Fashion Image Retrieval based on Parallel Branched Attention Network

Sangam Man Buddhacharya[1]
Department of Electronics and
Computer Engineering
Institute of Engineering, Pulchowk Campus
Lalitput, Pulchowk, Nepal

Sagar Adhikari[2]
Department of Electronics and
Communication Engineering
Paschimanchal Campus
Pokhara, Nepal

Ram Krishna Lamichhane[3]
Department of Electronics and
Communication Engineering
Paschimanchal Campus
Pokhara, Nepal

Abstract—**With the increase in vision-associated applications in e-commerce, image retrieval has become an emerging application in computer vision. Matching the exact user clothes from the database images is challenging due to noisy background, wide variation in orientation and lighting conditions, shape deformations, and the variation in the quality of the images between query and refined shop images. Most existing solutions tend to miss out on either incorporating low-level features or doing it effectively within their networks. Addressing the issue, we propose an attention-based multiscale deep Convolutional Neural Network (CNN) architecture called Parallel Attention ResNet (PAResNet50). It includes other supplementary branches with attention layers to extract low-level discriminative features and uses both high-level and low-level features for the notion of visual similarity. The attention layer focuses on the local discriminative regions and ignores the noisy background. Image retrieval output shows that our approach is robust to different lighting conditions. Experimental results on two public datasets show that our approach effectively locates the important region and significantly improves retrieval accuracy over simple network architectures without attention.**

Keywords—Convolutional neural network (CNN); image retrieval; attention mechanism; convolutional block attention module (CBAM)

I. Introduction

In the last decade, due to our increased computational ability, there have been tremendous improvements in Deep learning [1], [2] and Computer Vision, leading to an exponential proliferation of applicational possibilities. Among the various engineering applications of computer vision ranging from Drug Design [3] to Monocular depth estimation [4], image retrieval has become an emerging one. This particular application has both academic and business ramifications. Academically, it can bring about new innovative approaches to solving image comparison problems, whereas commercially, it can create a disruptive shopping experience for the users. Among all the product categories, due to its dynamic product nature, variations, and immense use case, Clothing/Fashion has received the highest amount of attention.

When similar kinds of images (i.e., consumer to consumer or shop to shop) are compared, there is a certain homogeneity in the images. Thus, they can be treated as from the same domain, not neglecting multiple variations such as lighting, view, backgrounds, product orientation, etc. Nonetheless, comparing different kinds of images (professional with amateur) will contain images from other domains.

Despite the difference in image types, these comparisons can be achieved by analyzing the human-detectable details in the clothes, such as cloth category, color, pattern, prints on the clothes, and so on. Most current retrieval solutions [[5], [6], [7], [8], [9], [10]] incorporate deep learning models that convert actual images into vector representation so that the query image's embedding can be compared against all the images' embeddings from the list, and the closest one can be returned. For that, triplet loss is the most widely used comparative loss technique. As suggested by [[11], [10], [12], [13]], despite being superior to other approaches, the triplet loss approach has its demerits, such as the inability to achieve top performance, being computationally expensive, and being prone to noisy labels and outliers. To mitigate that improvement has been proposed by using the Centroid Triplet Loss function in [14].

Nevertheless, as described in [9], high intra-class variability in clothes and the possibility of different kinds of deformations for the same type of clothes were the significant hurdles for achieving the most acceptable retrieval results. The problem with most of these existing approaches is that it ignores low-level features and those which use low-level features take all the information without selecting discriminative features which introduce noise. Deep networks, which are being used as a solution, tend to go deep and lose vital information from low-level features. Shallow networks can provide those low-level features, but the output is prone to noise. Thus, some form of noise elimination is required. Attention mechanisms emphasize the essential features and suppress the non-essential features. CBAM [15] sequentially applies channel and spatial attention along the respective principle dimensional axes to achieve the same. A shallow network - combined with the attention layer - outputs noiseless low-level features. Thus we have proposed a new architecture that utilizes both deep and attention-shallow networks to extract high-level and discriminative low-level features.

Along with the new architecture proposal, other factors were also considered for improving the overall retrieval accuracy. Here are our contributions:

1) Experimentation with multiple architectures for Image retrieval.
2) Propose a new attention-based architecture for better retrieval performance.
3) Experimentation with the impact of image size on the model's performance.

4) Experimentation with different classification models as our backbone network.
5) Performance comparison across multiple fashion data datasets (DeepFashion [6] and DeepFashion2 [16]).

II. Problem Statement

From a consumer's point of view, there might be different scenarios where a user could benefit from various forms of fashion image comparison and automated searches. All such applications usually include these three kinds of image comparisons:

- Image comparison between a shop image with another shop image.
- Image comparison between a shop image with a consumer image.
- Image comparison between a consumer image with another consumer image.

