Visualizing Paired Image Similarity in Transformer Networks

Samuel Black[1], Abby Stylianou[2], Robert Pless[3], Richard Souvenir[1]
[1]Temple University, [2]Saint Louis University, [3]George Washington University
{sam.black,souvenir}@temple.edu, abby.stylianou@slu.edu, pless@gwu.edu

Abstract

Transformer architectures have shown promise for a wide range of computer vision tasks, including image embedding. As was the case with convolutional neural networks and other models, explainability of the predictions is a key concern, but visualization approaches tend to be architecture-specific. In this paper, we introduce a new method for producing interpretable visualizations that, given a pair of images encoded with a Transformer, show which regions contributed to their similarity. Additionally, for the task of image retrieval, we compare the performance of Transformer and ResNet models of similar capacity and show that while they have similar performance in aggregate, the retrieved results and the visual explanations for those results are quite different. Code is available at `https://github.com/vidarlab/xformer-paired-viz`.

1. Introduction

The lack of explainability or interpretability is often cited as one of the main drawbacks of deep neural networks used in computer vision and machine learning [2, 25, 27]. The primary approach to addressing this problem in vision domains is through visualization approaches that highlight the portion(s) of the input that most contributed to the output prediction. For images, these heatmap visualizations tend to depict the relative importance of particular pixel regions.

Generating these visualizations often depends on both the (1) network architecture and (2) prediction task. Convolutional neural networks (CNNs) have been the dominant architecture in computer vision and most visualization approaches were specific to these models. Typically, for CNNs, the feature maps at each layer have local support; the representation at a particular location is computed from a small neighborhood of nearby pixel locations. Relevancy maps can be generated and overlaid with the input to produce an intuitive visualization. While most of the work deals with classification networks [31, 30, 37, 45, 29], there has been some recent work on embedding networks [35, 10] which typically use representation learning for downstream tasks, such as image retrieval.

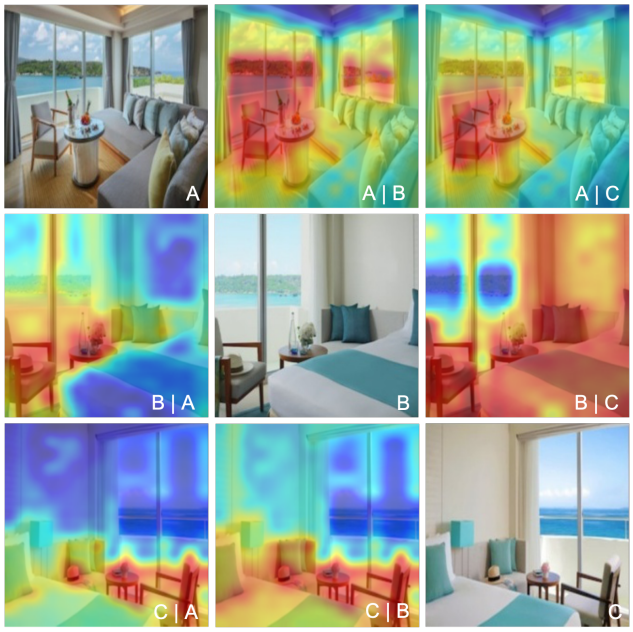

Figure 1. Our approach visualizes paired image similarity in Transformer networks trained for image embedding. For the images along the diagonal, the off-diagonal visualizations show the pairwise similarity maps generated by our approach, where red indicates a high contribution. For example, for image **B**, **B**|**A**, shows that the chair region contributes the most similarity with image **A**, while **B**|**C** also includes the bed and wall contributing to the similarity with image **C**.

Recently, there has been an increase in the use of Transformer networks in computer vision. This architecture, imported from the field of natural language processing, is quite different from a CNN architecture and requires new approaches to explainability due to the global attention operations at each layer and the strong degree of token mixing throughout the model [6]. The early visualization methods for Transformers followed a similar trajectory as with CNNs. First, there were output-agnostic methods that visualized the regions attended to by the model [1], fol-

lowed by methods focused on networks trained for classification [8, 7].

In this paper, we develop a visualization approach for Transformer networks used to learn image embeddings. In this context, interpretability can be considered in a pairwise fashion; a visualization can help explain why pairs of images result in high (or low) similarity scores, typically computed as the dot product of normalized feature vectors. Specifically, this work:

- extends paired image visualizations, originally designed for convolutional neural networks, to image Transformer networks;
- compares ResNet and ViT architectures on image retrieval on standard benchmark datasets; and
- qualitatively demonstrates the utility of the approach for understanding Transformer embedding networks.

2. Related Work

Our method is a paired image similarity visualization for Transformer networks used for image embeddings. In this section, we review related visualization methods for CNNs and Transformer network architectures.

2.1. CNN Visualization

Modern CNN interpretability methods can be categorized into two classes: output-agnostic and output-specific. Output-agnostic methods produce visualizations of the activation maps independent of the output feature or predictive task. Some approaches generate a binary relevancy map by thresholding the values of a feature map from a given layer in the network [44, 4]. Other work [43] incorporates deconvolutional neural networks to transform activation maps into the original pixel space.

