Multi Module for Composed Image Retrieval Systems

Binh H. Nguyen, Nam D. Mai, and Hung N. Trinh

Abstract—Image retrieval systems have seen considerable advances in the last decade, with progress primarily focuses on text-based and content-based methods. However, the inputs to these systems are mainly uni-model, queries are expressed as either texts or images, leaving little to no works are done in the situation of multi-model inputs. This paper considers the problem of retrieving images where the input is a combination of texts and images. We propose an autoencoder-based model in which, retaining the deep metric learning approach as in ComposeAE, we present a more efficient model that projects the text-image composition close to the target image in real embedding space, rather than complex space as per ComposeAE. Our method manages to outperform the state-of-the-art method ComposeAE on the challenging benchmark Fashion IQ dataset.

Index Terms—Image retrieval, compositional learning, deep metric learning, autoencoder

1 Introduction

THE field of image retrieval has been studied for over two decades and has undergone major development. Traditional techniques in retrieving images from a database include text-based and content-based image retrieval (CBIR) systems. Text-based image retrieval systems attempt to annotate the images, that is, to add metadata to each image and use this metadata to perform the search procedure. CBIR automates the annotation process by employing computer vision techniques to analyze the "content" of the images and extract the notable features.

Both techniques, however, provide only two input-output models, either image-to-image or text-to-image models. In image-to-image models, the users specify one or more source images as input and the system will return a result image that is similar to the sources. In text-to-image models, the users specify the desired features of the image in the form of text queries, and the system will retrieve the images that have those desired features. This limitation in the form of input to the system prevents the users from fully express their needs which are very diverse and complex. In [1], Lai et al. proposed a user-oriented CBIR system that combines users' interaction and subjective evaluation with the images' intrinsic features to provide a better-fitted result to users' preferences. This improves the relevance of the results but does not provide the users enough freedom in making queries. In many practical scenarios, the user already has a referenced image, he or she needs to find a similar image that has some modification to the original features. The most well-explained example is in fashion shopping where the customer wants to find apparels that are similar to his/her referenced apparel but have some additional features. This introduces a new problem where the input is cross-modal: a source image plus a text string that describes additional features to the source image.

Researches on text-image composition have not been conducted extensively. Vo et al.'s Text Image Residual Gating (TIGR) [2] was the first attempt to tackle this problem. The ComposeAE model proposed by Anwaar et al [3]. improved Vo's model and had achieved state-of-the-art status in this task. Despite the success, we found that the results can be significantly improved by replacing some elements.

First, we reviewed the results of the TIGR model. As justified by Anwaar et al.'s experiments, the method was too focused on adjusting the image space and did not give the text query its important role. TIRG overrated the importance of the query image features and put them directly into the final composed representation. Also pointed out by Anwaar et al., the use of LSTM as the text feature extractor could lead to poor performance in real-world scenarios, where the text queries are usually long and complex.

ComposeAE used a pre-trained BERT model for text feature extracting tasks rather than LSTM. This created better feature vectors for these text queries. To extract features of query images, ComposeAE fine-tuned the ResNet-18 backbone. The most remarkable change in this approach was that extracted features are mapped into a complex space. They solved the problem based on the assumption that the target image's representation is an element-wise rotation of the source image's representation in this complex space, with text features specify the degree of rotation.

The approach of ComposeAE is not a perfect solution in some cases. We think that the assumption of symmetric source-target transition is the core problem. In many cases, the source text only describes how the target image is like, without providing many comparisons between source and target images. This is a problem because the conjugation of the coordinate-wise complex rotations determined from this text query will hardly be able to encode the reverse transition from target images to source images.

ComposeAE used the pre-trained BERT model for text feature extracting rather than LSTM. This created better feature vectors for these text queries. To extract feature vectors of the query images, ComposeAE fine-tuned the pre-trained Resnet-18 backbone. The most remarkable change in this approach is that the composed features and the target image features are mapped into a complex space. They solved the problem based on the assumption that the target image's representation is an element-wise rotation of the source image's representation in this complex space, with text features specify the degree of rotation.

In this work, we assume the modified image and target image are in the same vector space. Our approach is to learn how to generate the modified image features based on text query and to learn the embedding space where query and matching target is close. The target image is the combination of the query image's non-modified features and the modified features calculated by the text and image query. We propose a model that is heavily inspired by the autoencoder-based model of ComposeAE. The preserved features are computed by an element-wise product of the query image features with one minus the attention mask calculated from a stack of two-layer attention in Attention Network. In this way, the attention mask helps represent which parts of the image need to be retained. The image feature maps of the query image are passed through some FiLM-ed ResBlocks. These blocks enable us to influence the modified feature vector using the text query's feature vector before feeding it to a linear module. We reuse the standard deep metric learning method - batch-based loss to learn the mapping of the composed text-image representation closer to the target images and further to negative samples.

