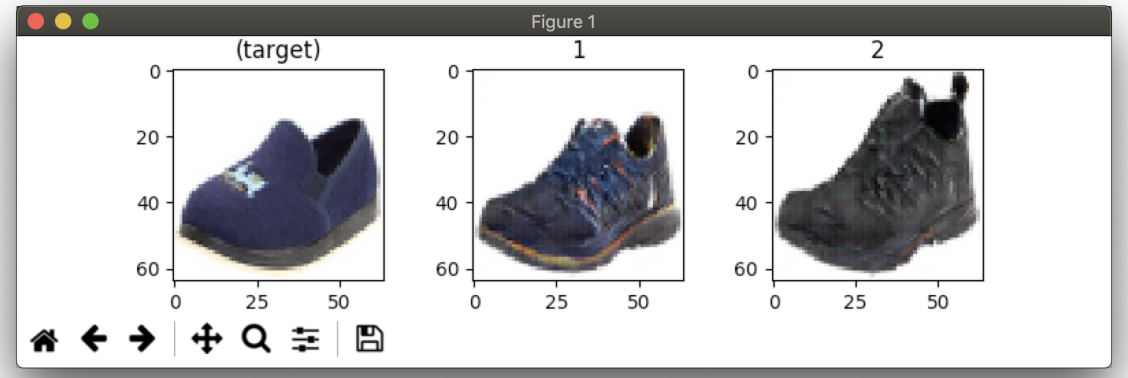

Figure 8: The search experiment UI showing the target image and two options.

results. Therefore, the higher the percentile rank, the closer we are to the target image at that iteration of the search session, 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.

4.1.3 Protocol

For each experiment session, we run 30 search sessions in total. 10 of the search sessions will use the Syntharch method, 10 of them using the alternative (baseline) method described in Section 4.2 and the rest 10 using the method described in Section 4.3. There are 10 random search targets in total, and each appears precisely one time for each method. 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, select the option between the two that is closer to the target. Figure 8 shows the search experiment user interface (UI) with one target image and two option images labeled as 1 and 2 where the user is prompted to make a selection

between them. Note that the two options do not directly correspond to x_1 and x_2 as their order is also randomized.

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 forward to the next session (if any remaining). When that happens, we consider the missing iterations as having a percentile rank of 100% when computing the average later.

4.1.4 Implementation

The preprocessing module used for the user study 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[1]. In particular, I 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 from their open-source repository[2]. For training, I used the recommended configuration with a mini-batch size of 64 and trained the Adam optimizer [14] with $\beta_1 = 0.5, \beta_2 = 0.999$, and a learning rate $\eta = 0.0002$.

I trained the RankCGAN networks for 200 epochs. Because the networks did not converge (as shown in Figure 9), I handpicked the checkpoint from the epoch that seemed to produce the best synthesis and ranking results (epoch #176).

I then used the learned generator to synthesize 100,000 (x, y, z) tuples and trained the encoder networks for 500 epochs. The networks converged (as shown in Figure 10) and I determined that the checkpoint after 500 epochs is sufficient for the image editing task.

The search module is also implemented in Python, with command line interaction and Matplotlib [8] for displaying the questions (see Figure 8).

