Composing Text and Image for Image Retrieval - An Empirical Odyssey

Nam Vo[1]*, Lu Jiang[2], Chen Sun[2], Kevin Murphy[2]
Li-Jia Li[2,3], Li Fei-Fei[2,3], James Hays[1]
[1]Georgia Tech, [2]Google AI, [3]Stanford University

Abstract

In this paper, we study the task of image retrieval, where the input query is specified in the form of an image plus some text that describes desired modifications to the input image. For example, we may present an image of the Eiffel tower, and ask the system to find images which are visually similar, but are modified in small ways, such as being taken at nighttime instead of during the day. To tackle this task, we embed the query (reference image plus modification text) and the target (images). The encoding function of the image text query learns a representation, such that the similarity with the target image representation is high iff it is a "positive match". We propose a new way to combine image and text through residual connection, that is designed for this retrieval task. We show this outperforms existing approaches on 3 different datasets, namely Fashion-200k, MIT-States and a new synthetic dataset we create based on CLEVR. We also show that our approach can be used to perform image classification with compositionally novel labels, and we outperform previous methods on MIT-States on this task.

1. Introduction

A core problem in image retrieval is that the user has a "concept" in mind, which they want to find images of, but they need to somehow convey that concept to the system. There are several ways of formulating the concept as a search query, such as a text string, a similar image, or even a sketch, or some combination of the above. In this work, we consider the case where queries are formulated as an input image plus a text string that describes some desired modification to the image. This represents a typical scenario in session search: users can use an already found image as a reference, and then express the difference in text, with the aim of retrieving a relevant image. This problem is closely related to attribute-based product retrieval (see e.g., [12]), but differs in that the text can be multi-word, rather than a single attribute.

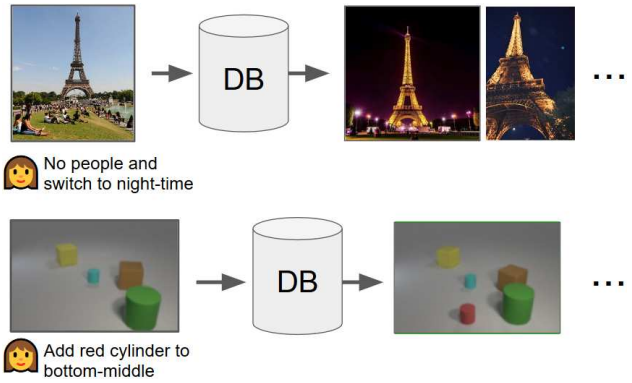

Figure 1. Example of image retrieval using text and image query. The text states the desired modification to the image and is expressive in conveying the information need to the system.

We can use standard deep metric learning methods such as triplet loss (e.g., [15]) for computing similarity between a search query and candidate images. The main research question we study is how to represent the query when we have two different input modalities, namely the input image and the text. In other words, how to learn a meaningful cross-modal feature composition for the query in order to find the target image.

Feature composition between text and image has been extensively studied in the field of vision and language, especially in Visual Question Answering (VQA) [2]. After encoding an image (e.g., using a convolutional neural network, or CNN) and the text (e.g., using a recurrent neural network, or RNN), various methods for feature composition have been used. These range from simple techniques (e.g., concatenation or shallow feed-forward networks) to advanced mechanisms (e.g., relation [43], or parameter hashing [35]). These approaches have also been successfully used in related problems such as query classification, compositional learning, etc. (See Section 2 for more discussion of related work.)

The question of which image/text feature composition to use for image retrieval has not been studied, to the best of our knowledge. In this paper, we compare several existing methods, and propose a new one, which often gives

*Work done during an internship at Google AI.

improved results. The key idea behind the new method is that the text should modify the features of the query image, but we want the resulting feature vector to still "live in" the same space as the target image. We achieve this goal by having the text modify the image feature via a gated residual connection. We call this "Text Image Residual Gating " (or TIRG for short). We give the details in Section 3.

We empirically compare these methods on three benchmarks: Fashion-200k dataset from [12], MIT-States dataset [17], and a new synthetic dataset for image retrieval, which we call "CSS" (color, shape and size), based on the CLEVR framework [20]. We show that our proposed feature combination method outperforms existing methods in all three cases. In particular, significant improvement is made on Fashion-200k compared to [12] whose approach is not ideal for this image retrieval task. Besides, our method works reasonably well on a recent task of learning feature composition for image classification [31, 33], and achieves the state-of-the-art result on the task on the MIT-States dataset [17].

To summarize, our contribution is threefold:

- We systematically study feature composition for image retrieval, and propose a new method.
- We create a new dataset, CSS, which we will release, which enables controlled experiments of image retrieval using text and image queries.
- We improve previous state of the art results for image retrieval and compositional image classification on two public benchmarks, Fashion-200K and MIT-States.

2. Related work

Image retrieval and product search: Image retrieval is an important vision problem and significant progress has been made thanks to deep learning [5, 51, 10, 38]; it has numerous applications such as product search [28], face recognition [44, 36] or image geolocalization [13]. Cross-modal image retrieval allows using other types of query, examples include text to image retrieval [52], sketch to image retrieval [42] or cross view image retrieval [26], and event detection [19]. We consider our set up an image to image retrieval task, but the image query is augmented with an additional modification text input.