We evaluate the performance of our model on the two benchmark datasets: MIT-States [4] and Fashion IQ [5]. In Sec. 4.3, the result shows that our method can learn a better text-image composition and outperforms the state-of-the-art method on Fashion IQ. It is important to show that the ComposeAE model's performance can be enhanced by using a pre-trained model RoBERTa for extracting text features rather than BERT. The results are shown in Table. 5. This suggests that the better the query embedding vector, the more efficient the model.

In general, our key contributions to this task are as below:

1) We propose a new approach based on old components to learn a better-composed representation for text and image features.
2) We observe that the composed representation is affected by the quality of the query embedding vector.
3) Our model outperforms prior works on the Fashion IQ dataset. The performance on R@50 and R@100 metric is better around 7% than ComposeAE with BERT and better around 4% than ComposeAE with RoBERTa.

2 Related Works

2.1 Deep Metric Learning

Metric learning has been used in numerous deep learning studies in recent years [6], [7], [8], [9]. This approach aims to learn a better representation for each object based directly on a distance metric so that reduces the metric distance between similar objects and conversely. In recent years, deep learning and metric learning have been mixed together to introduce the concept of deep metric learning. Deep metric learning utilizes deep network architectures to make use of a metric space to come up with solutions for solving image retrieval problems. This approach has prominent in many image retrieval tasks because the data this task dealing with is unstructured data.

2.2 Cross-Modal Image Retrieval

Cross-modal retrieval considers the learning of shared representation spaces where information is multimedia data that come from different modalities. In recent year, due to the success of deep neural network in representation learning [10], cross-modal retrieval has attracted the attention of many researcher both on academic and industry. In our scenario, we are dealing with cross-modal image retrieval where the input data include text and image have the different distribution since they has been extracted from different modals.

2.3 Interactive Image Retrieval with Natural Language Queries

Image retrieval has been applied in a wide range of applications such as product retrieval in e-commerce, face verification and recognition, and signature verification. With the prominence of deep learning, image retrieval has been drastically improved in performance. There is no longer the need to apply complex processing on images since the results are relatively perfect. (insert picture) The methods lie on the intersection between computer vision and natural language processing has made significant progress thanks to deep learning. Among previous efforts in image retrieval [11], [12], [13], [14], [15], some other methods tackle a promising task which is Interactive Image Search for retrieving complex scenes.

2.4 Dialog-based Interactive Image Retrieval (DIIR)

DIIR carries out the cross-modal image retrieval task in the dialog context [16]. Instead of retrieving product images based merely on the current input query, they made the system more applicable in the real world by improving the results turn-by-turn with the users' text queries and resulting images in previous steps. They interpret the task as a reinforcement learning problem and emphasize the capability of aggregating historical information to iteratively provide results. In addition, they also introduce the Fashion IQ dataset which we use as our primary benchmark.

2.5 Visual Question Answering

Many methods in the Visual Question Answering domain (VQA) have the similar goal as ours, that is, to attempt to fuse the features of texts and images from multimodal input [17], [18]. We adopted their approach into our model because of this similarity. Feature-wise Linear Modulation (FiLM) [19] is one of the most notable methods. It provides us a way to adaptively influence the modified image features using text features: applying feature-wise affine transformations on image feature maps based on the text features influences. As their state, this transform modulates the per-feature-map distribution of activation based on text features, agnostic to spatial location.

Another method that greatly inspires our work is Stack Attention Network (SAN) [20]. Their approach in the effort of tackling image QA tasks is that apply multi-step reasoning for image QA. According to their work, this approach can be viewed as an extension of the attention mechanism successfully employed in other image tasks [21]. Because