[1] Repository available at `https://github.com/saquil/RankCGAN`

[2] `https://github.com/Guim3/IcGAN`

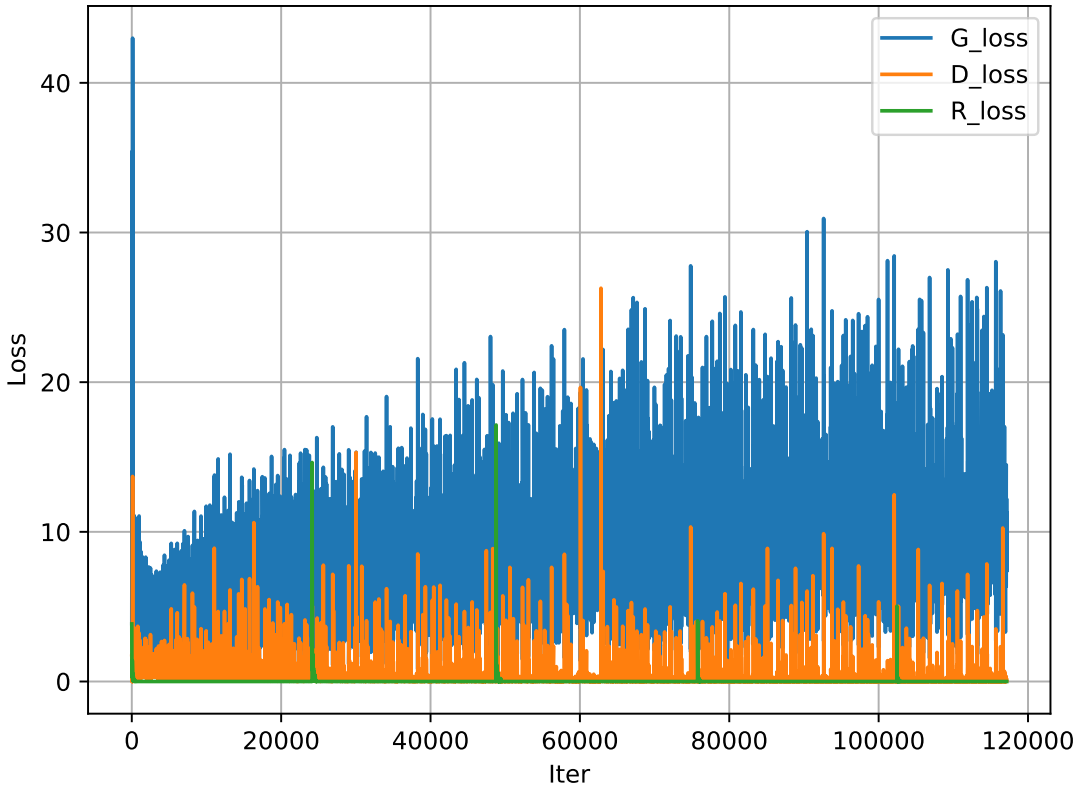

Figure 9: The training loss of RankCGAN networks over time.

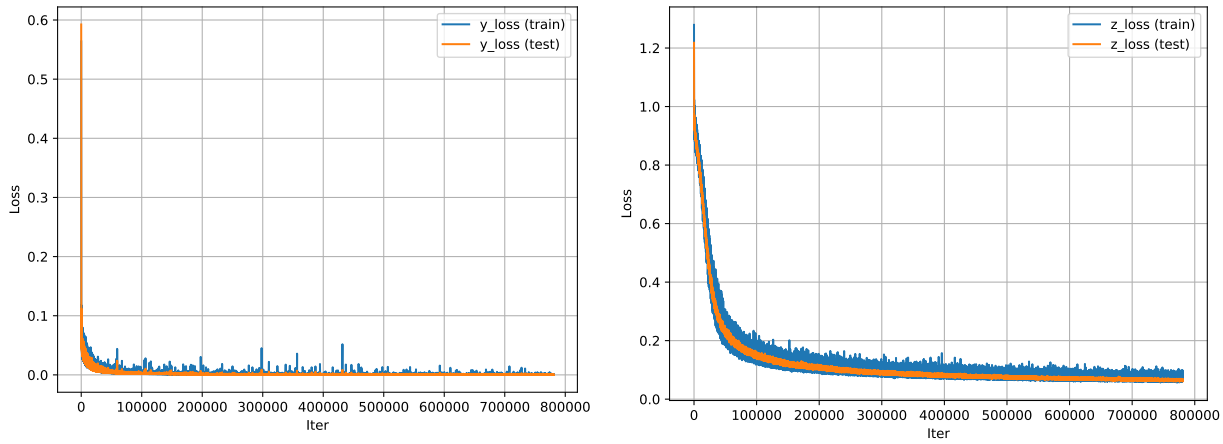

Figure 10: The training loss of the encoder networks over time.

4.2 HYPOTHESIS: BENEFITS OF IMAGE EDITING

One major difference between my proposed search approach and conventional interactive image search approaches 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. Note that although we are still using the y-encoder E_y to estimate the attribute vectors, it does not necessarily have to be learned after the generator as it is in the Syntharch design (Section 3.2). Instead, we can train a ranking function using only images from the database and their comparison labels.

Note that this method is similar to an alternative discussed in Section 3.3 of using z-vectors of images in the database that has the nearest attribute vectors to y'_1 and y'_2 for the synthesis. Moreover, we have the same concern 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. For example, unlike the image editing results in Figure 7, the retrieved images might have different detailed patterns unrelated to the attributes yet causing the user to choose one option over another. Additionally, the distribution of the images might be sparse in certain regions of the attribute space, that the images with the nearest attribute vectors to y'_1 and y'_2 might be the same, rendering the question ineffective.

In my experiment, I want to validate the benefits of image editing by comparing the Syntharch method to the method using retrieved images as options. I implemented the modified search method discussed above, and perform 10 search sessions of this method in each experiment session.

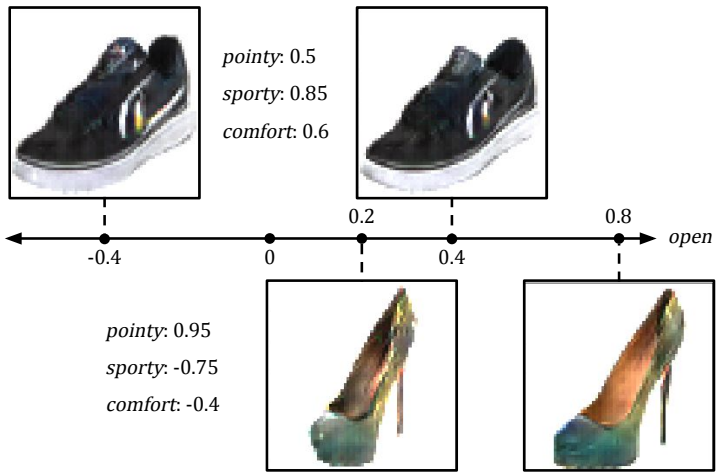

Figure 11: Manipulation of the "open" attribute exhibited differently on the synthesized images in different regions of the attribute space.