Output-specific methods highlight the regions that are most relevant for generating a given target. In the case of classification, these visualizations depict the regions of the input image that contributed to the prediction of a particular class. One method computes the derivative of the target class with respect to the input and produces a saliency map from the pixels with the largest gradient values [31]. Another method, Layer-wise Relevance Propagation (LRP) [3], backpropagates the value of a chosen output neuron to the input, determining the contribution of image pixels using a set of well-defined rules tailored to network's architecture and activation function. Contrastive [16] and Softmax Gradient LRP [19] extend LRP and contrast the chosen target with all possible outputs.

Other output-specific methods include [30, 37] which multiply the gradients with respect to the target class with the input pixel values, rather than visualize the magnitude of the gradients alone. These approaches tend to generate similar visualizations for different target classes, thus negating the aspect of output specificity. Class-Activation Maps (CAM) [45] were shown to produce discriminative visualizations. CAMs are generated by taking a weighted sum of the feature maps produced by the last convolutional layer in the network, using the weights of the global pooled feature with respect to the target class as a multiplier. GradCAM [29] generalizes this framework for different network layers and architectures, weighting the feature maps instead by the gradients with respect to the target class.

There is less work in output-specific visualizations for image embedding networks. For networks trained with triplet-style loss functions [38, 18], one such method [10] employs an approach, following the style of GradCAM, that averages the gradients from sampled training triplets. To produce the visualization of a test image, the gradients of the most similar training image are used for the weighted sum of the feature maps. Stylianou et al. [35] introduced a method for generating heatmaps from a pair of images which highlight the regions that contribute the most to their pairwise similarity. This method decomposes the similarity calculation across each spatial location in the final feature maps of both images, producing a separate, but complementary, visualization for each input image.

2.2. Transformer Interpretability

The distinguishing feature of Transformer networks is the use of attention in each encoding block. One contested notion has been whether analyzing the attention weights is suitable for explaining predictions made by the model. The weights of attention modules used in RNNs have been shown to have low correlation with other methods of assigning feature importance [20] and Transformer training can be manipulated such that low attention weights are assigned to salient tokens at inference [26]. On the other hand, it has been shown that attention weights can provide useful insights into the Transformers used for language processing, such as correlations to grammatical relationships in BERT [11]. Attention weights have also been shown to provide model explainability for question answering, while [40] even introduced a method of probing them in BERT to guide fine-tuning on downstream tasks.

Transformers have only recently migrated to the computer vision community, so there has been less work on interpretability methods for these models. Most of the work so far focuses on using the attention weights to produce intuitive visualizations. The nature of global attention operations have been shown to facilitate high levels of token mixing, particularly at later layers of the network [6]. Compared to CNN feature maps, whose components have strong local receptive fields, this makes it more difficult to assign relevance with respect to the input pixel locations. One approximation, Rollout [1], involves iteratively multiplying the attention weights across each layer to compute the attention flow. While Rollout itself is output-agnostic, it has been

used as the foundation for output-specific visualizations for Transformer networks used for classification [8, 7].

As Transformer networks gain popularity in computer vision, they are being applied to more tasks. In this paper, we consider Transformer networks used for image embeddings, as in [15]. These networks are typically trained using one of the various pairwise, triplet, or proxy loss functions used in deep metric learning (e.g., [28, 33, 41, 21]). The output of these networks is a dense vector and, for two input images, pairwise image similarity is typically computed as the cosine similarity of the feature vectors. Our work is applicable to Transformers whose input is non-overlapping image patches with a varying number of encoding blocks followed by a pooling layer to generate the output feature (Figure 2). This encompasses a wide range of Transformer models including Swin [23], HVT [24], and PSViT [9].

3. Method

Our method, depicted in Figure 3, adapts a paired image visualization approach designed for CNNs [35] in combination with a Transformer attention flow approximation [1]. For a pair of images, the output is a pair of heatmaps that highlight the spatial regions which contributed to the similarity between the images.

3.1. Paired Image Visualization

We focus on Transformer architectures whose final operation is a global pooling operation on the output tokens of the last encoding block[1]. Let $\boldsymbol{\alpha} \in \mathbb{R}^{K \times K \times C}$ represent the output tensor of $K \times K$ C-dimensional tokens. The output feature, $\boldsymbol{\beta} \in \mathbb{R}^{C}$ is generated by a pooling operation. Here, we consider mean pooling:

$$\boldsymbol{\beta} = \frac{1}{K^2} \sum_{x,y} \boldsymbol{\alpha}_{x,y} \tag{1}$$

For two image features $\boldsymbol{\beta}^i$ and $\boldsymbol{\beta}^j$, their cosine similarity is given by:

$$s(\boldsymbol{\beta}^i, \boldsymbol{\beta}^j) = \frac{\boldsymbol{\beta}^i \cdot \boldsymbol{\beta}^j}{\left\|\boldsymbol{\beta}^i\right\| \left\|\boldsymbol{\beta}^j\right\|} \tag{2}$$

We can decompose the similarity across the output tokens of image i with respect to image j by substituting Equation 1 into Equation 2.

$$\begin{aligned} s(\boldsymbol{\beta}^i, \boldsymbol{\beta}^j) &= \frac{\frac{1}{K^2} \sum_{x,y} \boldsymbol{\alpha}^i_{x,y} \cdot \boldsymbol{\beta}^j}{\left\|\boldsymbol{\beta}^i\right\| \left\|\boldsymbol{\beta}^j\right\|} \\ &= \frac{\boldsymbol{\alpha}^i_{1,1} \cdot \boldsymbol{\beta}^j + ... + \boldsymbol{\alpha}^i_{K,K} \cdot \boldsymbol{\beta}^j}{K^2 \left\|\boldsymbol{\beta}^i\right\| \left\|\boldsymbol{\beta}^j\right\|} \end{aligned} \tag{3}$$

[1]The approach is also applicable to networks with an additional fully connected layer, often added for dimensionality reduction. For the sake of clarity, we present the derivation without accounting for these weights.