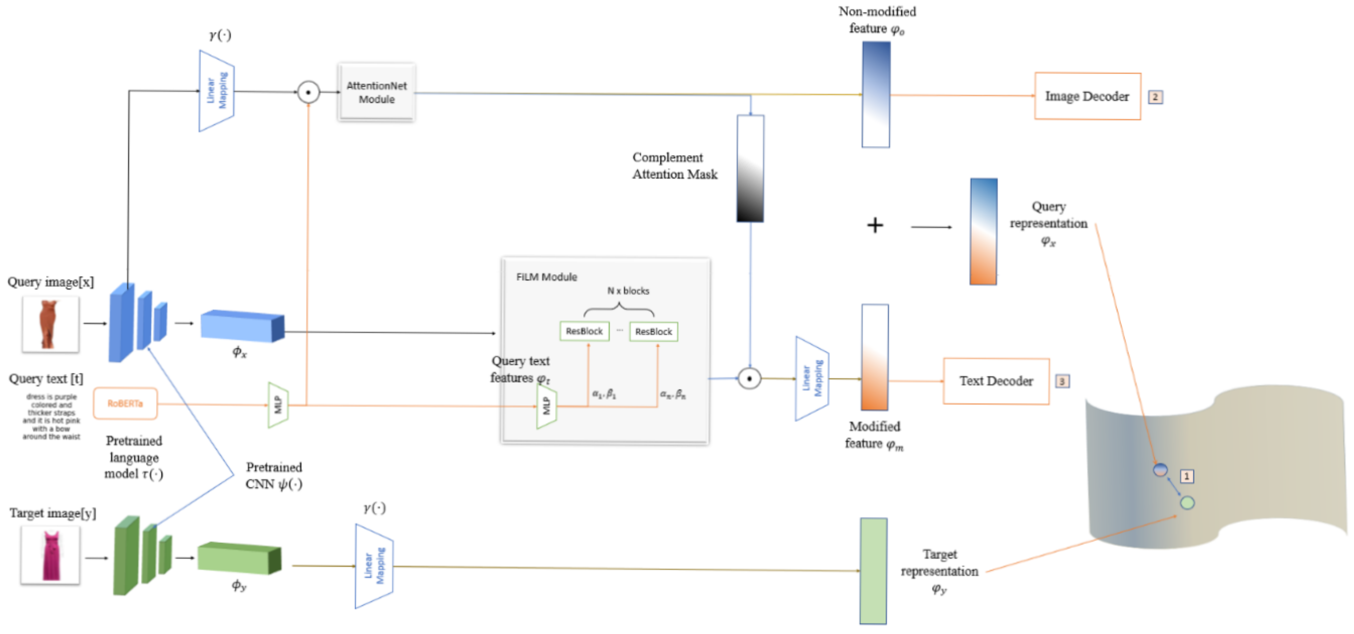
γ(·)
Linear Mapping
AttentionNet Module
Non-modified feature φ_o
Image Decoder
2
Complement Attention Mask
+
Query representation φ_x
Query image[x]
ϕ_x
FiLM Module
N x blocks
ResBlock
ResBlock
Query text features φ_t
MLP
α_1, β_1
α_n, β_n
Linear Mapping
Text Decoder
3
Modified feature φ_m
Query text [t]
dress is purple colored and thicker straps and it is hot pink with a bow around the waist
RoBERTa
MLP
Pretrained language model τ(·)
Pretrained CNN ψ(·)
1
Target image[y]
ϕ_y
γ(·)
Linear Mapping
Target representation φ_y

Fig. 1. Our auto-encoder based model architecture. Here 1 refers to L_B, 2 refers to L_{RI} and 3 refers to L_{RT}. Those losses are defined in Sec. 3.3

of their ability to give an attention distribution focusing on regions that are more likely relevant to the text query, we integrate their stacked attention model as a module for locating non-modified regions of image features.

3 Method

3.1 Problem Formulation

As introduced in Sec. 1, our method learns how to construct the modified image based on the image and text query.

Throughout this paper, x prefers to the query image, y to the target image and t to the query text. Let $\phi_x, \phi_y \in R^{W \times H \times C}$ be the image feature maps extracted from the pre-trained image model of the query and the target image, respectively; and $\gamma : R^{W \times H \times C} \longrightarrow R^d$ be the mapping from the image feature maps to the d-dimension space. Let $\kappa(\cdot, \cdot)$ be the similarity kernel. The task is to construct the modified image from the query image's feature maps, feature vector, and text feature vector, denoted by $c(\phi_x, \gamma(\phi_x), t; \Theta)$. Finally, we learn the embedding space for it and the target image:

$$\max_{\Theta} \kappa(c(\phi_x, \gamma(\phi_x), t; \Theta), \gamma(\phi_y)) \quad (1)$$

where Θ denotes all parameters of our model.

3.2 Network Architectures

Our model as illustrated in Fig. 1 is an autoencoder-based model with two separate decoder blocks inspired by ComposeAE. The image and text queries are encoded in a typical way. For the text query t, we use RoBERTa model $\tau(.)$ to extract the text feature vector living in h-dimensional space. h is 768 in the case of RoBERTa. The reason why we choose RoBERTa is that it has better performance than BERT and has been trained on a large dataset. The benefit of this is that we do not need to spend a lot of time finetune the model

$$\phi_t = \tau(t) \in R^h \quad (2)$$

As in recent methods, we use an image model $\psi(.)$ (e.g. ResNet-18) to extract the feature maps from the query image x as well as the target image y.

$$\phi_x = \psi(x) \in R^{W \times H \times C} \quad (3)$$

where W and H is the width and the height of feature maps and C is the number of feature channels.