4.3 HYPOTHESIS: BENEFITS OF RANGE-BASED SEARCH

In Section 3.3, I illustrated the method of performing a ranged-based binary search with relaxed constraint. The intuition of maintaining a search range despite not needing it for ranking results (Section 3.4) is that the synthesized feedback options can, therefore, be more realistic and that subtle differences can then better reflect the attribute value variations.

I now contrast this method with the approach in [15] which considers each attribute independently. In their work, a binary search tree is formed for each individual attribute, and every tree contains all images in the database. In my range-based search, I formulate the problem as range searching in a multidimensional space of visual attributes. The main reason to proceed with the search differently is that without textual labels of the semantic attribute to pay attention to for each search iteration, the user is unaware of the detail to compare against their target image. Therefore, we need the expression of the attribute 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 11, in the UT-Zap50K dataset, the expression of *openness* in sports shoes is

different from that in high heels.

To verify this assumption and the benefits of range searching, I implemented the pivot method based on the Attribute Pivots work. 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, and y_p is added or subtracted the fixed value of 0.15 in the attribute dimension we are modifying and becomes y'_1 and y'_2.

Similar to the experiment setup in Section 4.2, we perform 10 search sessions of this method in each experiment session.

4.4 QUANTITATIVE RESULTS

I asked 9 people (mostly undergraduate and graduate students) to complete live experiment sessions for my user study. From them, I collected search interactions of 270 search sessions (90 sessions for each method), with a total of 3,236 relevance feedback responses.

Table 1 lists the means and the standard deviations of the percentile rank over search iterations. The mean percentile rank from the table is also plotted in 12 for visualization. We observe that by the end of 12 search iterations, the average percentile rank for the Syntharch is the highest at 71.90%, compared to 69.77% for the baseline method of using retrieved images as options (Section 4.2) and 56.59% for using synthesized pivot images for the search (Section 4.3). 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. On the other hand, across all search iterations, the average percentile ranks across all search iterations for the Syntharch, retrieved, and pivot methods are 68.80%, 67.10%, and 54.10% respectively. Therefore, we also establish that the Syntharch method is more likely to perform better than the alternative methods for most iterations.

With the quantitative results, we can perform a statistical test on the percentile ranks after the last iteration (iteration 11) with the null hypothesis H_0 that the all three methods result in the same overall rank percentile. I decided to use the Friedman test instead of the one-way analysis of variance (ANOVA) test because the sample are not normally dis-

Table 1: Percentile rank means and standard deviations for each method over iterations.

Method	Syntharch		"Retrieved"		"Pivot"	
Iteration	mean	stdev	mean	stdev	mean	stdev
0	61.27%	0.2647	61.22%	0.2733	54.21%	0.2930
1	64.66%	0.2594	64.14%	0.2576	52.43%	0.2709
2	65.60%	0.2527	63.79%	0.2633	50.93%	0.2863
3	68.75%	0.2538	66.22%	0.2607	50.12%	0.2817
4	69.28%	0.2515	69.21%	0.2583	51.58%	0.2777
5	69.38%	0.2437	68.84%	0.2656	54.91%	0.2660
6	70.60%	0.2437	67.51%	0.2742	53.36%	0.2749
7	71.47%	0.2523	67.69%	0.2759	55.36%	0.2839
8	71.33%	0.2563	69.23%	0.2651	57.00%	0.2805
9	70.01%	0.2591	68.06%	0.2610	56.50%	0.2796
10	71.30%	0.2593	69.57%	0.2551	56.19%	0.2862
11	71.90%	0.2538	69.77%	0.2575	56.59%	0.2911

Table 2: Pairwise comparison p-values of the final percentile rank of the three methods, using Nemenyi multiple comparison test.

	Syntharch	"retrieved"	"pivot"
Syntharch	1.000	–	–
"retrieved"	0.6907	1.000	–
"pivot"	0.0007599	0.01283	1.000

tributed, according to Table 1 and 3. The Friedman Test rejects H_0 and indicates statistical significance among the three methods with $\chi^2(2) = 14.69$ and $p = 6.462\times10^{-4} < 0.001$. To examine differences between the three methods, we then perform the Nemenyi post-hoc test for multiple joint samples. The pair-wise p-values are shown in Table 2, where

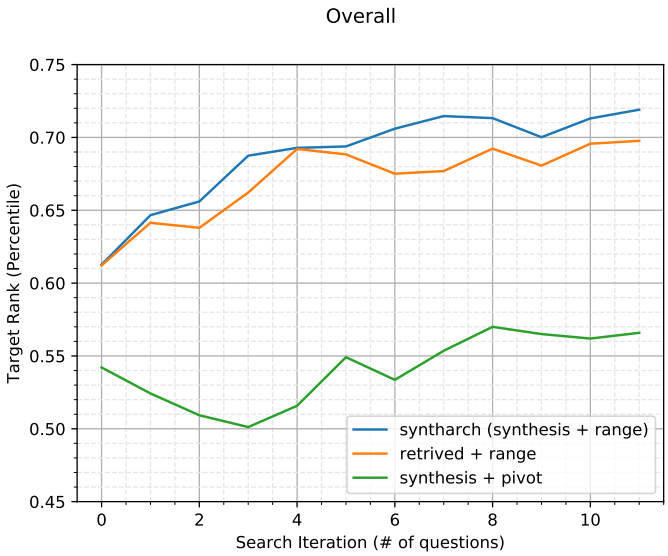

Figure 12: Average percentile rank over iterations grouped by the method.

darker shade indicates higher significance. In particular, we fail to reject the null hypothesis that Syntharch's final percentile rank is similar to than that of the "retrieved" method ($p = 0.6907 > 0.5$), likely due to the limited number of samples from the experiment. On the other hand, the difference between Syntharch and the "pivot" method is highly significant ($p = 7.599 \times 10^{-4} < 0.001$).