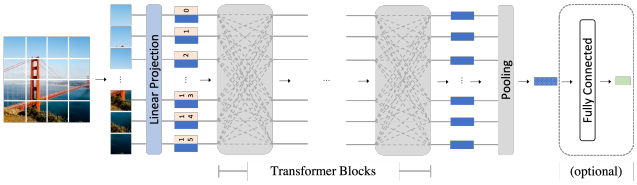

Figure 2. Our visualization approach is applicable to Transformers whose input is non-overlapping image patches with varying numbers of encoding blocks followed by a pooling layer (and perhaps a fully-connected layer) to generate the output feature, such as Swin [23], HVT [24], and PSViT [9].

For the output token indexed by (x, y), let $\boldsymbol{\gamma}^{i|j}_{x,y}$ represent the contribution of that token to the pairwise similarity.

$$\boldsymbol{\gamma}^{i|j}_{x,y} = \frac{\boldsymbol{\alpha}^i_{x,y} \cdot \boldsymbol{\beta}^j}{K^2 \left\|\boldsymbol{\beta}^i\right\| \left\|\boldsymbol{\beta}^j\right\|} \tag{4}$$

The corresponding value for the paired image, $\boldsymbol{\gamma}^{j|i}_{x,y}$, can be computed in a symmetric fashion. In the case of CNN architectures, such a representation could be directly used for the relevancy map at the spatial location (x, y) because the convolution and pooling kernels have local receptive fields to that point. However, due to the successive mixing operations at each layer of the Transformer network, the output modules do not maintain the same level of spatial consistency as with CNNs. In the next section, we describe how we apply methods for attention flow approximation to account for this issue and generate an intuitive visualization.

3.2. Attention Flow Approximation

The self-attention modules in each Transformer block (see Figure 2) facilitate global token mixing. As such, the receptive field of late-stage tokens is not limited to neighboring regions in the input. Reversing the attention process and recovering the exact contribution of each input pixel location is not possible due to successive non-invertible functions in the forward computation graph. Recent work has focused on approximations to the attention flow. Our visualization approach is agnostic to the particular approximation approach. For the experiments in this paper, we use Rollout, a straightforward and efficient attention flow approximation method, which has been previously used for single-image Transformer visualizations [14].

Rollout relies on the assumption that attention flows can be combined linearly between Transformer blocks. Let $\boldsymbol{R} \in \mathbb{R}^{K^2 \times K^2}$ represent the pairwise attention flow from each of the K^2 input tokens to each of the K^2 output tokens. That is, each row in $\boldsymbol{R}$ corresponds to the normalized distribution of attention flow from each input image patch. Rollout estimates $\boldsymbol{R}$ iteratively. Let $\overline{\boldsymbol{A}}_l$ represent the attention weights, averaged over multiple heads, at layer l. Let $\boldsymbol{R}_l$ be the cumulative attention flow up to block l and $\boldsymbol{R}_0$ be

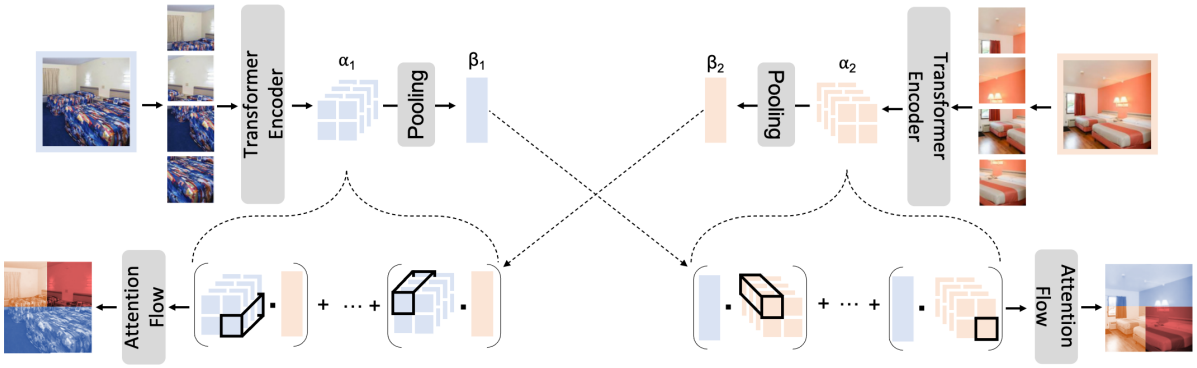

Figure 3. The top row shows the typical forward pass for an Transformer trained for image embedding. We highlight the tokens generated in the last block, α, and the post-pooled feature vector, β. The similarity between two feature vectors, typically computed as $\frac{\beta_1}{||\beta_1||} \cdot \frac{\beta_2}{||\beta_2||}$, can be spatially decomposed using the approach described in Section 3, resulting in a pair of heatmaps that highlight the input regions in each image that most contributed to the similarity with the other image.

the identity matrix, $\boldsymbol{I}$. For each block, $l > 0$:

$$\boldsymbol{R}_l = \boldsymbol{R}_{l-1} \cdot \left\| \overline{\boldsymbol{A}}_l + \boldsymbol{I} \right\|_1 \tag{5}$$

The resulting attention flow approximation, $\boldsymbol{R} = \boldsymbol{R}_L$, for L Transformer blocks, can be used to redistribute the similarity contribution, γ spatially.