We assume that the query and target images are in the same space R. Their feature maps are extracted from the same CNNs, denoted by θ_x and θ_y, respectively. We learn a mapping $\gamma : R^{W \times H \times C} \longrightarrow R^d$ that maps these feature maps to the solution space R^d.

The target image is directly mapped to R^d, denoted by φ_y:

$$\varphi_y = \gamma(\phi_y) \quad (4)$$

where γ is implemented by a pooling layer followed by multi layer perceptron (MLP).

The query image is processed in two branches. The first branch — Stack Attention Network (SAN) [20] — is in charge of mapping non-modified features, denoted by φ_o, to R^d. The second branch is Feature-wise Linear Modulation(FiLM) [19] which is responsible for modifying the distribution of each feature channel in feature maps before mapping them to R^d, denoted by φ_m. The representation of the query image in the solution space R^n is φ_x:

$$\varphi_x = a\varphi_o + b\varphi_m \quad (5)$$

where a and b are learnable parameters.

As in other autoencoder-based models, after learning the representation of the modified image in the solution space,

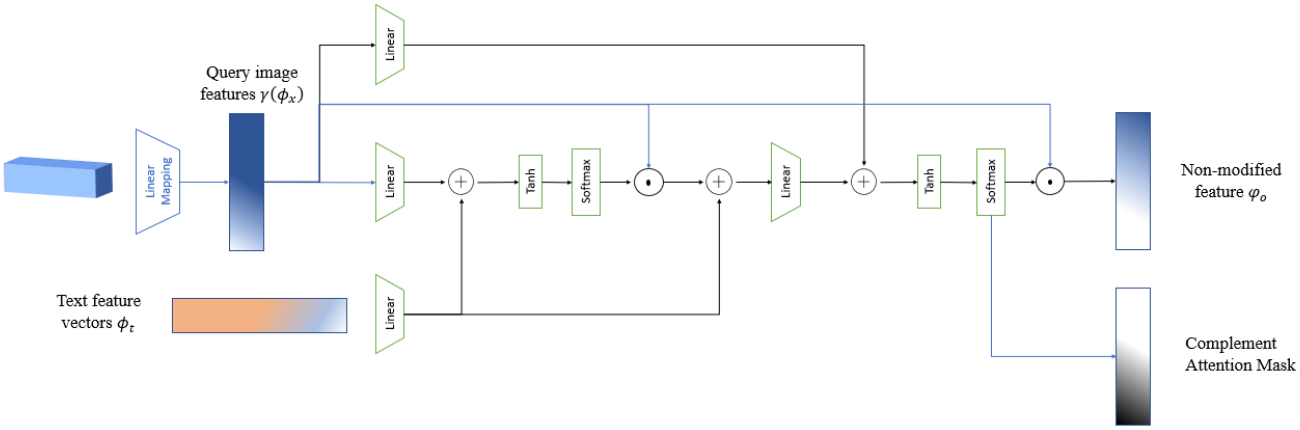

Fig. 2. Stack Attention Network module by stacking two attention layers.

the encoder then learns how to reconstruct the original queries including both image and text by separate decoders. The text decoder and the image decoder are denoted by $d_t(\cdot)$ and $d_i(\cdot)$, respectively. The text decoder aims to reconstruct the feature vector of text query from φ_m, while the image decoder aims to reconstruct the feature vector of the query image from φ_o. The idea of using decoders to reconstruct the queries is what ComposeAE inspires us most. As ComposeAE stated in their work, using separate decoders can help the model avoid overfitting when learning the composed representation φ_x. These decoders also force the model to retain text and image information in φ_x.

3.2.1 *Stack Attention Network*

Stack Attention Network (SAN) does its duty by producing the attention mask that captures regions in the feature vector that is modified by the text query. More formally, the network compute the attention mask m^k of the k-th layer as the following formula:

$$h_A^k = tanh(W_I^k \gamma(\phi_x) + (W_Q^k u^{k-1} + b_Q^k)) \tag{6}$$

$$m^k = softmax(W_A^k h_A^k) \tag{7}$$

where $\gamma(\phi_x) \in R^d$ is the feature vector of the query image. $W_I^k, W_Q^k, W_A^k \in R^{d \times d}$ are in charge of performing linear mapping. $u^{k-1} \in R^d$ is the query vector generated by the previous attention layer, which is computed as:

$$\begin{cases} u^k = m^k \gamma(\phi_x) + u^{k-1} \\ u^0 = \phi_t \end{cases} \tag{8}$$

where $\phi_t \in R^d$ is the feature vector of the query text. Finally, we obtain the non-modified features φ_o by performing element-wise product the image feature vector with one minus the mask generated from the second attention layer of the network:

$$\varphi_o = (1 - m^2) \cdot \gamma(\phi_x) \tag{9}$$

3.2.2 *Feature-wise Linear Modulation*

Feature-wise Linear Modulation (FiLM) allows us to change the distribution of each channel in the feature maps based on the text query through its internal blocks called ResBlocks. More formally, these ResBlocks take two parameters generated from text feature to perform affine transformations on the channels of the feature maps. In this module, we learn the function f and h to generate α, β for each channel of each feature map and perform affine transformation on them through a stack of ResBlocks that return the modified feature maps based on the text query. We obtain the modified feature maps by multiplying with the mask m^2 obtained from SAN module. After that, we learn a mapping $\eta(\cdot,\cdot)$ that map these feature maps to R^n to obtain the modified feature vector φ_m, as:

$$\alpha = f(\phi_t), \beta = h(\phi_t) \tag{10}$$

$$\varphi_m = \eta(m^2 \cdot FiLM(\phi_x | \alpha, \beta), \phi_t) \tag{11}$$

where $\eta(\cdot,\cdot)$ is implemented by concatenating inputs followed by MLP.

3.3 Training and Loss

The training objective is to push closer the features of the "modified" and target image in R^d, and vice versa. That is the reason we use the similarity kernel $\kappa(\cdot,\cdot)$ as the metric.

For a sample i in a training mini-batch of B queries. For this ith sample, we create a set $\mathcal{N}_i$ of K samples including the target sample and $K-1$ negative samples randomly chosen from mini-batch. This procedure is repeated M times to capture any possible set denoted by $\mathcal{N}_i^m$. Let φ_{x_i} be the composed feature of the text and image query, φ_{y_i} be the feature vector of the target corresponding, $\varphi_{\tilde{y}_i}$ be the feature vector of the random negative sample from this mini-batch.

As in TIRG, we define batch-based classification L loss as:

$$L_B = \frac{-1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} log \frac{\exp \kappa(\varphi_{x_i}, \varphi_{y_i}))}{\sum_{\varphi_{y_i} \in \mathcal{N}_i^m} \exp \kappa(\varphi_{x_i}, \varphi_{y_i})} \tag{12}$$

We use this batch-based classification loss in training on both Fashion IQ and MIT-States datasets.

We also add the text and image reconstruction losses which act as regularizers for our model while learning the composed feature vector. The image reconstruction loss is defined as:

$$L_{RI} = \frac{1}{B}\sum_{i=1}^{B} ||\gamma(\phi_x) - d_i(\varphi_o)||_2^2 \tag{13}$$

The text reconstruction loss is defined as:

$$L_{RT} = \frac{1}{B}\sum_{i=1}^{B} ||\phi_t - d_t(\varphi_m)||_2^2 \tag{14}$$

where $\gamma(\cdot)$ is the mapping from feature maps to feature vector. Both $\gamma(\cdot)$, $d_t(\cdot)$ and $d_i(\cdot)$ are implemented by MLP.

The total loss is calculated as:

$$L = L_B + \alpha_{RT} * L_{RT} + \alpha_{RI} * L_{RI} \tag{15}$$

where α_{RT} and α_{RI} are hyperparameters.

4 Experiment

4.1 Setup

We test our model on two datasets: MIT-States, Fashion IQ. Recall at rank k (R@k) is used as the metric for evaluation, which represents the percentage of test queries containing the target image within top k retrieved results. The results shown in 2 and 1 are the mean and standard deviation of the performance after we repeated the experiment five times on each dataset.

In our experiment, we use ResNet-18 for image feature extraction which returns image feature maps $\in R^{512 \times 14 \times 14}$. For text feature extraction, we use RoBERTa instead of BERT in prior works, which return the text feature vector $\in R^{768}$. The whole model is implemented by using PyTorch framework. We only need to fine-tune its weights after five epochs warm-up since the image and text models has been trained on large dataset. In training phase, we train our model about 30k iterations with the batch size is 32. Our choice of the optimizer is Stochastic Gradient Descent (SGD) which helps the model converges quickly. In the inference, the model will take both query image and modify text as the input and return the composed representation. For retrieving images, we just find the k-nearest neighbors of this composed representation.

4.2 Datasets

We evaluate the performance of our model on two datasets that have different properties. To make a fair comparison, we use the same training and testing protocol as TIRG. *MIT-States* [4] has 63,440 images from different real-world objects (animals, foods, etc.), which are described by 245 object classes (e.g. cheese, forest, etc.) and modified by 115 attribute classes (e.g. cluttered, melted, etc.). They constructed the modified text query just by an attribute classes. We also use the baseline called Complete Text Query for the MIT-States dataset where the text query includes an attribute classes and the noun of the target object. In Fig. 3 and Fig. 4 show the examples of retrieving image by an attribute classes versus the Complete Text Query.