A lot of research has been done to improve product retrieval performance by incorporating user's feedback to the search query in the form of relevance [40, 18], relative [23] or absolute attribute [56, 12, 1]. Tackling the problem of image based fashion search, Zhao *et al*. [56] proposed a memory-augmented deep learning system that can perform attribute manipulation. In [12], spatially-aware attributes are automatically learned from product description labels and used to facilitate attribute-feedback product retrieval application. We are approaching the same image search task, but incorporating text into the query instead, which can be potentially more flexible than using a predefined set of attribute values. Besides, unlike previous work which seldom shows its effectiveness beyond image retrieval, we show our method also work reasonably well for a classification task on compositional learning.

Parallel to our work is dialog-based interactive image retrieval [11], where Guo *et al*. showed promising result on simulated user and real world user study. Though the task is similar, their study focuses on modeling the interaction between user and the agent; meanwhile we study and benchmark different image text composition mechanisms.

Vision question answering: The task of Visual Question Answering (VQA) has achieved much attention (see e.g., [2, 20]). Many techniques have been proposed to combine the text and image inputs effectively [7, 34, 22, 35, 37, 43, 27, 48, 25]. Fukui et al [7] proposed Multimodal Compact Bilinear Pooling as a feature fusion mechanism to combine image and text. In [35], the text feature is incorporated by mapping into parameters of a fully connected layer within the image CNN. Another important tool that's proved effective for VQA task is attention [34, 48, 27]. In [22, 37], residual connections are used to combine image and text. Specifically, [22] proposed method outputs the text feature plus a residual mapping obtained by joint element-wise multiplication of image and text. [37] introduced FiLM layer as a way to inject text features into an image CNN, notably by residual connections. While using similar technical components, we actually try to keep and "modify" the input image feature, instead of "fusing" it with text creating a "brand new" feature.

Vision and Language: beside VQA, there's other tasks that also learn to make prediction from image and text input. Chen et al [4] proposed a recurrent attentive model to edit and colorize images given text descriptions. [16, 29, 54, 53] study the referring expression comprehension task, which aim to localize the object in the input image given its reference description.

Compositional Learning: We can think of our query as a composition of an image and a text. The core of compositional learning is that a complex concept can be developed by combing multiple simple concepts or attributes [31]. The idea is reminiscent of earlier work on visual attribute [6, 41] and also related to zero-shot learning [24, 39, 55]. Among recent contributions, Misra *et al*. [31] investigated learning a composition classifier by combining an existing object classifier and attribute classifier. Nagarajan *et al*. [33] proposed an embedding approach to carry out the composition using the attribute embedding as an operator to change the object classifier. Kota *et al*. [21] applied this idea to action recognition. By contrast, our composition is cross-modal and only has a single image versus abundant training exam-

ples to train the classifiers.

3. Method

As explained in the introduction, our goal is to learn an embedding space for the text+image query and for target images, such that matching (query, image) pairs are close (see Fig. 2).

First, we encode the query (or reference) image x using a ResNet-17 CNN to get a 2d spatial feature vector $f_{\text{img}}(x) = \phi_x \in \mathbb{R}^{W\times H\times C}$, where W is the width, H is the height, and $C = 512$ is the number of feature channels. Next we encode the query text t using a standard LSTM. We define $f_{\text{text}}(t) = \phi_t \in \mathbb{R}^d$ to be the hidden state at the final time step whose size d is 512. We want to keep the text encoder as simple as possible. Encoding texts by other encoders, e.g. bi-LSTM or LSTM attention, is definitely feasible but beyond the scope of our paper. Finally, we combine the two features to compute $\phi_{xt} = f_{\text{combine}}(\phi_x, \phi_t)$. Below we discuss various ways to perform this combination.

3.1. Summary of existing combination methods

In this paper, we study the following approaches for feature composition. For a fair comparison, we train all methods including ours using the same pipeline, with the only difference being in the composition module.