Due to different image-type comparisons, we prioritize selecting distinctive features in fashion pattern matching, which - moreover - deals with these three main problems in pattern matching:

1) Common images contain different backgrounds, which are usually noisy features for the model. Even after cropping only the target section, the remaining background will still dominate the distinctive features and reduce the model's overall performance.
2) Clothes might contain only a small portion of areas that might cause differentiation from other clothes. Nevertheless, when we use all the features from the clothes to compare the similarity, there might be a low influence of the distinctive features, reducing the performance. Since the distinguishing area varies according to the type of clothes, we need a dynamic module that will focus more on those discriminating features.
3) The existing deep convolution networks - rightfully so - suppress the non-crucial features. While doing so, the low-level features are also being ignored in such a way that it is impacting the retrieval accuracy.

III. Methodology

In this section, we describe our proposed architecture (PAResNet50) with a two branched variation (DBAN) along with loss function, and augmentation policy used during training and testing the network.

A. Architecture

We use a deep Convolutional Neural Network (CNN) to generate feature embeddings. The feature vector is the abstract representation of patterns, color, and shape of the input images, which helps to distinguish between the two different clothes. We use a triplet-based network architecture with the ranking loss function to learn the feature vectors. As shown in the Fig. 1a, the three triplets q, p, and n are independently fed into three different deep CNN, which share similar architecture and parameters. The deep CNN computes respective feature embeddings $(\overrightarrow{q}, \overrightarrow{p}, and \overrightarrow{n})$ for triples p, q and n.

Inspired by [17], we use multiscale deep CNN. Our implementation is quite different than [17], we use ResNet-50 [18] instead of Alexnet [19] and a series of convolutional and CBAM [15] layers. As shown in Fig. 1b, it has two different parallel branches coupled at conv1 of ResNet-50 [18]. The two parallel branches are downsampled with 4:1 and 8:1 ratios respectively. The downsampled branches are followed by 3x3 convolutional and CBAM [15] layers, flattened to extract low-level features. The output from $conv5\_block3$ of ResNet-50 [18] is followed by a 1x1 convolutional layer and global average layer to extract high-level features. The high-level and low-level features are concatenated and followed by a dense layer to output the final embedding. Introducing an attention mechanism in the shallow branches helps the model to focus on the low-level details like color, texture, and materials regarding its shape. Since the low-level features have lots of noise, reducing the retrieval performance, we used CBAM [15] as the attention module to enhance the essential features while fading out the non-relevant information.

During image retrieval, the embeddings of each image are extracted, and cosine similarity between the embeddings is calculated to find the best matching clothes. Distance between the embeddings estimates the similarity or dissimilarity between the images. Similar images are closer in the embedding space while the dissimilar images are distant.