3.3. Method Summary

Our approach combines paired image visualizations with the attention flow approximation, Rollout. For a given pair of input images, i and j, we compute the pre-pooled output token representations, α^i and α^j, and post-pooled feature vectors, β^i and β^j. Following Equation 4, compute $\boldsymbol{\gamma}^{i|j}_{x,y}$ and $\boldsymbol{\gamma}^{j|i}_{x,y}$ for each output token (x, y) in both images. Using the Rollout matrix, R, compute $\boldsymbol{\gamma}'$ to redistribute the similarity values to input image space:

$$\boldsymbol{\gamma}' = \boldsymbol{R} \cdot \boldsymbol{\gamma} \tag{6}$$

$\boldsymbol{\gamma}'$ is then reshaped and upsampled to the original image resolution to generate the heatmap visualization.

4. Evaluation

To evaluate our approach to paired image visualization for Transformers, we employ the following image retrieval datasets: Hotels-50k [36], Clean Google LandmarksV2 (GLM) [39, 42], and Stanford Online Products (SOP) [34]. Table 1 summarizes the dataset properties. For Hotels-50k and GLM, our models are trained using ArcFace loss [13], while for SOP we use Proxy Anchor loss [21].

Our image Transformer is a Base ViT model with 16×16 patch size and 12 encoder blocks. For 256×256 input images, there are 16×16 output tokens, which are aggregated with mean pooling. A fully-connected projection layer reduces the dimensionality of the resulting features to 512-D.

Dataset	# Train	# Test	Loss
Hotels-50k	1,085,862	16,121	ArcFace
GLM	1,580,470	761,757	ArcFace
SOP	59,551	60,502	Proxy Anchor

Table 1. Datasets and settings used in the experiments.

The model consists of 86M learnable parameters. For optimization, we use stochastic gradient descent with a 1-cycle learning rate schedule [32].

4.1. Visualizing Pairwise Similarity

Figure 4 shows paired image similarity maps for examples from the Hotels-50k, GLM, and SOP datasets. Overall, the similarity maps tend to highlight large contiguous activated regions, often tightly segmenting entire objects in the scene when present in both images. In the first Hotels-50k example, the most activated regions in each image are the brown bed cover and grey-patterned runner, indicating that these were the most important features in the similarity calculation. Likewise, the most significant regions in the second example are the tiled walls and floor of the showers. The similarity maps form a clear boundary surrounding these areas with the adjacent objects, such as the towel on the glass door in the first image.

When only parts of an object are visible in both images, our method is also able to distinguish which components are shared. In the first GLM example, only the brick facade contributes the most similarity, since it is the part of the building that is seen in both images. Likewise, in the first SOP example, the backs of the chairs in the first image contribute the most to the similarity, while the seats and legs do not. This is due to the close-up shot of the back of a matching chair in the complementing image.