4.5 QUALITATIVE RESULTS

In this section, I analyze the overall search performance of the three methods, examine some real search sessions, and suggest possible reasons for the failure cases.

As shown in Figure 12, 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. In each specific 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[3]."

[3]In the sense that the search system will be confused." This name does not imply that the user making

Table 3: Percentile rank at three quantiles for each method over iterations.

M.	Syntharch			"Retrieved"			"Pivot"		
It.	Q_1	Q_2	Q_3	Q_1	Q_2	Q_3	Q_1	Q_2	Q_3
0	46.19%	65.03%	82.69%	40.97%	67.00%	80.83%	31.99%	57.48%	78.61%
1	50.07%	71.39%	85.45%	42.68%	71.56%	86.37%	35.35%	54.55%	75.66%
2	50.40%	69.22%	85.18%	45.65%	64.09%	85.28%	26.78%	52.12%	74.95%
3	54.65%	75.20%	89.77%	50.00%	72.39%	85.85%	21.61%	54.37%	71.36%
4	50.07%	75.39%	88.85%	55.77%	79.24%	88.98%	29.48%	52.49%	73.64%
5	52.89%	75.09%	90.08%	50.70%	78.15%	90.38%	34.56%	55.95%	79.07%
6	53.24%	76.74%	90.75%	50.15%	76.53%	90.12%	31.15%	57.64%	78.25%
7	58.03%	79.17%	92.22%	51.43%	78.00%	88.94%	30.07%	59.34%	81.03%
8	55.48%	79.41%	93.87%	49.59%	78.61%	90.91%	33.61%	60.35%	82.41%
9	51.87%	77.74%	90.85%	50.64%	75.07%	90.43%	32.69%	59.42%	81.47%
10	58.17%	79.79%	91.18%	50.72%	75.25%	92.15%	31.49%	60.30%	84.08%
11	58.17%	79.92%	91.93%	55.45%	75.50%	91.09%	31.48%	59.45%	85.16%

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 at certain iterations, we will observe a decline of average percentile rank (e.g. from iteration[4] 8 to 9 in Syntharch, from iteration 4 to 6 in the "retrieved" method and from iteration 8 to 10 in the "pivot" method).

To understand the conditions of receiving confusing feedback during the interactive search using various methods, we look at some specific search sessions where aggressive percentile rank declines take place.

During the search session shown in Figure 13, there is a sharp declining trend of percentage rank for the "retrieved" method starting from iteration 2. Moreover, the Syntharch

the comparison is at fault.

[4]All iteration indices in this thesis use 0-based numbering.

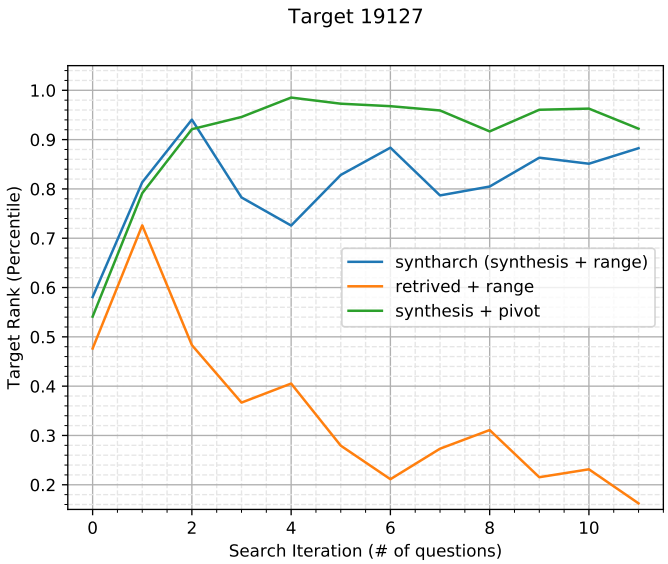

Figure 13: Percentile rank over iterations by the method for one specific search target, showing issues with the "retrieved" method.

method also experiences rank regression starting from iteration 3.

Figure 14 shows the feedback options and the user's choices (marked by [*]) from these three search sessions, reconstructed from the saved log. One immediate observation we make is that during the session with the "retrieved" method, the feedback options remain the same after iteration 3. 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 3, $y_1 = (-1/3, -1/3, 1/3, 0.5), y_2 = (-1/3, -1/3, 1/3, -0.5)$, 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 instructed to make an arbitrary selection, meaning that they will provide confusing feedback half of the time.