- Image Only: we set $\phi_{xt} = \phi_x$.
- Text Only: we set $\phi_{xt} = \phi_t$.
- Concatenate computes $\phi_{xt} = f_{\text{MLP}}([\phi_x, \phi_t])$. This simple has proven effective in a variety of applications [2, 11, 56, 31]. In particular, we use two layers of MLP with RELU, the batch-norm and the dropout rate of 0.1.
- Show and Tell [49]. In this approach, we train an LSTM to encode both image and text by inputting the image feature first, following by words in the text; the final state of this LSTM is used as representation ϕ_{xt}.
- Attribute as Operator [33] embeds each text as a transformation matrix, T_t, and applies T_t to ϕ_x to create ϕ_{xt}.
- Parameter hashing [35] is a technique used for the VQA task. In our implementation, the encoded text feature ϕ_t is hashed into a transformation matrix T_t, which can be applied to image feature; it is used to replace a fc layer in the image CNN, which now outputs a representation ϕ_{xt} that takes into account both image and text feature.
- Relationship [43] is a method to capture relational reasoning in the VQA task. It first uses CNN to extract a 2D feature map from image, then create a set of relationship features, each is a concatenation of the text feature ϕ_t and 2 local features in the 2D feature map; this set of features is passed through an MLP and the result is summed to get a single feature. Another MLP is applied to obtain the output ϕ_{xt}.
- Multimodal Residual Networks (MRN) [22] is a VQA method that uses element-wise multiplication for the joint residual mappings. Here starting with $\phi^0_{xt} = \phi_t$, each of its block layer is defined as $\phi^i_{xt} = \phi^{i-1}_{xt} + fc(tanh(\phi^{i-1}_{xt}))\cdot fc(tanh(fc(tanh(\phi_x))))$. The last feature is linearly transformed to obtain the image text composition output.
- FiLM [37] is another VQA method where the text feature is also injected into the image CNN. In more detail, the text feature ϕ_t is used to predict modulation features: $\gamma^i, \beta^i \in \mathbb{R}^C$, where i indexes the layer and C is the number of feature or feature map. Then it performs a feature-wise affine transformation of the image features, $\phi^i_{xt} = \gamma^i \cdot \phi^i_x + \beta^i$. As stated in [37], a FiLM layer only handles a simple operation like scaling, negating or thresholding the feature. To perform complex operations, it has to be used in every layer of the CNN. By contrast, we only modify one layer of the image feature map, and we do this using a gated residual connection, described in 3.2.

3.2. Proposed approach: TIRG

Inspired by [47, 14, 30], we propose to combine image and text features using the following approach which we call Text Image Residual Gating (or TIRG for short).

$$\phi^{rg}_{xt} = w_g f_{\text{gate}}(\phi_x, \phi_t) + w_r f_{\text{res}}(\phi_x, \phi_t), \quad (1)$$

where $f_{\text{gate}}, f_{\text{res}} \in \mathbb{R}^{W\times H\times C}$ are the gating and the residual features shown in Fig. (2). w_g, w_r are learnable weights to balance them. The gating connection is computed by:

$$f_{\text{gate}}(\phi_x, \phi_t) = \sigma(W_{g2} * \text{RELU}(W_{g1} * [\phi_x, \phi_t]) \odot \phi_x \quad (2)$$

where σ is the sigmoid function, $\odot$ is element wise product, $*$ represents 2d convolution with batch normalization, and W_{g1} and W_{g2} are 3x3 convolution filters. Note that we broadcast ϕ_t along the height and width dimension so that its shape is compatible to the image feature map ϕ_x. The residual connection is computed by:

$$f_{\text{res}}(\phi_x, \phi_t) = W_{r2} * \text{RELU}(W_{r1} * ([\phi_x, \phi_t])), \quad (3)$$

The intuition is that we want to "modify" the query image feature instead of traditional "feature fusion" that creates a new feature from existing ones. This is facilitated by the ResBlock design: the gated identity establishes the input image feature as a reference to the output composition feature, as if they were in the same meaningful image feature space; then the added residual connection represents the modification or "walk" in this feature space.

When training, it essentially starts off as a working image to image retrieval system, then gradually learn meaningful modification. Differently, other methods would start off with random retrieval result at the beginning.

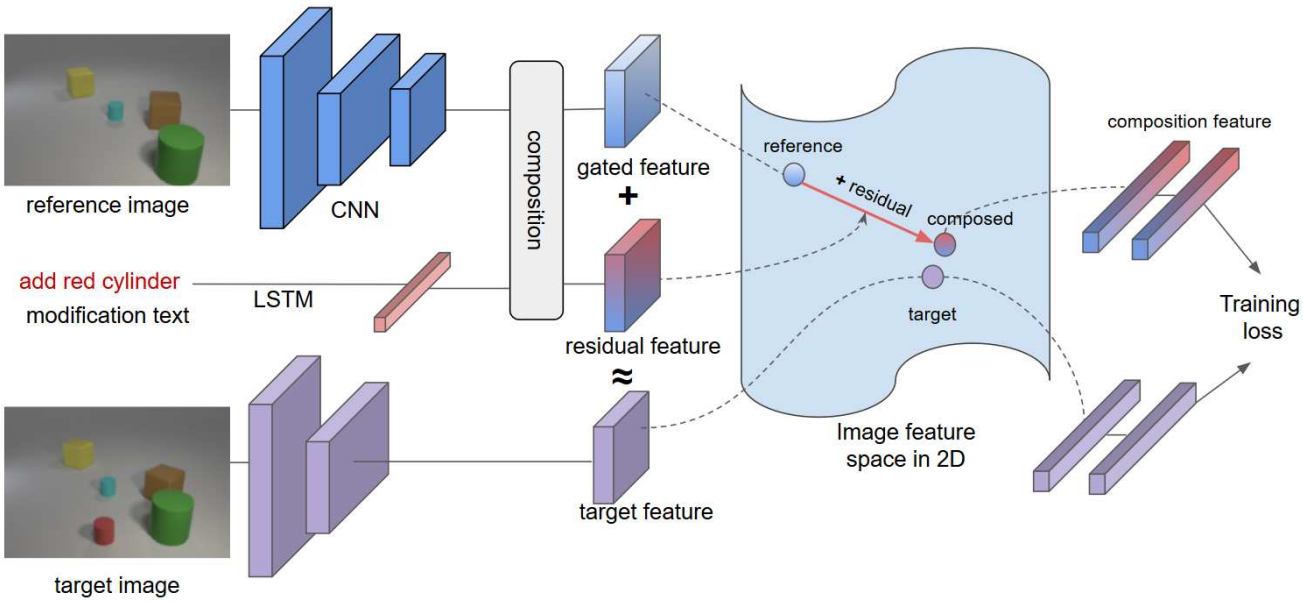

Figure 2. The system pipeline for training. We show a 2d feature space for visual simplicity.