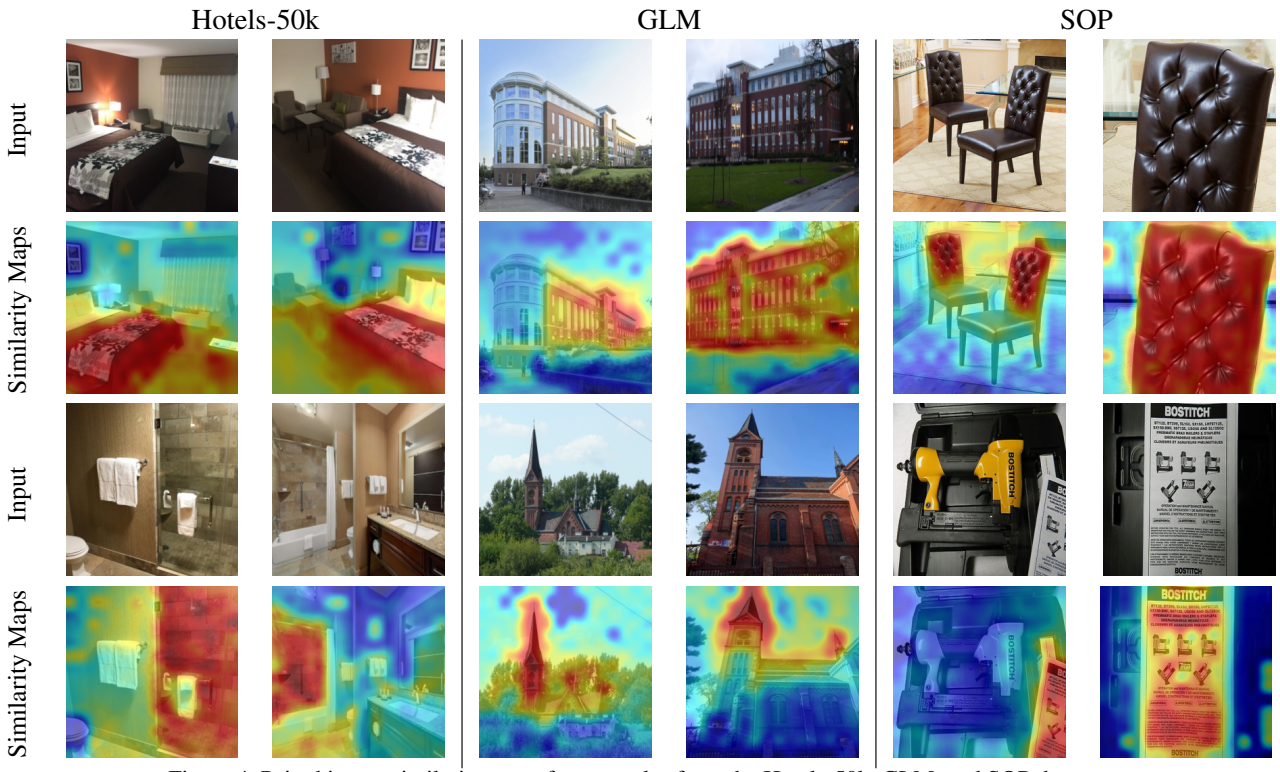

Figure 4. Paired image similarity maps for examples from the Hotels-50k, GLM, and SOP datasets.

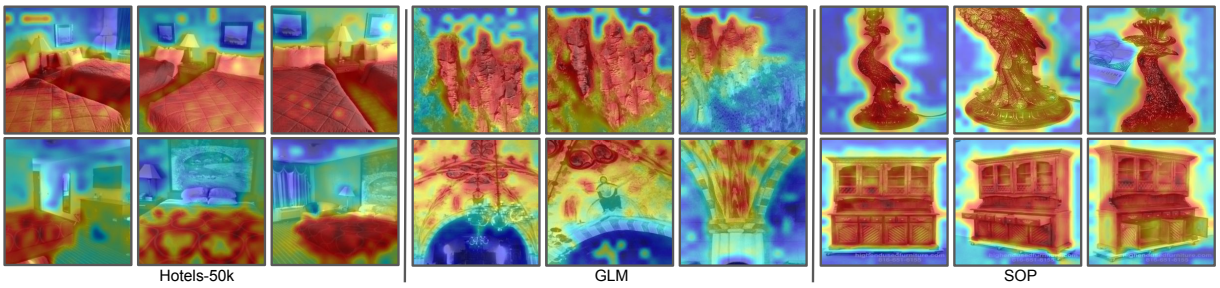

Figure 5. Class similarity maps from selected images. All images in a given row belong to the same class.

4.2. Class Similarity Maps

Like [35], we can adapt our visualization approach to generate class similarity maps. For a given query image, we can compute the pairwise similarity with respect to each gallery image from the same class, and then sum the resulting heatmaps. Figure 5 shows example class similarity maps computed from selected images on each dataset. Class similarity maps such as these help determine which image regions are strongly associated with each class. For the Hotels-50k images, we notice that one of the most salient objects are unique bed covers. They frequently contribute the most similarity, like in both examples from this dataset. Most of the images in SOP are of one central object, taken at different angles. It is common for the entire object to contribute to the similarity in all of the images that share the same label, as seen in their similarity maps. In the visualizations from the GLM dataset, whose images come from a greater variety of scenes, we observe that often only the most unique features have the greatest similarity values. These include the distinctive rock formations in the first example and the ceiling artwork in the second.

4.3. Effect of Training

Figure 6 shows the paired image visualizations for a query image and its three most similar images at different iterations during the training process. We observe that the regions that contribute to the similarity often change as the network converges, with the most significant shifts occurring within the first 25,000 iterations. In the given example, the building does not contribute strongly in the initial similarity maps of the three retrieved images. As training

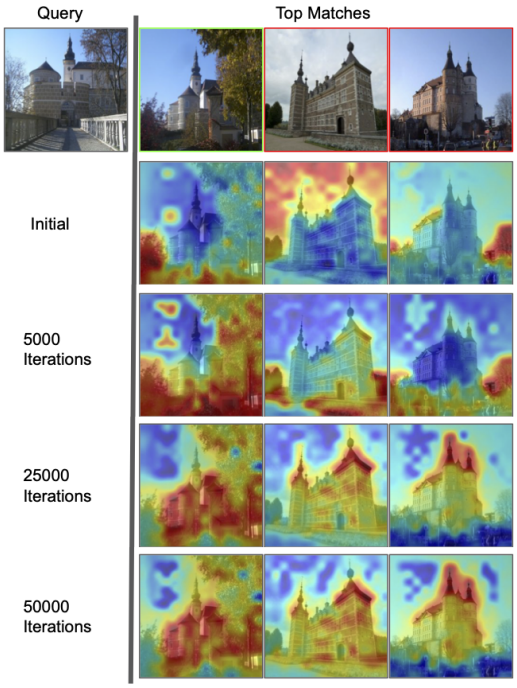

Figure 6. A query image and its top 3 matches, along with their accompanying similarity maps at different stages of training. A green border indicates a correct retrieval.