On the other hand, while both the Syntharch and "pivot" methods 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 3, the quality

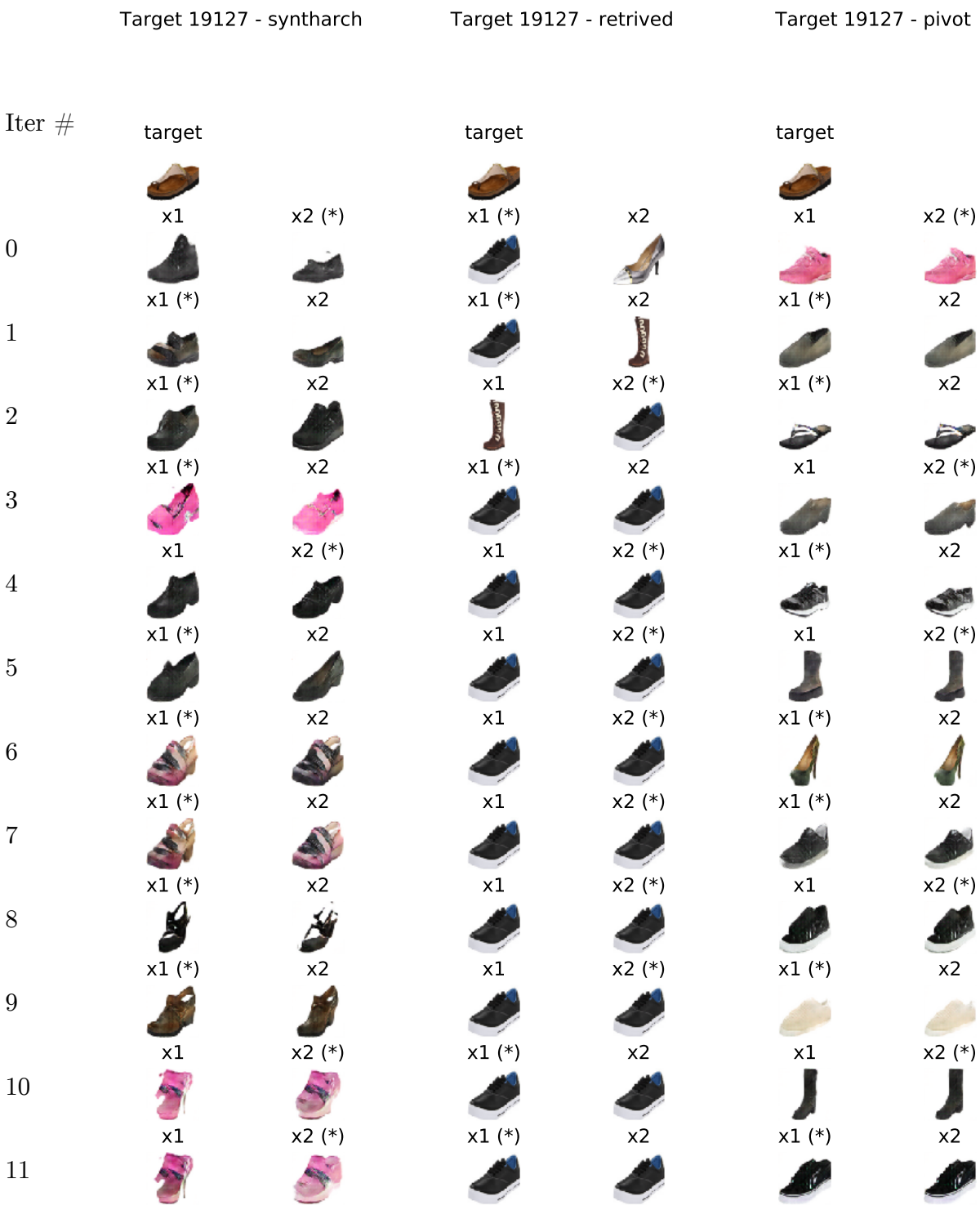

Figure 14: Search session histories of the same target with three different methods.

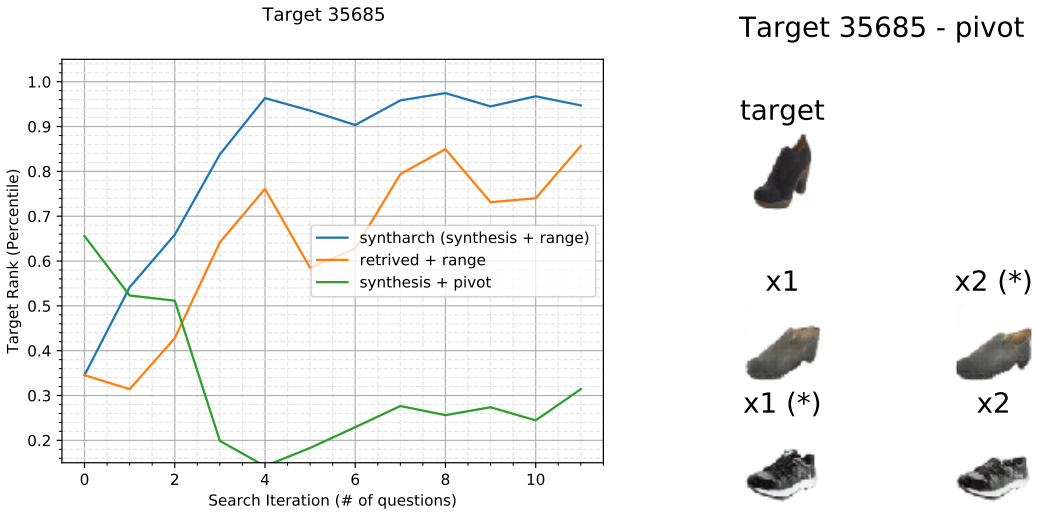

Figure 15: Percentile rank over iterations with iteration 3 and 4 of the "pivot" method for one specific search target.

of the options are too bad for the user to provide 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 13.

