Leveraging Style and Content features for Text Conditioned Image Retrieval

Pranit Chawla
IIT Kharagpur

Surgan Jandial
IIT Hyderabad

Pinkesh Badjatiya
Media and Data Science Research Lab, Adobe

Ayush Chopra
MIT

Mausoom Sarkar
Media and Data Science Research Lab, Adobe

Balaji Krishnamurthy
Media and Data Science Research Lab, Adobe

Abstract

Image Search is a fundamental task playing a significant role in the success of wide variety of frameworks and applications. However, with the increasing sizes of product catalogues and the number of attributes per product, it has become difficult for users to express their needs effectively. Therefore, we focus on the problem of Image Retrieval with Text Feedback, which involves retrieving modified images according to the natural language feedback provided by users. In this work, we hypothesise that since an image can be delineated by its content and style features, modifications to the image can also take place in the two sub spaces respectively. Hence, we decompose an input image into its corresponding style and content features, apply modification of the text feedback individually in both the style and content spaces and finally fuse them for retrieval. Our experiments show that our approach outperforms a recent state of the art method in this task, TIRG, that seeks to use a single vector in contrast to leveraging the modification via text over style and content spaces separately.

1. Introduction

The online shopping industry is a billion dollar industry that is envisioned to grow multi-folds over the next years. Especially with the pandemic, the number of people turning online for their day-to-day needs has been more than ever. With this increasing interest, there are significant efforts in providing smart, interactive & intuitive experiences for online commerce including Image Search [3, 9], Virtual Try-On [6, 5] and Fashion Compatibility. In this work, we focus our attention in the direction of interactive image search. A major drawback of current frameworks in image search [3, 10] which use a single image or text for searching is that they are not able to capture fine-grained user requirements through a single image or a natural language query. Therefore, recent efforts focus on refining the result through user feedback in form of spatial layouts [8], scene-graphs [7] or relative attributes [12]. This feedback can be further improved by using natural language expressions as user feedback, allowing them immense flexibility in interactive image search [4]. In this work, we focus on incorporating diverse natural language descriptions as feedback for interactive image search with the specific task denoted as text conditioned image retrieval (TCIR). Formally, given a reference image as input and a support text description as feedback, TCIR is concerned with retrieving the best matching images that satisfy similarity constraints imposed by both the components of multi-modal input. Figure 1 further shows an illustration of the task.

TIRG [11] being the one to introduce the task, forms the basis of our approach. TIRG uses a CNN to obtain image embedding, an RNN to obtain text embedding and then they fuse the two using a gated connection. Their intuition

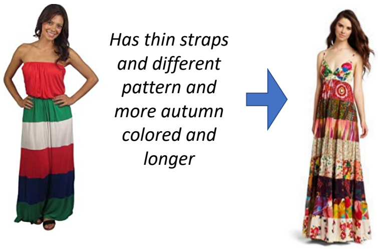

Figure 1. Given a *reference image* and a *support text*, we focus on the task of retrieving images that resemble the *reference image* while also satisfying constraints imposed by the *support text*

.