Fig. 2 shows modification applied to the convolutional layer of the CNN. However, we can alternatively apply modification to the fully-connected layer (where $W = H = 1$) to alter the non-spatial properties of the representation. In our experiments, we modify the last fc layer for Fashion200k and MIT-States, since the modification is more global and abstract. For CSS, we modify the last 2D feature map before pooling (last conv layer) to capture the low-level and spatial changes inside the image. The choice of which layer to modify is a hyperparameter of the method and can be chosen based on a validation set.

3.3. Deep Metric Learning

Our training objective is to push closer the features of the "modified" and target image, while pulling apart the features of non-similar images. We employ a classification loss for this task. More precisely, suppose we have a training minibatch of B queries, where $\psi_i = f_{\text{combine}}(x_i^{query}, t_i)$ is the final modified representation (from the last layer) of the image text query, and $\phi_i^+ = f_{\text{img}}(x_i^{target})$ is the representation of the target image of that query. We create a set $\mathcal{N}_i$ consisting of one positive example ϕ_i^+ and $K-1$ negative examples $\phi_1^-, \ldots, \phi_{K-1}^-$ (by sampling from the minibatch ϕ_j^+ where j is not i). We repeat this M times, denoted as $\mathcal{N}_i^m$, to evaluate every possible set. (The maximum value of M is $\binom{B}{K}$, but we often use a smaller value for tractability.) We then use the following softmax cross-entropy loss:

$$L = \frac{-1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} \log\{\frac{\exp\{\kappa(\psi_i, \phi_i^+)\}}{\sum_{\phi_j \in \mathcal{N}_i^m} \exp\{\kappa(\psi_i, \phi_j)\}}\}, \quad (4)$$

where κ is a similarity kernel and is implemented as the dot product or the negative l_2 distance in our experiments. When we use the smallest value of $K = 2$, Eq. (4) can be easily rewritten as:

$$L = \frac{1}{MB} \sum_{i=1}^{B} \sum_{m=1}^{M} \log\{1 + \exp\{\kappa(\psi_i, \phi_{i,m}^-) - \kappa(\psi_i, \phi_i^+)\}\}, \quad (5)$$

since each set $\mathcal{N}_i^m$ contains a single negative example. This is equivalent to the the soft triplet based loss used in [50, 15]. When we use $K = 2$, we choose $M = B - 1$, so we pair each example i with all possible negatives.

If we use larger K, each example is contrasted with a set of other negatives; this loss resembles the classification based loss used in [9, 46, 32, 45, 8]. With the largest value $K = B$, we have $M = 1$, so the function is simplified as:

$$L = \frac{1}{B} \sum_{i=1}^{B} -\log\{\frac{\exp\{\kappa(\psi_i, \phi_i^+)\}}{\sum_{j=1}^{B} \exp\{\kappa(\psi_i, \phi_j^+)\}}\}, \quad (6)$$

In our experience, this case is more discriminative and fits faster, but can be more vulnerable to overfitting. As a result, we set $K = B$ for Fashion200k since it is more difficult to converge and $K = 2$ for other datasets. Ablation studies on K are shown in Table 5.

4. Experiments

We perform our empirical study on three datasets: Fashion200k [12], MIT-States [17], and a new synthetic dataset we created called CSS (see Section 4.3). Our main metric for retrieval is recall at rank k (R@K), computed as the percentage of test queries where (at least 1) target or correct

labeled image is within the top K retrieved images. Each experiment is repeated 5 times to obtain a stable retrieval performance, and both mean and standard deviation are reported. In the case of MIT-States, we also report classification results.

We use PyTorch in our experiments. We use ResNet-17 (output feature size = 512) pretrained on ImageNet as our image encoder and the LSTM (hidden size is 512) of random initial weights as our text encoder. By default, training is run for 150k iterations with a start learning rate 0.01. We will release the code and CSS dataset to the public. Using the same training pipeline, we implement and compare various methods for combining image and text, described in section 3.1, with our feature modification via residual values, described in section 3.2, denoted as **TIRG**.

4.1. Fashion200k

Fashion200k [12] is a challenging dataset consisting of ∼200k images of fashion products. Each image comes with a compact attribute-like product description (such as black biker jacket or wide leg culottes trouser). Following [12], queries are created as following: pairs of products that have one word difference in their descriptions are selected as the query images and target images; and the modification text is that one different word. We used the same training split (around 172k images) and generate queries on the fly for training. To compare with [12], we randomly sample 10 validation sets of 3,167 test queries (hence in total 31,670 test queries) and report the mean.[1].