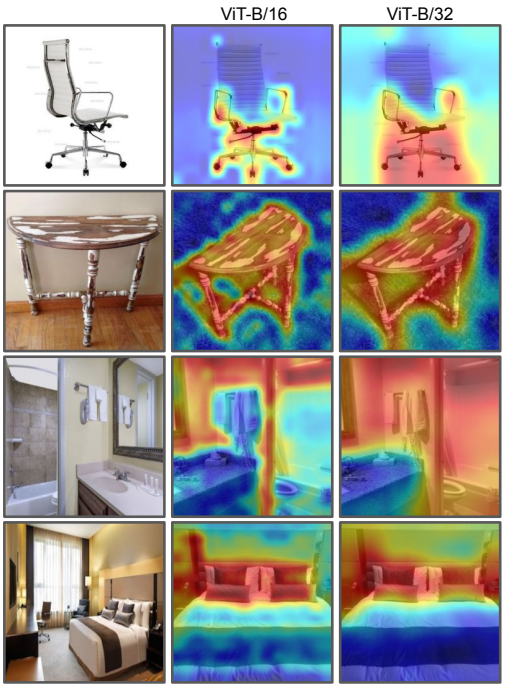

Figure 7. Similarity maps of the images in the center and right columns computed with respect to the image on the left. Smaller patch-size causes noisy fluctuations in similarity of the background regions. Larger patch-size causes the similarity of smaller objects to be influenced by that of larger, surrounding regions.

progresses, gradually more of the building becomes highlighted, before settling on the most salient parts, such as the facade in the first, correct match, and the spire in the second and third matches. By 50,000 iterations, the maps provide a visual indication that training has converged.

4.4. Effect of Patch Size

The paired visualization can be used to examine the effect of varying the patch size in Transformer networks. We trained the same model with a patch size of 32×32 (ViT-B/32) on Hotels-50k and SOP (decreasing the number of tokens to 64 and the output feature map to 8×8).

Comparisons are shown in Figure 7. One difference is that the segmentations around objects are tighter in the ViT-B/16 similarity maps due to the larger number of patches. In the first example in Figure 7, the ViT-B/16 visualization has a tighter boundary surrounding the legs and wheels of the chair. In the ViT-B/32 version, the same components contribute to the similarity, but are not as tightly delineated. In the ViT-B/32 similarity maps, smaller objects tend to match the similarity values of the larger, surrounding regions. This is present in the third example; the hand towels do not strongly contribute to the similarity in the ViT-B/16 visualization, while the surrounding wall does. However, in the ViT-B/32 counterpart, the towels and the wall areas have high similarity contributions.

Higher resolution feature maps also show more instances of seemingly arbitrary variations in similarity compared to surrounding pixels. This is especially noticeable in the second example in Figure 7, where the ViT-B/16 visualization shows background patches that have higher similarity compared to their immediately neighboring regions. One explanation is that, for some networks, the linearity assumption in Rollout is too strong, resulting in a poor approximation of attention flow, an issue raised by Chefer et al. [8]. With fewer tokens, we observe fewer instances of this phenomenon, leading to visually smoother maps like those generated by the ViT-B/32 model.

4.5. Comparison to Output-Agnostic Visualizations

Previous work has used Rollout to produce output-agnostic visualizations of the attention weights [14, 8]. For a given image, we can see that there are substantial differences between its output-agnostic relevancy map and the paired image similarity maps, as shown in Figure 8. In general, the tokens with the greatest relevancy scores are scattered across the image and overlap with areas containing prominent edges. For the pairwise similarity maps, only the features that are shared between images (and therefore contribute to similarity) are highlighted. For instance, in the third example, the Rollout map of the first image shows that patches from each of the four different colored cushions are

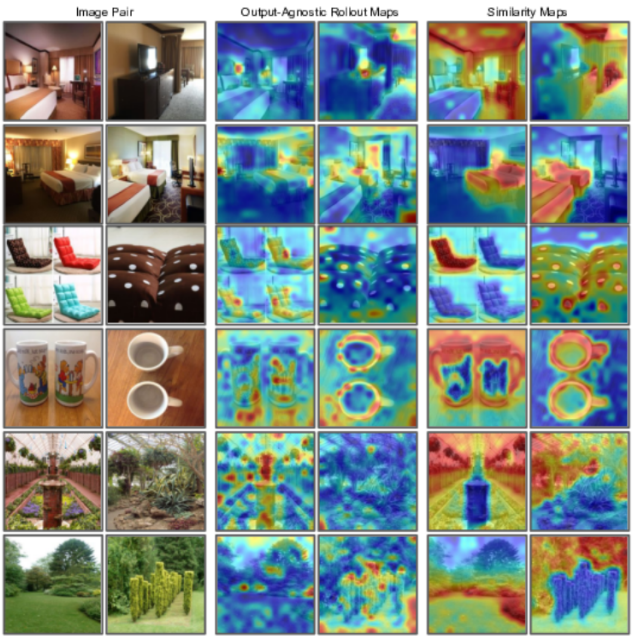

Figure 8. For a pair of images, output-agnostic Rollout maps and their corresponding similarity maps generated with respect to each other. In the Rollout maps, red regions have higher relevancy; in the similarity maps, red regions contribute more to the similarity.

strongly attended to by the model. However, in the similarity visualization, only the brown cushion is highly activated as it is the only match to the paired image. Likewise, in the fourth example, the Rollout map highlights the artwork on the cups, but these features do not strongly contribute when the image is compared to one belonging to the same class but from a very different point of view.

5. Comparison with CNN

For comparison, we also trained a representative CNN model, Big Transfer (BiT) ResNet-101-v2 [22, 17], on the same data for the same tasks. The network is pretrained on the same dataset as the ViT, ImageNet-21k [12]. The ResNet consists of 43M learnable parameters. Similar to the Transformer optimization, we use stochastic gradient descent with a 1-cycle learning rate schedule [32]. We use our approach for the Transformer with [35] for the ResNet.