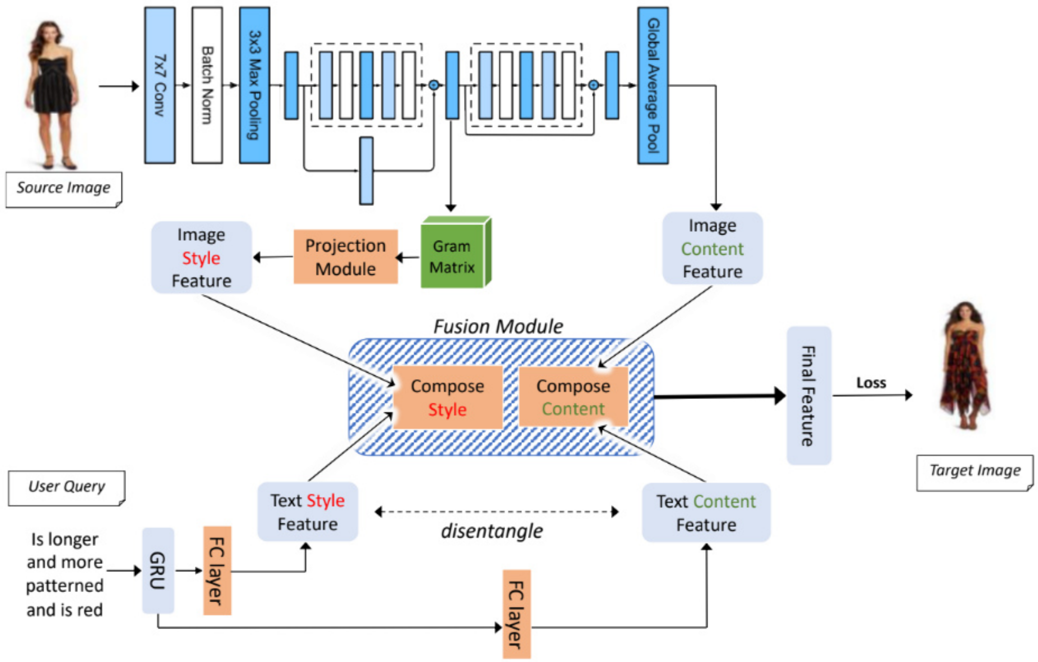

Figure 2. Outline of our proposed framework

for the gated connection is that they try to keep the space of the query image same and only modify certain aspects using the text in the same image space as the query. Contrary to TIRG, which uses a single feature obtained as a result of Global Average Pool from the last layer to obtain image embedding, we propose to represent the image using its style and content component and then transform each of them individually for further retrieval. Since each image can be well characterized by their content and style, with this, we hypothesize that given the modifying text query, both style and content will be modified accordingly in their own spaces, thereby motivating us to study each of the modification separately to further refine the retrieval. Our detailed set of experiments show how separating features help in making a more refined decision for TCIR and improves retrieval performance.

2. Methodology

Motivated by [2], we also leverage the disentangled style and content features of an image for the purpose of TCIR. For the content feature vector, we take the global average pooled features from last layer of a CNN like ResNet-50 while for style features we use the inter-channel correlations obtained using Gram Matrix (as mentioned in [2]). Our approach in detail is hereby explained in sections 2.1 till 2.3 while section 2.4 explains the loss functions we use to train our model. An overview of the entire process is shown in Figure 2.

2.1. Generating Content and Style Based Features

Given a query image I^q, a modifying text query T and a target image I^t as shown in Figure 1, we generate their separate style and content based representations. To obtain these, we take the Resnet-50 model and slice it into two parts, Φ_1 and Φ_2. Then, the query image is passed through I^q through Φ_1 to get $\mathcal{R}_s \in \mathbb{R}^{C \times H \times W}$. R_s is then passed through the GM module which returns a 2D map $G_s \in \mathbb{R}^{C \times C}$, where C is the number of channels. We then use $AvgPool1d$ to reduce G_s to U_s. Finally, $\mathcal{I}_s \in \mathbb{R}^{1 \times 512}$ and $\mathcal{I}_c \in \mathbb{R}^{1 \times 512}$ are obtained as shown below:

$$\begin{aligned}
R_s &= \Phi_1(I^q) \\
G_s &= GM(R_s) \\
U_s &= AvgPool1d(G_s) \\
I_s^q &= F_s^{image}(ReLU(BatchNorm(U_s))) \\
GM(Z)_{ij} &= \sum_k Z_{ik} Z_{jk}
\end{aligned} \tag{1}$$

where GM refers to the GramMatrix operation and $BatchNorm$ represents Batch Normalization operation. F_c^{image} represents a fully connected layer for the style operation. To explain I_s^q as style representation, we can observe that since 2-D matrix G_s in the above equation represents inter-channel correlations, on taking average along one dimension, we obtain the scores for each channel conditioned on its interactions and co-occurrence with the other channels. Hence , I_s^q, encoding the relationship

Method	Dress		Toptee		Shirt		Average	
	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
Image Only	2.30	7.51	3.32	9.42	3.24	8.15	2.95	8.36
Text Only	6.39	20.92	4.60	15.80	5.94	21.10	5.64	19.27
Concat	7.30	22.31	5.73	17.87	7.29	21.31	6.77	20.50
TIRG [11]	11.11	29.15	12.95	30.80	10.20	25.46	11.42	28.47
Ours	**19.33**	**43.52**	**19.73**	**44.56**	**14.47**	**35.47**	**17.84**	**41.18**

Table 1. Quantitative results on Validation Set of FashionIQ dataset. Best numbers are highlighted in **bold**.

between different filter responses, can intuitively said to be a style equivalent representation.

$$\begin{aligned} R_c &= \Phi_2(R_s) \\ BN_c &= BatchNorm(R_c) \\ I_c^q &= F_c^{image}(R_c) \end{aligned} \tag{2}$$

where $BatchNorm$ refers to the Batch Normalization operation and F_c^{image} refer to a fully connected layer for content. The simple global average pooled representation I_c^q acts as the content embedding as it encodes spatial relationships in the query image.