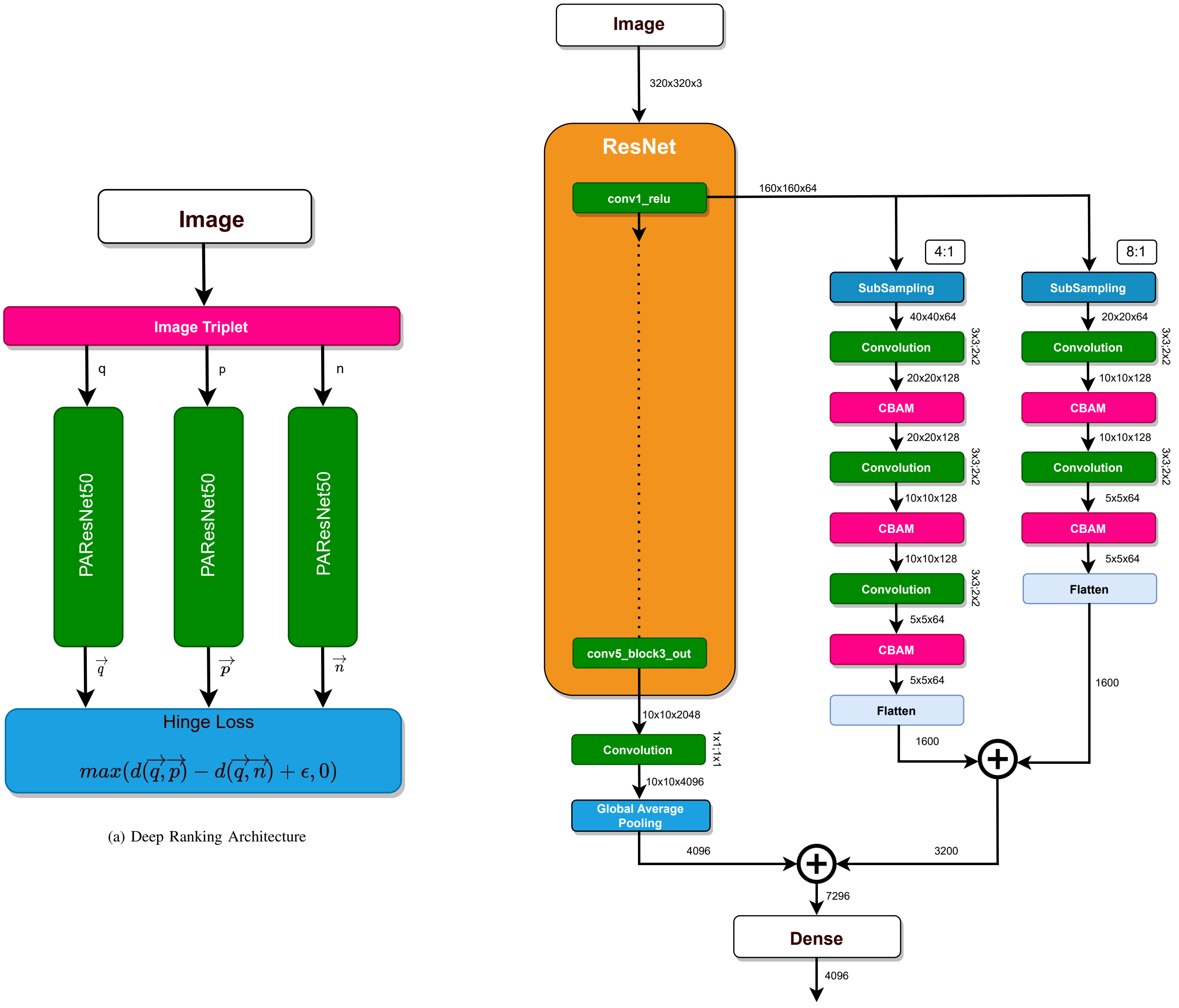
B. Attention Mechanism

Attention mechanism is a technique by which computers try to simulate how human vision focuses in terms of computer-based algorithms. It is a method that tries to enhance the significant parts while fading out the non-relevant information. It can dynamically adjust the weights based on features of the input image.

We use Convolutional Block Attention Module (CBAM [15])as our attention module. As in Fig. 2, CBAM [15] is composed of two sequential sub-modules, the Channel Attention Module (CAM) and the Spatial Attention Module (SAM).

a) Channel Attention Module: Channels are feature maps stacked in a tensor, where each cross-sectional slice is, basically, a feature map of dimension (h x w). The input feature map of the channel is regarded as a feature detector. Channel attention is calculated by compressing the feature map in the spatial dimension using max pooling and average pooling to obtain two different spatial context descriptors. The descriptors are fed into a shared network to produce a feature vector. The shared network comprises an MLP (Multi-Layer Perceptron) and one hidden layer. The output feature vectors from MLP are merged using element-wise summation, and the sigmoid function is applied to compute the channel attention map.

b) Spatial Attention Module: Spatial attention represents the attention mechanism masks on a single cross-sectional slice of the tensor or each feature map representing the Spatial Attention Map. As in Fig. 2, Spatial attention is calculated with the two different feature descriptions obtained from maximum pooling and average pooling in the channel dimension. The two feature descriptions are merged, and a convolutional operation is applied to generate a spatial attention map.



(a) Deep Ranking Architecture

(b) Dual Branched Attention Network of PAResNet50

Fig. 1. Overall Architecture of PAResNet50 with Deep Ranking

C. Loss Function

We have used a triplet loss function with batch-all online-mining strategies. A batch of B embedding is extracted from a batch of B inputs. B is composed of C different styled clothes with N images each. A valid triplet $\vec{q}, \vec{p}, \vec{n}$ is generated from B embeddings. These three indices $(\vec{q}, \vec{p}, \vec{n}) \in [1, B]$ are query, positive and negative pairs, respectively. Batch all online mining produces a total of T (1) valid triplets

$$T = C * N * (N - 1) * (C * N - N) \tag{1}$$

where $C * N$ is the number of query images, $N - 1$ is the possible positive pair per query images and $C * N - N$ is the possible negative pair. Hinge loss is calculated from each valid triplets $(q, p, n) \in [1, B]$.