I think the reason that the "pivot" method 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, the "pivot" method 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 15, the "pivot" method suffers harshly at iteration 3 and 4. The feedback questions and the user's choices for the two iterations are shown to the right of the percentile rank plot. At iteration 3, 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 11), the expressions of "comfort" of the shoes for the

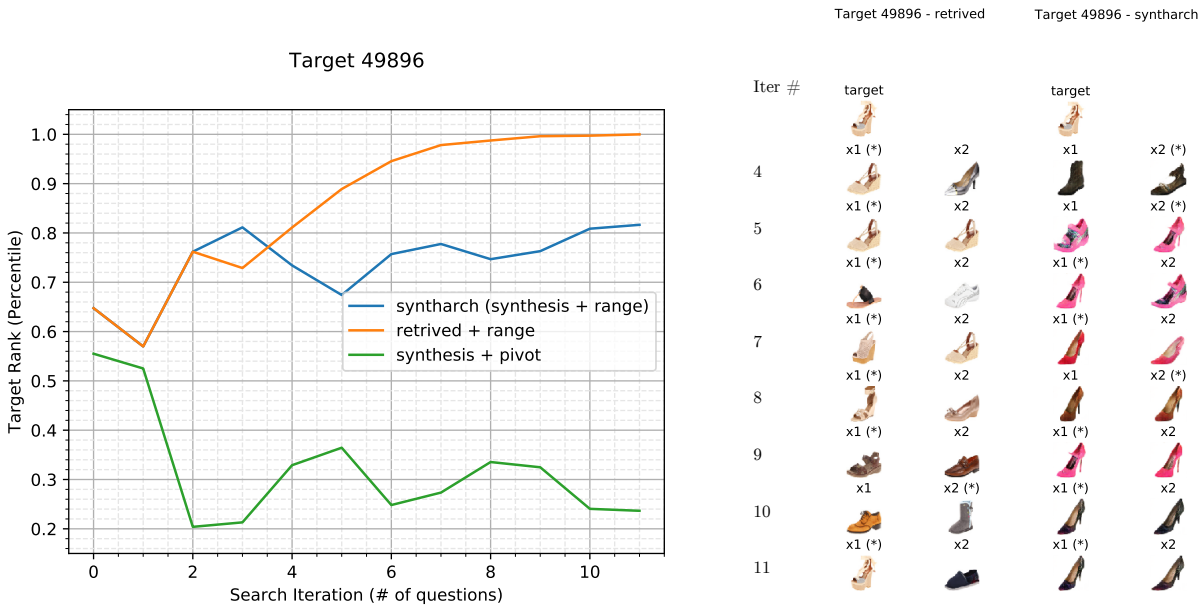

Figure 16: Percentile rank over iterations with partial search iterations of the "retrieved" and Syntharch method for one specific search target.

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 $E_y(x_{35685})^{(4)}$. The similar pattern occurred at iteration 4, possibly accounting for the confusing feedback from the user and inferior search performance.

Finally, we look at one search session where the "retrieved" method performed the best. As we can see in Figure 16, both the Syntharch and the "retrieved" method elicited the same feedback information during the first three iterations, resulting in the same percentile rank initially. Starting from iteration 4, the user made a series of informative feedback leading to a gradually improving rank ending near 100% with the "retrieved" method. In contrast, the user made confusing feedback during 3 out of the 8 remaining iterations with the Syntharch method, causing a lower and unstable rank. We notice that during the Syntharch search session, the user avoided low-quality synthesis results. Specifically, during both iteration 4

and iteration 8, the option x_1 looks less realistic than the alternative option x_2, which the user opted for. Both of the choices, however, led to confusing feedback. In comparison, the feedback options from the "retrieved" method successfully gained informative feedback during these two iterations. I noticed that this issue is more severe when the target image is visually distinct among the samples. In such case, the detail of the target image is harder for the generator to reconstruct, and the image editing results are then more likely to be lacking.

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 in all three methods tested in the user study. However, Syntharch still suffers from low-quality synthesis results in certain individual search sessions.

5.0 CONCLUSIONS

Nowadays, interactive image search systems rely predominantly on attribute labels and comparison feedback in the textual form. In contrast, I explored a novel approach using only visual feedback to accomplish the same task. To supplement the change of feedback form, I proposed Syntharch, which incorporates image synthesis and range searching to achieve better accuracy, as a proof of concept for the new approach. The user study results confirmed the hypotheses in Syntharch that (1) using image editing 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. However, it is also shown in the overall percentile-rank-over-iteration plot in Figure 12 that even the Syntharch method, with both improvements, suffers from "confusing feedback" and experiences regression of average percentile rank in one later iteration. I will discuss some limitations of Syntharch in Section 5.1. Next, in Section 5.2, I will suggest some directions Syntharch can be expanded to overcome some of the limitations and to address a more generalized problem.