We follow a similar approach to obtain the respective representations, I_s^t and I_c^t for the target image as well.

Now, for getting representations for support text T (as shown in Fig 1), having D number of words, we first get word level feature vectors $\Omega_{word} \in \mathbb{R}^{1\times 768}$ using the pretrained BERT model [1]. These are then passed through a GRU to get the sentence level feature vector $\Omega_{sent} \in \mathbb{R}^{1\times 1024}$ as

$$\Omega_{sent} = \text{GRU}([\Omega^1_{word},\ ^2_{word},\ \cdots\ ,\ \Omega^D_{word}]) \tag{3}$$

Since there is no clear description to obtain style and content vectors for text and also to avoid further complexity, we finally use two separate linear projections to obtain the style vector $\mathcal{T}_s \in \mathbb{R}^{1\times 512}$ and the content vector $\mathcal{T}_c \in \mathbb{R}^{1\times 512}$ respectively as

$$T_s = F_s^{text}(\Omega_{sent}), T_c = F_c^{text}(\Omega_{sent}) \tag{4}$$

where F_s^{text} and F_c^{text} are fully connected layers for the style and content feature respectively.

We try to enforce the characteristics corresponding to the style and content features of the image through these linear projections.

2.2. Composing Image and Text Features

Once we get the style and content representations for the image, R_c and R_s and the text, T_c and T_s respectively, we compose them individually to obtain joint style and content feature vectors. We use residual offsetting followed by $L2$ normalisation to obtain the composed feature vectors for style and content as

$$V_c = \frac{I_c^q + T_c}{\|I_c^q + T_c\|_2}, V_s = \frac{I_s^q + T_s}{\|I_s^q + T_s\|_2} \tag{5}$$

where $\|.\|_2$ denotes the L_2 norm.

2.3. Fusing Style and Content Features

Finally, we obtain a fused feature vector X_{query} which fuses the composed style and content vectors for the query and X_{target} which fuses the image style and content vectors for the target as follows:

$$X_{query} = \frac{V_c + V_s}{\|V_c + V_s\|_2}, X_{target} = \frac{I_s^t + I_c^t}{\|I_s^t + I_c^t\|_2} \tag{6}$$

2.4. Loss Functions

We train our model using two loss functions the triplet loss and the discriminator loss. The triplet loss is used to push the fused query vector closer to its corresponding target (X^{pos}_{target}) and further from other target vectors (X^{neg}_{target}). It is defined as:

$$\mathcal{L}_T = \log(1 + e^{\|X_{query} - X^{pos}_{target}\|_2\ -\ \|X_{query} - X^{neg}_{target}\|_2}) \tag{7}$$

where $\|.\|_2$ operator denotes the L_2 norm.

We also use a discriminator loss to better align X_{query} and X_{target}

$$\mathcal{L}_D = -\mathbb{E}\big[log(\mathcal{D}(X_{query})\big] - \mathbb{E}\big[log(1 - \mathcal{D}(X_{target}))\big] \tag{8}$$

We train our network using a linear combination of L_T and L_D where

$$\mathcal{L} = \lambda_1 \mathcal{L}_T + \lambda_2 \mathcal{L}_D \tag{9}$$

where λ_1 and λ_2 are scalars.

3. Experiments

In this section, we explain the dataset, the evaluation metric and our implementation details.