$$l(\vec{q}, \vec{p}\,\vec{n}) = max(d(\vec{q}, \vec{p}) - d(\vec{q}, \vec{n}) + \epsilon, 0) \tag{2}$$

where, ϵ is the margin and $d(\vec{x}, \vec{y})$ is the Euclidean Distance between $\vec{x}\,and\,\vec{y}$. The hinge loss function tries to push $d(\vec{q}, \vec{p})$ to 0 (i.e. pulling $\vec{q}\,and\,\vec{p}$ closer) and $d(q, n)$ to be greater than $d(\vec{q}, \vec{p}) + \epsilon$ (i.e. pushing $\vec{q}\,and\,\vec{n}$ farther). Our final training loss L is as follows:

$$L = \sum_{(\vec{q}, \vec{p}\,\vec{n}) \in B}^{T} l(\vec{q}, \vec{p}\,\vec{n}) \tag{3}$$

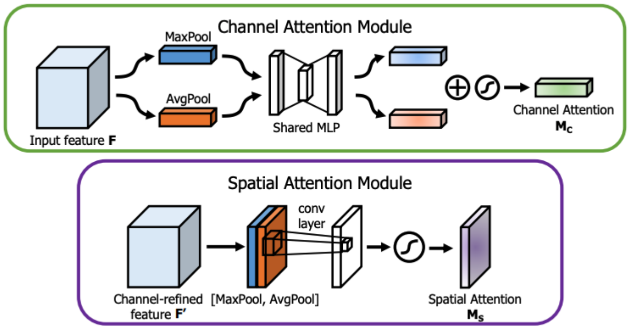


Fig. 2. Structure of Spatial and Channel Attention in CBAM. Source: [15].

D. Training Data Generation

For both DeepFashion [6] and DeepFashion2 [16] datasets we used the provided benchmark training sets and placed clothes with same style in a common style folder. These style folders were kept in their respective category folder. Each image was cropped with provided bounding boxes. For DeepFashion2 [16], back-faced and heavily occluded data were removed. We created a list of all the images available in the folder. Two images from each style/group were randomly selected during training to create a batch. The selected pairs were excluded from the list until the next epoch. Epoch is completed when there is no image pair left in the list.
We have only used geometric augmentation for both the query and shop images. The input images are horizontally flipped with 50% chance and rotated randomly in a range of $[-1, 1]$ degrees. This helps to increase the generalization performance and avoid over-fitting. Colour augmentation might change the original color of both query and shop images which might cause the corresponding pairs to be dissimilar, so we didn't use colour augmentation.

IV. Experiments

A. Datasets

a) DeepFashion [6]: The dataset contains over $800,000$ images with the information of categories, landmarks, bounding boxes, clothes attributes, and image pairs for Consumer-to-Shop/In-shop clothes retrieval. For this paper, we have only used the Consumer-to-Shop Clothes Retrieval subset which contains $33,881$ unique clothing products, $239,557$ consumer and shop images and $195,540$ consumer and shop matching pairs.

b) DeepFashion2 [16]: The dataset contains $491k$ diverse images from both consumers and shopping where each item is labeled with scale, occlusion, zoom-in, viewpoint, category, style bounding box, dense landmark, and per-pixel mask. For this paper, we only use Commercial-Consumer clothes pairs which continents $319k$ training sets, $34k$ validation sets, and $67k$ test sets. From the available dataset, we removed back-face and heavily occluded clothes during training.

B. Implementation Details

For the implementation, Keras [20] and TensorFlow [21] have been used as our deep learning framework. Likewise, Adam optimizer has been used to train the model with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$ and an exponential decaying learning rate of 0.96 for every 150000 steps with starting learning rate of 10^{-6}. For online triplet loss, margin of 1.5 has been set. We have used a batch size of 4 composed of 2 different styles of clothes with 2 images each. Each model has been trained for different iterations; the training is stopped according to the model's performance on validation loss. All the experiments have been performed in Kaggle with NVidia K80 GPU.

C. Evaluation Metrics

For the evaluation of retrieval performance, we use top-k accuracy, as in [[22], [6]]. The top-k accuracy is defined as follows:

$$P(K) = \frac{\sum_{q \in N} hit(q, K)}{|N|} \tag{4}$$

where, N is the total number of queries performed.

$hit(q, K) = 1$ is a hit, if at least one shop image appears within the top-K ranking for the query image q.

$hit(q, K) = 0$ is a miss, if no any shop image appears with in the top-K ranking for the query image q.