For each dataset, we follow the most common training and evaluation regimes described in the literature.[2] In each case, the results can be summarized using common retrieval metrics: Recall@1 and mean average precision at R (mAP@R), where R is equal to the number of gallery images that share the same label as the query. Table 2 summarizes the results. In general, we observed similar performance and, frequently, one method would fare better with one metric and the opposite for the other, as seen in these results on the GLM dataset. In most cases, the differences in performance were modest. In this section, we employ our paired visualization method to highlight the differences between the Transformer and ResNet models.

[2]For GLM, we query the images in the test-gallery against each other due to the labels in the more commonly used query-test set being unavailable at the time.

Model	Hotels-50k		GLM		SOP	
	R@1	mAP@R	R@1	mAP@R	R@1	mAP@R
ViT-B/16	.223	.047	.469	.120	.824	.585
ResNet-101	.247	.060	.466	.123	.820	.578

Table 2. Image retrieval comparison between Transformer and ResNet models. For both metrics, higher is better. Both models achieve similar performance on each dataset.

5.1. Image Retrieval Comparison

While both models achieve similar performance in aggregate, the top returned images for a given query are often quite different. The paired image similarity visualizations provide insight into these differences, as shown in Figure 9, which depicts image queries and their top four most similar matches, along with their corresponding similarity visualizations. In the first example, the ViT visualization highlight an assortment of different objects, such as the chair, mini-fridge, and dresser. The similarity maps for the ResNet results instead show high activation solely on the carpet, resulting in a completely different set of images. In the second example, both models return the same correct image for the first result. They both match partially on the artwork, while the ViT also matches on the couch and the ResNet matches on the floor. The remaining returned images in each set also match for these different reasons.

Overall, the similarity maps for the ViT are spread across broader regions, with more objects contributing to the match in comparison to the ResNets, which are primarily focused on localized textures. Consider again the first example in Figure 9. The similarity maps for the ViT show more regions with moderate levels of similarity (yellow in the color scale) compared to the ResNet maps, which are more bimodal – high activation in a localized region with little similarity distributed to the remainder of the image. This observation is consistent with previous research; Bhojanapalli et al. [5] found that ViT models key on large-scale structural features, while ResNets key more on texture.

5.2. Semantic Similarity

For the Hotels-50k dataset, we observed a higher degree of overlap in the objects (e.g., beds, lamps and curtains) between the query and top retrieved matches when compared to the ResNet. For instance, in the last example in Figure 9, the query and top 4 images retrieved by the ViT all show a a TV situated on top of a dresser captured from similar viewpoints. In contrast, the images returned using the ResNet model show a more diverse set of perspectives.

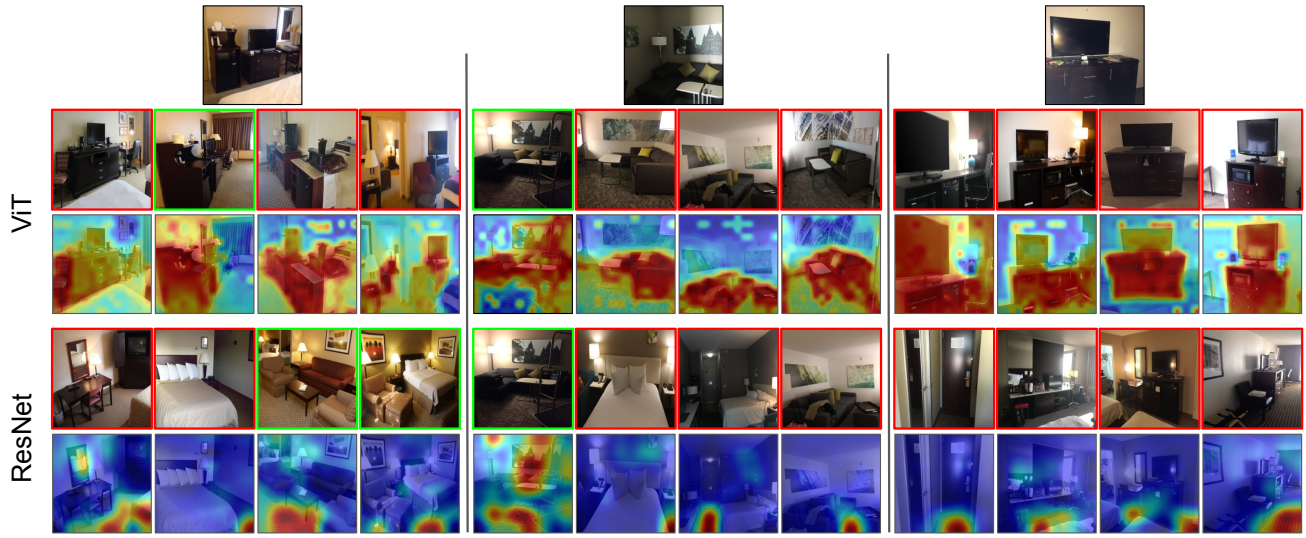

Figure 9. Image retrieval queries on Hotels-50k dataset (top row) and their the top-4 retrieved results for the ViT and the ResNet, along with their similarity maps computed with respect to the query. A green border indicates a correct retrieval.