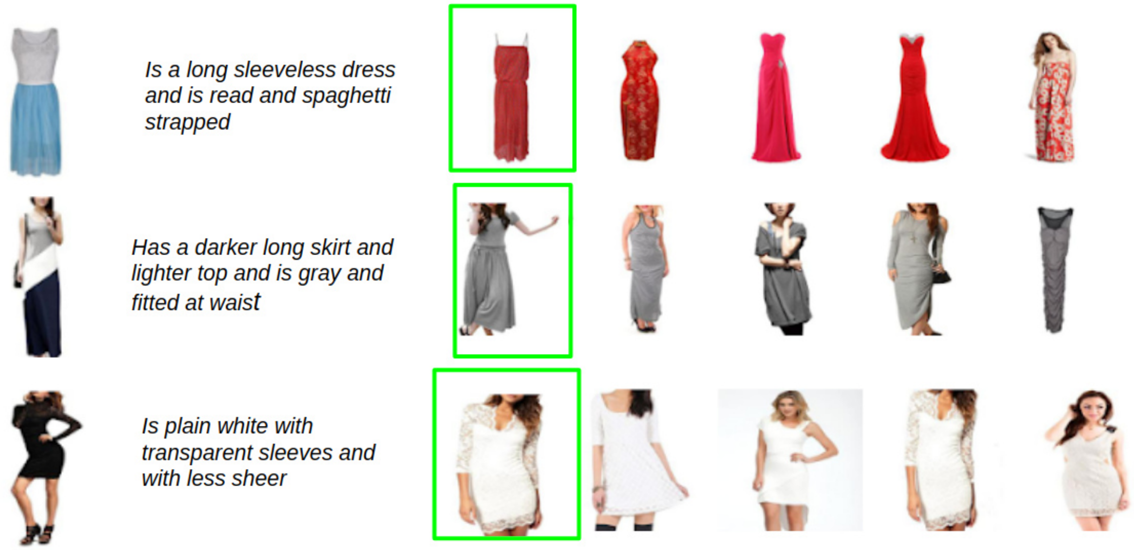
Is a long sleeveless dress
and is read and spaghetti
strapped
Has a darker long skirt and
lighter top and is gray and
fitted at waist
Is plain white with
transparent sleeves and
with less sheer

Figure 3. Qualitative results on the Fashion IQ dataset. First column shows query image, second column shows modifying text and third column shows target image.

3.1. Dataset

For evaluating our proposed approach, we use a standard benchmark dataset, FashionIQ and standard metric R@K for evaluation. FashionIQ is a natural language based interactive fashion product retrieval dataset. This dataset is characterized by natural language queries with an average query length of 10.69. It contains 77,684 images crawled from Amazon.com, covering three categories: Dresses, Tops Tees and Shirts. Among the 46,609 training images, there are 18,000 image pairs, with each pair accompanied with around two natural language sentences that describe one or multiple visual properties to modify in the reference image.

3.2. Implementation Details

We use Pytorch in our experiments. We use Resnet-50 (output feature size = 512) pretrained on ImageNet as our backbone for the image encoding and a GRU (output feature size = 512) as our encoder for the natural language caption. We initialise the word vectors using the BERT model's word embeddings. During training, we use Adam optimizer for our image and text encoders (initial learning rate = 0.001) along with SGD optimizer for the discriminator (initial learning rate = 0.0002). The value of $\lambda 1$ is taken as 1 and $\lambda 2$ is taken to be 0.7.

4. Results

4.1. Quantitative Results

Here we present the quantitative results obtained using our method. We use the standard metric Recall@K or R@K to report results. As visible in the Table 1, our technique beats all the baselines comfortably and beats the best baseline TIRG by 6.42% on R@10 and 12.71% on R@50. The results are reported on the validation set of Fashion IQ as the ground truth target images for the test set were not available.

4.2. Qualitative Results

We show the qualitative results in Figure 3. The first column shows the query image, the second column the modifying text and the third column the retrieved results of our model. It can be observed that our model is able to identify multiple transforming features in the modifying text such as "sleeveless dress" and "spaghetti strapped" in the first row while preserving other qualities of the original query image. Similar transformations can be seen in the second and third row.

5. Conclusion and Future Work

In this work, we focus on the task of text conditioned image retrieval and introduce our method which leverages the decomposed style and content features in an image. We conduct extensive experiments on the Fashion IQ and consistently achieve better performance than TIRG [11], a recent state of the art in this task.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 3

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015. 2

[3] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016. 1

[4] Xiaoxiao Guo, Hui Wu, Yupeng Gao, Steven Rennie, and Rogerio Feris. Fashion iq: A new dataset towards retrieving images by natural language feedback. *arXiv preprint arXiv:1905.12794*, 2019. 1

[5] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis. Viton: An image-based virtual try-on network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7543–7552, 2018. 1

[6] Surgan Jandial, Ayush Chopra, Kumar Ayush, Mayur Hemani, Balaji Krishnamurthy, and Abhijeet Halwai. Sievenet: A unified framework for robust image-based virtual try-on. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2182–2190, 2020. 1

[7] J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015. 1

[8] L. Mai, H. Jin, Z. Lin, C. Fang, J. Brandt, and F. Liu. Spatial-semantic image search by visual feature synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1121–1130, 2017. 1

[9] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017. 1

[10] Nikolaos Sarafianos, Xiang Xu, and Ioannis A. Kakadiaris. Adversarial representation learning for text-to-image matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[11] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6439–6448, 2019. 1, 3, 4

[12] Aron Yu and Kristen Grauman. Thinking outside the pool: Active training image creation for relative attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 708–718, 2019. 1

*