Fashion IQ [5] has 77,684 diverse fashion images (dresses, shirts, and tops-tees). As introduced in Sec. 1 this dataset includes human-written relative language feedback for similar pairs of images. In total, this dataset contains about 1000 attribute labels, which were further grouped into five attribute types: texture, fabric, shape, part, and style. With a variety of attribute adjectives, the text query of this dataset can describe a more diverse and detailed description for the target image. Fig. 5 shows the examples of a query on the Fashion IQ dataset.

4.3 Results and Discussions

TABLE 1
Retrieval performance (R@k) on the Fashion IQ dataset. The first three lines of results are taken from the ComposeAE work [3]. The best number is in bold and the second best is underlined.

Method	R@10	R@50	R@100
TIRG with complete text query	$3.34^{\pm0.6}$	$9.18^{\pm0.9}$	$9.45^{\pm0.8}$
TIRG with BERT and complete text query	$11.5^{\pm0.8}$	$28.8^{\pm1.5}$	$28.8^{\pm1.6}$
ComposeAE with BERT	$11.8^{\pm0.9}$	$29.4^{\pm1.1}$	$29.9^{\pm1.3}$
ComposeAE with RoBERTa	$11.8^{\pm0.4}$	$32.1^{\pm0.6}$	$32.2^{\pm0.6}$
Ours with RoBERTa	**$14^{\pm0.2}$**	**$36.6^{\pm0.3}$**	**$36.6^{\pm0.3}$**

TABLE 2
Retrieval performance(R@k) on the MIT-States dataset. The first nine lines of results are taken from the TIRG work [2]

Method	R@1	R@5	R@10
Show and Tell	$11.9^{\pm0.1}$	$31.0^{\pm0.5}$	$42.0^{\pm0.8}$
Att. as Operator	$8.8^{\pm0.1}$	$27.3^{\pm0.3}$	$39.1^{\pm0.3}$
Relationship	$12.3^{\pm0.5}$	$31.9^{\pm0.7}$	$42.9^{\pm0.9}$
FiLM	$10.1^{\pm0.3}$	$27.7^{\pm0.7}$	$38.3^{\pm0.7}$
TIRG	$12.2^{\pm0.4}$	$31.9^{\pm0.3}$	$43.1^{\pm0.3}$
TIRG with BERT	$12.3^{\pm0.6}$	$31.8^{\pm0.3}$	$42.6^{\pm0.8}$
TIRG with complete text query	$7.9^{\pm1.9}$	$28.7^{\pm2.5}$	$34.1^{\pm2.9}$
TIRG with BERT and complete text query	$13.3^{\pm0.6}$	$34.5^{\pm1.0}$	$46.8^{\pm1.1}$
ComposeAE	**$13.9^{\pm0.5}$**	**$35.3^{\pm0.8}$**	**$47.9^{\pm0.7}$**
Ours	$9.4^{\pm0.2}$	$27.2^{\pm0.4}$	$38.7^{\pm0.5}$
Ours with complete text query	$8.7^{\pm0.5}$	$28.4^{\pm1.0}$	$41.2^{\pm0.5}$

As shown in Table 2, our model performs poorly on the MIT-States dataset. The reason is that our model gives the text query an important role in the composed representation. It decides which parts of features are preserved and how the query image is modified. Thus, in the MIT-States dataset where the text queries are too short and poor semantic but require modifying a large region in images (e.g. "old", "cooked", etc.). The next problem we stuck in some cases where the image queries are completely different from the target images corresponding to. That goes beyond our approach which assumes that the target images are composed of the original features and the modified features from the image queries.

In the case of Fashion IQ, our model outperforms earlier works. Compared to ComposeAE, the recall at k metric is improved around 7.2% in terms of R@50, 6.7% in terms of R@100, and 2.8% in terms of R@10. The recall at k metric is improved around 4% when it is compared to ComposeAE with RoBERTa. The improvement lies in the complexity of the text queries. Fashion IQ's text queries are more detailed and longer than those in the MIT-States dataset. Fashion IQ's texts have an average length of 13.5 words while the