Table 1 shows the results, where the recall of the first row is from [12] and the others are from our framework. We see that our pipeline even with different kind of composition mechanisms outperforms their approach. We believe this is because they perform image text joint embedding training, instead of attribute-feedback or text-modification image retrieval training. In terms of the different ways of computing ϕ_{xt}, we see that our approach performs the best. Some qualitative retrieval examples are shown in Fig. 3.

4.2. MIT-States

MIT-States [17] has ∼60k images, each comes with an object/noun label and a state/adjective label (such as "red tomato" or "new camera"). There are 245 nouns and 115 adjectives, on average each noun is only modified by ∼9 adjectives it affords. We use it to evaluate both image retrieval and image classification, as we explain below.

[1] We contacted the authors of [12] for the original 3,167 test queries, but got only the product descriptions. We attempted to recover the set from the description. However, on average, there are about 3 product images for each unique product description.

Method	R@1	R@10	R@50
Han *et al*. [12]	6.3	19.9	38.3
Image only	3.5	22.7	43.7
Text only	1.0	12.3	21.8
Concatenation	$11.9^{\pm 1.0}$	$39.7^{\pm 1.0}$	$\underline{62.6}^{\pm 0.7}$
Show and Tell	$12.3^{\pm 1.1}$	$40.2^{\pm 1.7}$	$61.8^{\pm 0.9}$
Param Hashing	$12.2^{\pm 1.1}$	$40.0^{\pm 1.1}$	$61.7^{\pm 0.8}$
Relationship	$13.0^{\pm 0.6}$	$\underline{40.5}^{\pm 0.7}$	$62.4^{\pm 0.6}$
MRN	$\underline{13.4}^{\pm 0.4}$	$40.0^{\pm 0.8}$	$61.9^{\pm 0.6}$
FiLM	$12.9^{\pm 0.7}$	$39.5^{\pm 2.1}$	$61.9^{\pm 1.9}$
TIRG	$\mathbf{14.1}^{\pm 0.6}$	$\mathbf{42.5}^{\pm 0.7}$	$\mathbf{63.8}^{\pm 0.8}$

Table 1. Retrieval performance on Fashion200k. The best number is in bold and the second best is underlined.

Method	R@1	R@5	R@10
Image only	$3.3^{\pm 0.1}$	$12.8^{\pm 0.2}$	$20.9^{\pm 0.1}$
Text only	$7.4^{\pm 0.4}$	$21.5^{\pm 0.9}$	$32.7^{\pm 0.8}$
Concatenation	$11.8^{\pm 0.2}$	$30.8^{\pm 0.2}$	$42.1^{\pm 0.3}$
Show and Tell	$11.9^{\pm 0.1}$	$31.0^{\pm 0.5}$	$42.0^{\pm 0.8}$
Att. as Operator	$8.8^{\pm 0.1}$	$27.3^{\pm 0.3}$	$39.1^{\pm 0.3}$
Relationship	$\mathbf{12.3}^{\pm 0.5}$	$\mathbf{31.9}^{\pm 0.7}$	$\underline{42.9}^{\pm 0.9}$
MRN	$11.9^{\pm 0.6}$	$30.5^{\pm 0.3}$	$41.0^{\pm 0.2}$
FiLM	$10.1^{\pm 0.3}$	$27.7^{\pm 0.7}$	$38.3^{\pm 0.7}$
TIRG	$\underline{12.2}^{\pm 0.4}$	$\mathbf{31.9}^{\pm 0.3}$	$\mathbf{43.1}^{\pm 0.3}$

Table 2. Retrieval performance on MIT-States.

4.2.1 Image retrieval

We use this dataset for image retrieval as follows: pairs of images with the same object labels and different state labeled are sampled. They are using as query image and target image respectably. The modification text will be the state of the target image. Hence the system is supposed to retrieve images of the same object as the query image, but with the new state described by text. We use 49 of the nouns for testing, and the rest is for training. This allows the model to learn about state/adjective (modification text) during training and has to deal with unseen objects presented in the test query.

Some qualitative results are shown in Fig. 4 and the R@K performance is shown in Table 2. Note that similar types of objects with different states can look drastically different, making the the role of modification text more important. Hence on this dataset, the [Text Only] baseline outperforms [Image Only]. Nevertheless, combining them gives better results. The difference between composition methods is not too significant here. Still TIRG is comparable to Relationship while outperforming others.

4.2.2 Classification with compositionally novel labels

To be able to compare to prior work on this dataset, we also consider the classification setting proposed in [31, 33]. The goal is to learn models to recognize unseen combination of

Figure 3. Retrieval examples on Fashion200k dataset.