5.1 LIMITATIONS

As noted in Section 4.5, the generator might produce low-quality synthesis results in regions of the attribute space where real images are sparse, limiting the overall search efficiency. There are two factors contributing to this issue: the quality of the image editing result, and the distribution of the attribute space which is formed during the concurrent training of the generator and the ranker. The former can be improved with more sophis-

ticated image editing approaches (to be discussed in Section 5.2). However, the latter is harder to change given the same pairwise comparison labels with the goals of decorrelating the attributes and normalizing each dimension to $\mathcal{U}(-1, 1)$. To construct an attribute space where the real images distribute more uniformly within, we will need to select the attributes ourselves.

Using relative attribute labels from the dataset not only makes it harder to learn a more uniform attribute space, but also qualifies the generalizability of Syntharch. Most image databases don't have relative attribute labels for their images. In fact, because Syntharch uses exclusively visual feedback, it does not need semantic attributes which can be understood by the user in words. Conventionally, semantic attributes for each search database are usually chosen by domain experts. However, the task of choosing nonsemantic attributes can be possibly automated with the objective to maximize separation. This idea is similar to converting a regression task to a dimensionality reduction task if class labels are not needed. And if the attributes can be learned in such way, then the preprocessing module of Syntharch can be completely unsupervised that given any database of (unlabeled) images, regardless of the content, we can perform interactive image searches on them.

5.2 FUTURE WORK

To further increase the accuracy and the efficiency of Syntharch, we first explore options to improve the image editing qualify in order to address issues with searching in sparse regions. One way to do that is via a hybrid approach proposed by Zhu et al. in [45]. Specific, they explored using the generator output as a constraint for image manipulation operations through morphing. Because the resulting images are not directly generated by rather morphed from real images, the manipulation outputs can be photo-realistic. Nevertheless, this approach might not work well with images with fine details and complex backgrounds, as image morphing will likely distort them. Other possible options include using different conditioning augmentation methods for the CGAN and adding a second-stage GAN to improve generator quality [41].

Another approach is to search within a non-uniformly distributed attribute space with a modified search method. Specifically, we recognize in order to proceed with the search efficiently, we want to maximize the information gain for each search iteration. For that purpose, when working with a non-uniformly distributed attribute space, the origin of search should be at the center of the image samples as distributed in the attribute space. And for each iteration, instead of reducing the search space by 1/3 of the range on some attribute m, we can reduce the search space by a fraction so that the number of samples distributed on the attribute dimension m is reduced by 1/3. This also means that certain dimensions should possibly be prioritized by the search, and that the round-robin approach to loop through all attribute dimensions would be inefficient. For each search iteration, when choosing the dimension to elicit feedback on for the highest expected information gain, we can follow the method proposed in [15].

There are also many ideas for automating the attribute selection process. Parikh and Grauman have studied building a discriminative vocabulary of semantic attributes [22]. Their approach uses discrete class labels to find attributes which can help discriminate among classes that are the most confused, and then pass them to a nameability model and suggest to humans for review. For our purpose, since the attributes do not necessarily have to be nameable, we can just take the top few attributes that are the most discriminative to start the training. This method, however, still requires binary class labels to bootstrap the attribute selection process. Alternatively, before training the RankCGAN networks, we can first use the images to learn a GAN which only has the latent noise vector as its image. Once that is completed, we can perform dimensionality reduction on the latent space to build the attribute vocabulary.

Finally, we observe that while Syntharch was proposed as a solution for the CBIR task, it can also be applied to the image browsing task for which the user wants to retrieve a set of images of certain characteristics that are not necessarily known beforehand rather than one specific target image of fixed attribute values. For the image browsing task, Syntharch can still use the relevance feedback to guide the browsing results towards certain directions. We can also further relax the tolerance, reducing the search space by a smaller fraction in each iteration to support the discovery and exploration of new attribute expressions. In addition,

because some attributes might be less relevant in retrieving the set of images the user wishes to browse (e.g. the user might care less about some attributes), we can allow the user to initiate the search by selecting an attribute dimension (represented by a pair of contrasting images) to refine on, similar to the mixed initiative search idea in [21].

BIBLIOGRAPHY

[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*, ser. ICML '05. New York, NY, USA: ACM, 2005, pp. 89–96. [Online]. Available: http://doi.acm.org/10.1145/1102351.1102363

[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*, vol. 9, no. 1, pp. 20–37, Jan 2000. [Online]. Available: https://dx.doi.org/10.1109/83.817596

[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*, ser. MM '08. New York, NY, USA: ACM, 2008, pp. 729–732. [Online]. Available: http://doi.acm.org/10.1145/1459359.1459471

[4] M. Ferecatu and D. Geman, "A statistical framework for image category search from a mental picture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1087–1101, June 2009. [Online]. Available: https://dx.doi.org/10.1109/TPAMI.2008.259

[5] 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*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 29–38. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357061