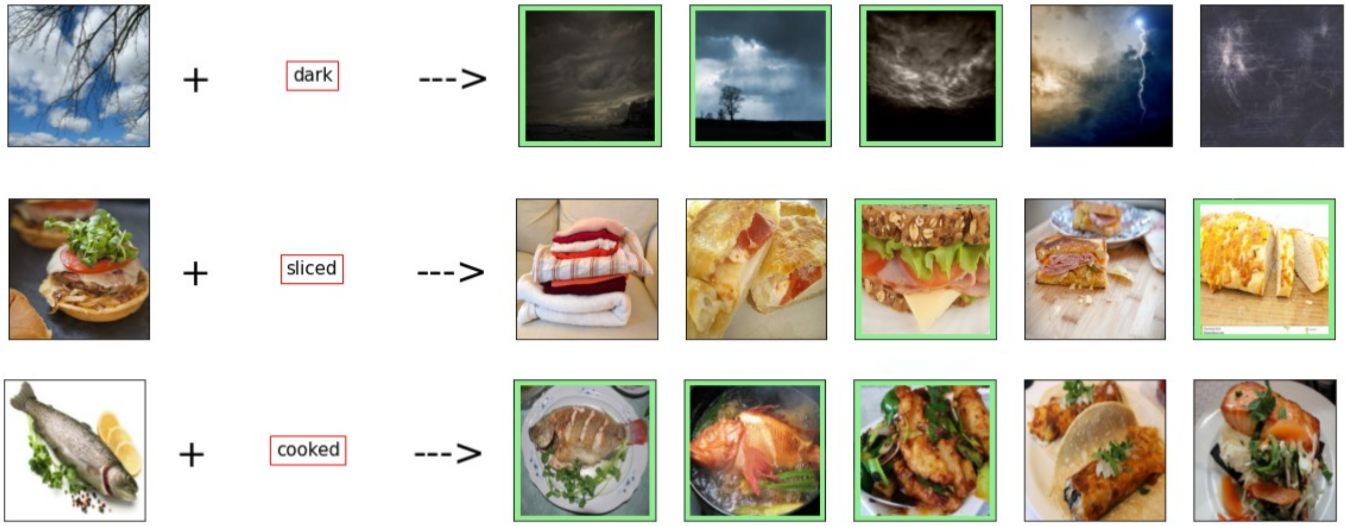

Fig. 3. Retrieval examples by an adjective on the MIT-States dataset.

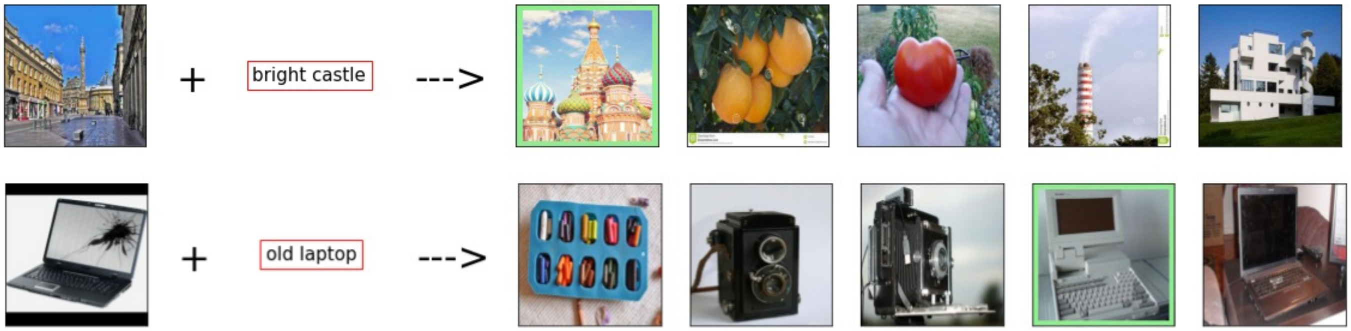

Fig. 4. Retrieval examples by Complete Text Query on the MIT-States dataset.