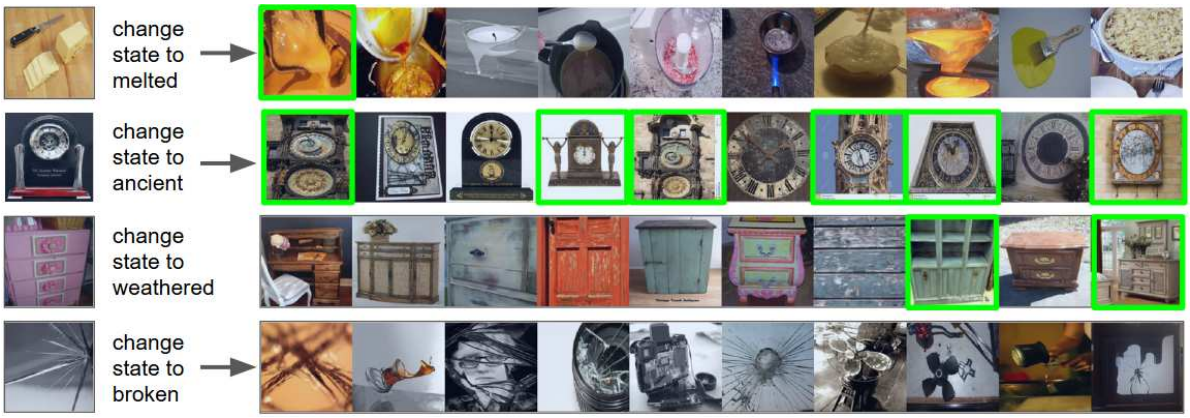

Figure 4. Some retrieval examples on MIT-States.

(state, noun) pairs. For example, training on "red wine" and "old tomato" to recognize "red tomato" where there exist no "red tomato" images in training.

To tackle this in our framework, we define ϕ_x to be the feature vector derived from the image x (using ResNet-17 as before), and ϕ_t to be the feature vector derived from the text t. The text is composed of two words, a noun n and an adjective a. We learn an embedding for each of these words, ϕ_n and ϕ_a, and then use our TIRG method to compute the combination ϕ_{an}. Given this, we perform image classification using nearest neighbor retrieval, so $t(x) = \arg\max_t \kappa(\phi_t, \phi_x)$, where κ is a similarity kernel applied to the learned embeddings. (In contrast to our other experiments, here we embed text and image into the same shared space.)

The results, using the same compositional split as in [31, 33], are shown in Table 3. Even though this problem is not the focus of our study, we see that our method outperforms prior methods on this task. The difference from the previous best method, [33], is that their composition feature is represented as a dot product between adjective transformation matrix and noun feature vector; by contrast, we represent both adjective and noun as feature vectors and combine them using our composition mechanism.

Method	Accuracy
Analogous Attribute [3]	1.4
Red wine [31]	13.1
Attribute as Operator [33]	14.2
VisProd NN [33]	13.9
Label Embedded+ [33]	14.8
TIRG	**15.2**

Table 3. Comparison to the state-of-the-art on the unseen combination classification task on MIT-States. All baseline numbers are from previous works.

4.3. CSS dataset

Since existing benchmarks for image retrieval do not contain complex text modifications, we create a new dataset, as we describe below.

4.3.1 Dataset Description

We created a new dataset using the CLEVR toolkit [20] for generating synthesized images in a 3-by-3 grid scene. We render objects with different *C*olor, *S*hape and *S*ize (CSS) occupy. Each image comes in a simple 2D blobs version and a 3D rendered version. Examples are shown in Fig. 5.

We generate three types of modification texts from templates: adding, removing or changing object attributes. The "add object" modification specifies a new object to be placed in the scene (its color, size, shape, position). If any of the attribute is not specified, its value will be randomly chosen. Examples are "add object", "add red cube", "add big red cube to middle-center". Likewise, the "remove object" modification specifies the object to be removed from the scene. All objects that match the specified attribute values will be removed, e.g. "remove yellow sphere", "remove middle-center object". Finally, the "change object" modification specifies the object to be changed and its new attribute value. The new attribute value has to be either color or size. All objects that match the specified attribute will be changed, e.g. "make yellow sphere small", "make middle-center object red".

In total, we generate 16K queries for training and 16K queries for test. Each query is of a reference image (2D or 3D) and a modification, and the target image. To be specific, we first generate 1K random scenes as the reference. Then we randomly generate modifications and apply them to the reference images, resulting in a set of 16K target images. In this way, one reference image can be transformed to multiple different target images, and one modification can be applied to multiple different reference images. We then repeat the process to generate the test images. We follow the protocol proposed in [20] in which certain object shape and color combinations only appear in training, and not in testing, and vice versa. This provides a stronger test of generalization.

Although the CSS dataset is simple, it allows us to perform controlled experiments, with multi-word text queries, similar to the CLEVR dataset. In particular, we can create queries using a 2d image and text string, to simulate the case where the user is sketching something, and then wants to modify it using language. We can also create queries using slightly more realistic 3d image and text strings.

4.3.2 Results

Table 4 summarizes R@1 retrieval performance on the CSS dataset. We examine two retrieval settings using 3d query images (2nd column) and 2d images (3rd column). As we can see, our TIRG combination outperforms other composition methods for the retrieval task. In addition, we see that retrieving a 3D image from a 2D query is much harder, since the feature spaces are quite different. (In these experiments, we use different feature encoders for the 2D and 3D inputs). Some qualitative results are shown in Fig. 6.

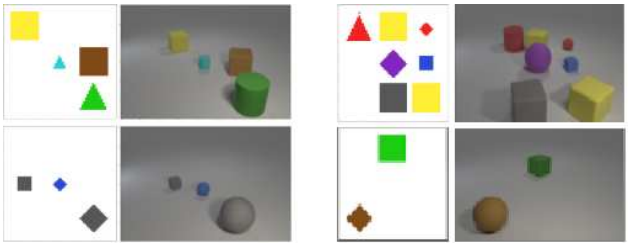

Figure 5. Example images in our CSS dataset. The same scene are rendered in 2D and 3D images.