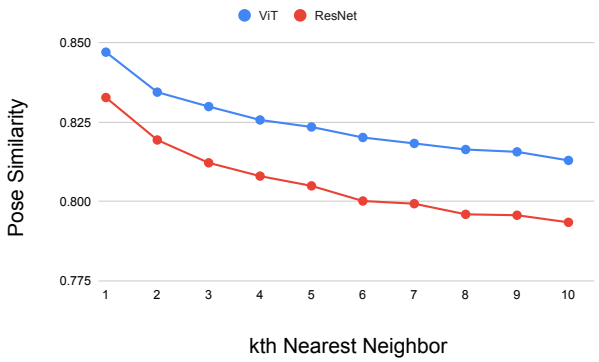

Figure 10. On average, the top matching images from a Transformer model have higher similarity of objects and camera pose to the query than for the ResNet, which suggests the presence of common objects contributes more strongly to similarity for Transformers than ResNets.

To quantify this phenomenon, we compute the *pose similarity* between the query and its top matching results. For each image, we compute a dense semantic segmentation and vector of the pixel semantic class distribution. The pose similarity is the dot product of these vectors computed for a pair of images. The pose similarity is a rough measure of whether two images contain the same set of objects in similar poses. For this experiment, we use the a ResNet-50 + PPM DeepSup network [47] trained on the ADE-20k dataset [46, 48]. The model is trained to predict 150 semantic classes. We retain the 44 classes that appear in at least 5% of the Hotels-50k images, and consolidate the remainder into an "other" category. For each query, we compute the pose similarity to the top 10 retrieved images, and the averages are shown for the Transformer and ResNet in Figure 10. Matching images from a Transformer generally share higher pose similarity with the query than those from a ResNet. This finding reinforces the notion that the image similarity learned differs between the two architectures and that larger, sometimes global, similar patterns between images play a larger role for Transformers.

6. Conclusions

Visualization methods have filled an important gap in the understanding of neural networks in computer vision. We presented a paired-image visualization approach for Transformers trained for image embedding. We demonstrated the utility of this approach to better understand image retrieval with Transformers and the differences between the learned representations compared to convolutional networks. While the two approaches may achieve similar performance using aggregate metrics, similarity can be encoded very differently. Transformers tend to encode larger regions and the entirety of objects, while convolutional embedding networks appear to encode smaller textured regions. Moreover, camera pose appears to be more of a factor in encodings generated by Transformers. Such insight may be useful in determining which model to choose for various image domains and tasks beyond image embedding.

Acknowledgements

This work was supported by the U.S. DOJ under award number 2018-75-CX-0038. In addition, calculations were carried out on HPC resources supported in part by the NSF through grant number 1625061 and by the U.S. ARL under contract number W911NF-16-2-0189.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online, July 2020. Association for Computational Linguistics.

[2] Plamen P. Angelov, Eduardo A. Soares, Richard Jiang, Nicholas I. Arnold, and Peter M. Atkinson. Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery*, 11(5):e1424, 2021.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 6541–6549, 2017.

[5] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*, 2021.

[6] Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *International Conference on Learning Representations*, 2020.

[7] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. *arXiv preprint arXiv:2103.15679*, 2021.

[8] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021.

[9] Boyu Chen, Peixia Li, Baopu Li, Chuming Li, Lei Bai, Chen Lin, Ming Sun, Junjie Yan, and Wanli Ouyang. Psvit: Better vision transformer via token pooling and attention sharing. *arXiv preprint arXiv:2108.03428*, 2021.

[10] Lei Chen, Jianhui Chen, Hossein Hajimirsadeghi, and Greg Mori. Adapting grad-cam for embedding networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2794–2803, 2020.

[11] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention. In *BlackboxNLP@ACL*, 2019.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[15] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *arXiv preprint arXiv:2102.05644*, 2021.

[16] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. In *Asian Conference on Computer Vision*, pages 119–134. Springer, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[18] Alexander Hermans*, Lucas Beyer*, and Bastian Leibe. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*, 2017.

[19] Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4176–4185. IEEE, 2019.

[20] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. 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 3543–3556, Stroudsburg, PA, USA, 2019. Association for Computational Linguistics.

[21] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3238–3247, 2020.

[22] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Proc. European Conference on Computer Vision*, pages 491–507. Springer, 2020.

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[24] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable visual transformers with hierarchical pooling. *arXiv preprint arXiv:2103.10619*, 2021.

[25] Frank Pasquale. *The black box society*. Harvard University Press, 2015.

[26] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. Learning to deceive with attention-based explanations. In *The 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Seattle, USA, July 2020.

[27] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[28] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[29] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[30] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713, 2016.

[31] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[32] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

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

[34] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] Abby Stylianou, Richard Souvenir, and Robert Pless. Visualizing deep similarity networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 2029–2037. IEEE, 2019.

[36] Abby Stylianou, Hong Xuan, Maya Shende, Jonathan Brandt, Richard Souvenir, and Robert Pless. Hotels-50k: A global hotel recognition dataset. In *Proc. National Conference on Artificial Intelligence*, 2019.

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proc. International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017.

[38] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[39] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2020.

[40] Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *Proceedings of the Web Conference 2021*, WWW '21, page 2466–2475, New York, NY, USA, 2021. Association for Computing Machinery.

[41] Hong Xuan, Richard Souvenir, and Robert Pless. Deep randomized ensembles for metric learning. In *Proc. European Conference on Computer Vision*, September 2018.

[42] Shuhei Yokoo, Kohei Ozaki, Edgar Simo-Serra, and Satoshi Iizuka. Two-stage discriminative re-ranking for large-scale landmark retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1012–1013, 2020.

[43] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[44] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

[45] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[46] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 2018.

[48] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.