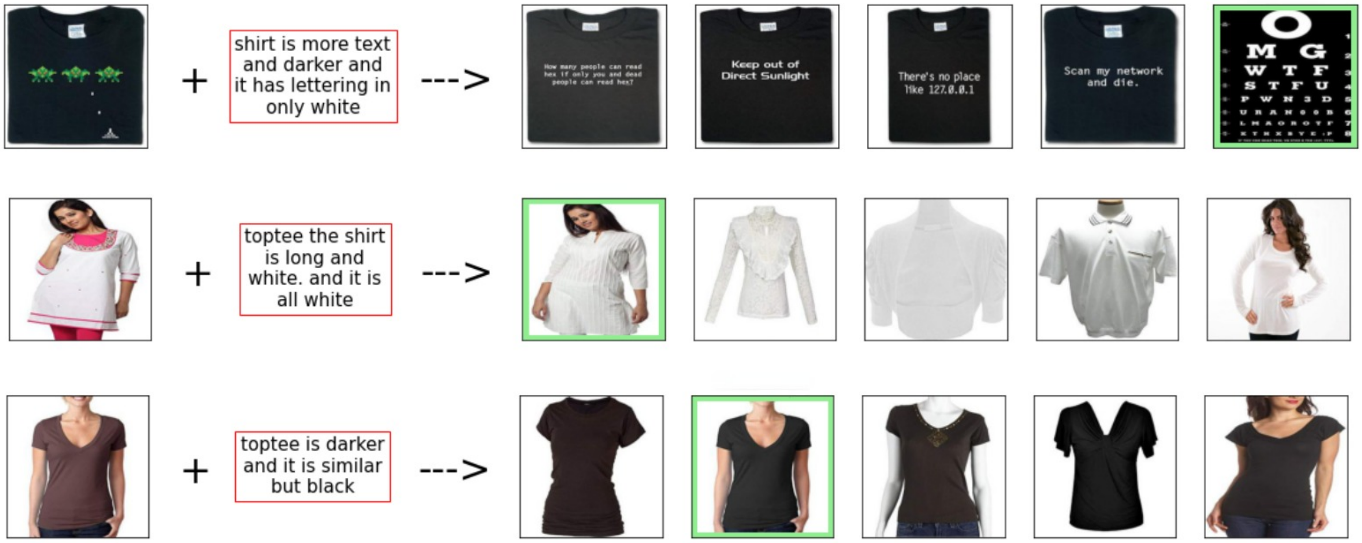
+
shirt is more text and darker and it has lettering in only white
--->
Keep out of Direct Sunlight
There's no place like 127.0.0.1
Scan my network and die.
+
toptee the shirt is long and white. and it is all white
--->
+
toptee is darker and it is similar but black
--->

Fig. 5. Retrieval examples on the Fashion IQ dataset.

average length of complete text queries in the MIT-States is 2. In compared to MIT-States, the query and target images in Fashion IQ are more likely to each other, which supports our assumption a lot.

5 Conclusion

In conclusion, we proposed an approach that reuses old materials to tackle the multi-modal image retrieval problem. We have shown that our model outperforms previous works in the Fashion IQ dataset. By experiment, we observed that the results are better if we use the efficient feature extraction at the beginning.

References

[1] C.-C. Lai and Y.-C. Chen, “A user-oriented image retrieval system based on interactive genetic algorithm,” *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 10, pp. 3318–3325, 2011.

[2] N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, and J. Hays, “Composing text and image for image retrieval - an empirical odyssey,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[3] M. U. Anwaar, E. Labintcev, and M. Kleinsteuber, “Compositional learning of image-text query for image retrieval,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan 2021, pp. 1140–1149.

[4] P. Isola, J. J. Lim, and E. H. Adelson, “Discovering states and transformations in image collections,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[5] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, “Fashion iq: A new dataset towards retrieving images by natural language feedback,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 307–11 317.

[6] M. Kaya and H. S. Bilge, “Deep metric learning: A survey,” *Symmetry*, vol. 11, no. 9, 2019. [Online]. Available: https://www.mdpi.com/2073-8994/11/9/1066

[7] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, “Deep metric learning to rank,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 34–39.

[9] Q. Wang, J. Wan, and Y. Yuan, “Deep metric learning for crowdedness regression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2633–2643, 2018.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[11] A. Kovashka, D. Parikh, and K. Grauman, “Whittlesearch: Image search with relative attribute feedback,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2973–2980.

[12] A. Kovashka and K. Grauman, *Attributes for Image Retrieval*. Cham: Springer International Publishing, 2017, pp. 89–117. [Online]. Available: https://doi.org/10.1007/978-3-319-50077-5_5

[13] ——, “Attribute pivots for guiding relevance feedback in image search,” in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 297–304.

[14] D. Parikh and K. Grauman, “Relative attributes,” in *2011 International Conference on Computer Vision*, 2011, pp. 503–510.

[15] A. Yu and K. Grauman, *Fine-Grained Comparisons with Attributes*. Cham: Springer International Publishing, 2017, pp. 119–154. [Online]. Available: https://doi.org/10.1007/978-3-319-50077-5_6

[16] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. Schmidt Feris, “Dialog-based interactive image retrieval,” *arXiv e-prints*, p. arXiv:1805.00145, Apr. 2018.

[17] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[18] B. Zhou, Y. Tian, S. Sukhbaatar, A. D. Szlam, and R. Fergus, “Simple baseline for visual question answering,” *ArXiv*, vol. abs/1512.02167, 2015.

[19] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11671

[20] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, “Stacked attention networks for image question answering,” *CoRR*, vol. abs/1511.02274, 2015. [Online]. Available: http://arxiv.org/abs/1511.02274

[21] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07-09 Jul 2015, pp. 2048–2057. [Online]. Available: http://proceedings.mlr.press/v37/xuc15.html

Binh H. Nguyen Student at VNUHCM - University of Science

Nam D. Mai Student at VNUHCM - University of Science

Hung N. Trinh Student at VNUHCM - University of Science