Method	3D-to-3D	2D-to-3D
Image only	6.3	6.3
Text only	0.1	0.1
Concatenate	$60.6^{\pm 0.8}$	27.3
Show and Tell	$33.0^{\pm 3.2}$	6.0
Parameter hashing	$60.5^{\pm 1.9}$	31.4
Relationship	$62.1^{\pm 1.2}$	30.6
MRN	$60.1^{\pm 2.7}$	26.8
FiLM	$\underline{65.6}^{\pm 0.5}$	$\underline{43.7}$
TIRG	$\mathbf{73.7}^{\pm 1.0}$	**46.6**

Table 4. Retrieval performance (R@1) on the CSS Dataset using 2D and 3D images as the query.

To gain more insight into the nature of the combined features, we trained a transposed convolutional network to reconstruct the images from their features and then apply it to composition feature. Fig. 7 shows the reconstructed images from the composition features of three methods. Images generated from our feature representation look visually better, and are closer to the top retrieved image. We see that all the images are blurry as we use the regression loss to train the network. However, a nicer reconstruction may not mean better retrieval, as the composition feature is learned to capture the discriminative information need to find the target image, and this may be a lossy representation.

4.4. Ablation Studies

Method	Fashion	MIT-States	CSS
Our Full Model	14.1	12.2	73.7
- gated feature only	13.9	07.1	06.5
- residue feature only	12.1	11.9	60.6
- mod. at last fc	14.1	12.2	71.2
- mod. at last conv	12.4	10.3	73.7
DML loss, $K = 2$	9.5	12.2	73.7
DML loss, $K = B$	14.1	10.9	69.8

Table 5. Retrieval performance (R@1) of ablation studies.

In this section, we report the results of various ablation studies, to gain insight into which parts of our approach matter the most. The results are in Table 5.

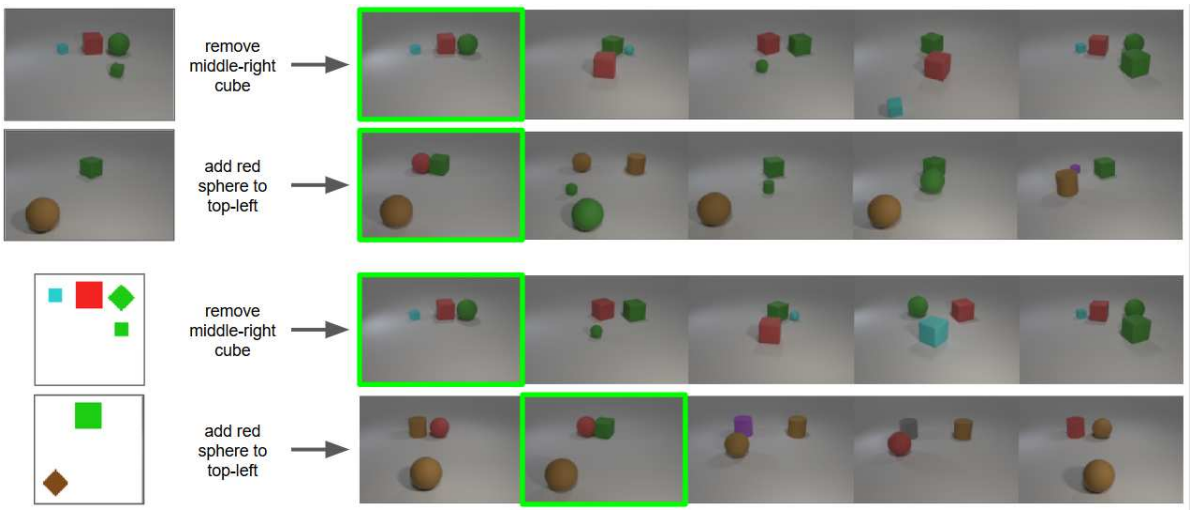

Figure 6. Some retrieval examples on CSS Dataset.

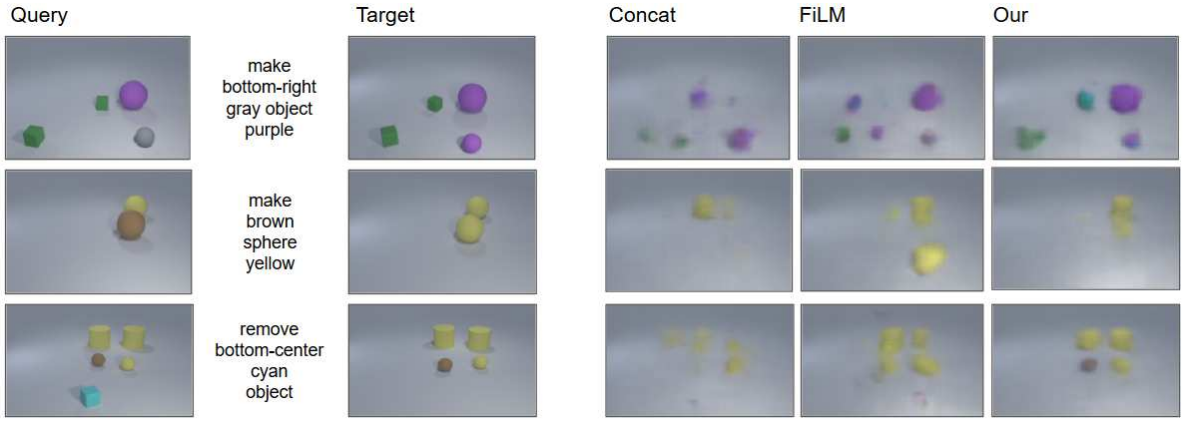

Figure 7. Reconstruction images from the learned composition features.