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[7] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Feris, "Dialog-based interactive image retrieval," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 678–688. [Online]. Available: http://papers.nips.cc/paper/7348-dialog-based-interactive-image-retrieval.pdf

[8] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. [Online]. Available: https://dx.doi.org/10.1109/MCSE.2007.55

[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)*, July 2017, pp. 5967–5976. [Online]. Available: https://dx.doi.org/10.1109/CVPR.2017.632

[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*, vol. 4, no. 3, pp. 184–198, 05 1997. [Online]. Available: https://dx.doi.org/10.1136/jamia.1997.0040184

[11] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 694–711. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-46475-6_43

[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)*, July 2017, pp. 7006–7015. [Online]. Available: https://dx.doi.org/10.1109/CVPR.2017.741

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. [Online]. Available: https://arxiv.org/abs/1312.6114

[14] D. Kingma and L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, F. Amsterdam Machine Learning lab (IVI, Ed. arXiv.org, 2015, urn:nbn:nl:ui:29-1.505367. [Online]. Available: https://dx.doi.org/11245/1.505367

[15] A. Kovashka and K. Grauman, "Attribute pivots for guiding relevance feedback in image search," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 297–304. [Online]. Available: https://dx.doi.org/10.1109/ICCV.2013.44

[16] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Interactive image search with relative attribute feedback," *International Journal of Computer Vision*, vol. 115, no. 2, pp. 185–210, Nov 2015. [Online]. Available: https://dx.doi.org/10.1007/s11263-015-0814-0

[17] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2539–2547. [Online]. Available: http://papers.nips.cc/paper/5851-deep-convolutional-inverse-graphics-network.pdf

[18] 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)*, June 1998, pp. 24–28. [Online]. Available: https://dx.doi.org/10.1109/IVL.1998.694480

[19] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. DENOYER, and M. A. Ranzato, "Fader networks:manipulating images by sliding attributes," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5967–5976. [Online]. Available: http://papers.nips.cc/paper/7178-fader-networksmanipulating-images-by-sliding-attributes.pdf

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

[21] 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*, 2018, p. 310. [Online]. Available: http://bmvc2018.org/contents/papers/0151.pdf

[22] D. Parikh and K. Grauman, "Interactively building a discriminative vocabulary of nameable attributes," in *CVPR 2011*, June 2011, pp. 1681–1688. [Online]. Available: https://dx.doi.org/10.1109/CVPR.2011.5995451

[23] D. Parikh and K. Grauman, "Relative attributes," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 503–510. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126281

[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*. MIT Press, 1999, pp. 61–74.

[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. [Online]. Available: http://arxiv.org/abs/1511.06434

[28] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceedings of The 33rd International Conference*

on Machine Learning, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1060–1069. [Online]. Available: http://proceedings.mlr.press/v48/reed16.html

[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*, vol. 8, no. 5, pp. 644–655, Sep. 1998. [Online]. Available: https://dx.doi.org/10.1109/76.718510

[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. [Online]. Available: http://bmvc2018.org/contents/papers/0534.pdf

[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*, vol. 52, no. 5, pp. 783 – 800, 2016. [Online]. Available: https://dx.doi.org/10.1016/j.ipm.2016.03.001

[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*, June 1997, pp. 2–9. [Online]. Available: https://dx.doi.org/10.1109/IVL.1997.629714

[33] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image ranking and retrieval based on multi-attribute queries," in *CVPR 2011*, June 2011, pp. 801–808. [Online]. Available: https://dx.doi.org/10.1109/CVPR.2011.5995329

[34] M. J. Swain, "Interactive indexing into image databases," in *Storage and Retrieval for Image and Video Databases*, vol. 1908. International Society for Optics and Photonics, 1993, pp. 95–104. [Online]. Available: https://dx.doi.org/10.1117/12.143659

[35] B. Thomee and M. S. Lew, "Interactive search in image retrieval: a survey," *International Journal of Multimedia Information Retrieval*, vol. 1, no. 2, pp. 71–86, Jul 2012. [Online]. Available: https://dx.doi.org/10.1007/s13735-012-0014-4

[36] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 776–791. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-46493-0_47

[37] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1099–1107. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969239.2969362

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

[39] ——, "Semantic jitter: Dense supervision for visual comparisons via synthetic images," in *International Conference on Computer Vision (ICCV)*, Oct 2017.

[40] ——, "Thinking outside the pool: Active training image creation for relative attributes," *CoRR*, vol. abs/1901.02551, 2019. [Online]. Available: http://arxiv.org/abs/1901.02551

[41] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 5908–5916. [Online]. Available: https://dx.doi.org/10.1109/ICCV.2017.629

[42] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, no. 6, pp. 536–544, Apr 2003. [Online]. Available: https://dx.doi.org/10.1007/s00530-002-0070-3

[43] 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*, ser. CIVR '08. New York, NY, USA: ACM, 2008, pp. 571–580. [Online]. Available: http://doi.acm.org/10.1145/1386352.1386436

[44] 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*, ser. MM '16. New York, NY, USA: ACM, 2016, pp. 754–756. [Online]. Available: http://doi.acm.org/10.1145/2964284.2973834

[45] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 597–613. [Online]. Available: https://dx.doi.org/10.1007/978-3-319-46454-1_36