Efficacy of feature modification: as shown in Fig. 2, our composition module has two types of connections, namely residual connection and gated connection. Row 2 and 3 show that removing the residual features or gating features leads to drops in performance. In these extreme cases, our model can degenerate to the concatenate fusion baseline.

Spatial versus non-spatial modification: Row 5 and 6 compares the effect of applying our feature modification to the last fc layer versus the last convolution layer. When our modification is applied to the last fc layer feature, it yields competitive performance compared to the baseline across all datasets. Applying the modification to the last convolution feature map only improves the performance on CSS. We believe this is because the modifications in the CSS dataset is more spatially localized (see Fig. 6) whereas they are more global on the other two datasets (See Fig. 3 and Fig. 4)

The impact of K in the loss function: The last two rows compares the loss function of two different K values in Section 3.3. We use $K = 2$ (soft triplet loss) in most experiments. As Fashion200k is much bigger, we found that the network underfitted. In this case by using $K = B$ (same as batch size in our experiment), the network fits well and produces better results. On the other two datasets, test time performance is comparable, but training becomes less stable. Note that the difference here regards our metric learning loss and does not reflect the difference between the feature composition methods.

5. Conclusion

In this work, we explored the composition of image and text in the context of image retrieval. We experimentally evaluated several existing methods, and proposed a new one, which gives improved performance on three benchmark datasets. In the future, we would like to try to scale this method up to work on real image retrieval systems "in the wild".

References

[1] K. E. Ak, A. A. Kassim, J. H. Lim, and J. Y. Tham. Learning attribute representations with localization for flexible fashion search. In *CVPR*, 2018. 2

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015. 1, 2, 3

[3] C.-Y. Chen and K. Grauman. Inferring analogous attributes. In *CVPR*, 2014. 6

[4] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu. Language-based image editing with recurrent attentive models. In *CVPR*, 2018. 2

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 2

[6] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2

[7] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016. 2

[8] S. Gidaris and N. Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 4

[9] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2005. 4

[10] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In *ECCV*, 2016. 2

[11] X. Guo, H. Wu, Y. Cheng, S. Rennie, and R. S. Feris. Dialog-based interactive image retrieval. *arXiv preprint arXiv:1805.00145*, 2018. 2, 3

[12] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017. 1, 2, 4, 5

[13] J. Hays and A. A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[15] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 1, 4

[16] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, 2016. 2

[17] P. Isola, J. J. Lim, and E. H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2, 4, 5

[18] L. Jiang, A. G. Hauptmann, and G. Xiang. Leveraging high-level and low-level features for multimedia event detection. In *ACM MM*, 2012. 2

[19] L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, and A. G. Hauptmann. Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *ICMR*, 2015. 2

[20] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2, 7

[21] K. Kato, Y. Li, and A. Gupta. Compositional learning for human object interaction. In *ECCV*, 2018. 2

[22] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *NIPS*, 2016. 2, 3

[23] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *CVPR*, 2012. 2

[24] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2

[25] J. Liang, L. Jiang, L. Cao, L.-J. Li, and A. Hauptmann. Focal visual-text attention for visual question answering. In *CVPR*, 2018. 2

[26] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *CVPR*, 2015. 2

[27] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun. ivqa: Inverse visual question answering. In *CVPR*, 2018. 2

[28] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 2

[29] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 2

[30] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 3

[31] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 2, 3, 5, 6

[32] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017. 4

[33] T. Nagarajan and K. Grauman. Attributes as operators. 2018. 2, 3, 5, 6

[34] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2017. 2

[35] H. Noh, P. Hongsuck Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *CVPR*, 2016. 1, 2, 3

[36] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, 2015. 2

[37] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. 2018. 2, 3

[38] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016. 2

[39] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 2

[40] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998. 2

[41] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011. 2

[42] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):119, 2016. 2

[43] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017. 1, 2, 3

[44] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[45] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 4

[46] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016. 4

[47] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015. 3

[48] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *CVPR*, 2018. 2

[49] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 3

[50] N. N. Vo and J. Hays. Localizing and orienting street views using overhead imagery. In *ECCV*, 2016. 4

[51] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 2

[52] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 2

[53] L. Yu, Z. Lin, X. Shen, J. Yang, X. Lu, M. Bansal, and T. L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 2

[54] L. Yu, H. Tan, M. Bansal, and T. L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017. 2

[55] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015. 2

[56] B. Zhao, J. Feng, X. Wu, and S. Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *CVPR*, 2017. 2, 3