D. Experiments with Different Embedding Layers

In this experiment, we have used different embedding layers keeping other parameters unchanged. We used Flatten layer, Spatial Pyramid Pooling(SPP) layer, and Global Average layer after $conv5\_block3$ of ResNet-50 [18]. Table I shows that the flatten layer has the highest number of feature vectors with the lowest accuracy. But the global average layer has less number of feature vectors with the highest accuracy. In the flatten layer, redundant features and noise reduced the influence of discriminative features. But in the global average layer, there are mostly discriminative features. Therefore the retrieval performance depends upon the size of the feature vector. We didn't find SPP efficient compared to Global Average, so we used GlobalAverage as our embedding layer to extract high-level features.

TABLE I. Comparison of Top-k (k= 1, 5, 10, 20, 50) Retrieval Accuracy on DeepFashion2 [16] Dataset for Different Embedding Layers Performed on 256x256 Image Size.

Last layer	# size	mAP	top-1	top-5	top-10	top-20	top-50
Flatten	65536	0.687	0.445	0.629	0.712	0.784	0.865
SPP	21504	0.720	0.485	0.663	0.743	0.815	0.893
GlobalAvg	4096	**0.785**	**0.576**	**0.747**	**0.812**	**0.863**	**0.927**

E. Experiments with Different Backbone Networks

In this section, we have experimented with different classification models to find the best retrieval performance. From Table II, it can be clearly observed that ResNet-50 [18] architecture has significantly higher performance in comparision to VGG-16 [23], and MobileNetV1 [24], so we used ResNet-50 [18] as our backbone network in PAResNet50 [1].

TABLE II. COMPARISON OF TOP-K (K= 1, 5, 10, 20, 50) RETRIEVAL ACCURACY ON DEEPFASHION2 DATASET FOR DIFFERENT ARCHITECTURES PERFORMED ON 256X256 IMAGE SIZE.

Models	mAP	top-1	top-5	top-10	top-20	top-50
VGG-16 [23]	0.699	0.453	0.633	0.715	0.804	0.894
MobilenetV1 [24]	0.566	0.315	0.486	0.572	0.665	0.793
ResNet-50 [18]	**0.798**	**0.588**	**0.761**	**0.822**	**0.882**	**0.937**

F. Experiments with Different Image Size

To find the influence of image size in PAResNet50, we have experimented with different image sizes while keeping other parameters constant. From Table III, we found the input images of size 320x320 to be the best for our settings. Therefore, a larger image size helps to increase the retrieval performance so we used 320x320 image size in PAResNet50 for both training and testing.

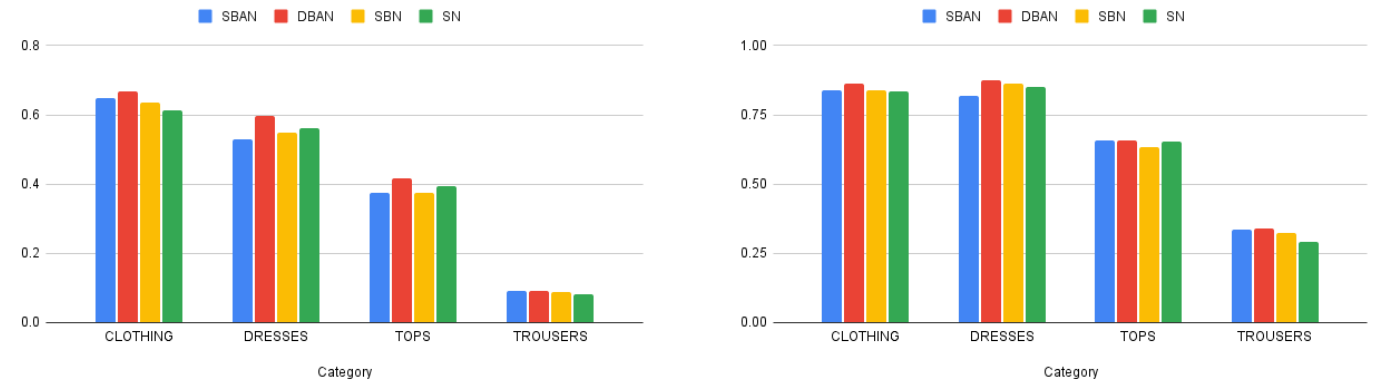
TABLE III. COMPARISON OF TOP-K(K=1,5,10,20,50) RETRIEVAL ACCURACY ON DEEPFASHION2 DATASET FOR DIFFERENT IMAGE SIZES.

Image size	mAP	top-1	top-5	top-10	top-20	top-50
256x128	0.7896	0.567	0.746	0.817	0.879	0.939
256x256	0.798	0.588	0.761	0.822	0.882	0.937
320x320	**0.813**	**0.617**	**0.774**	**0.834**	**0.895**	**0.943**

G. Experiments with Different Architectures

We experimented with different kinds of architectures. They are as follows:

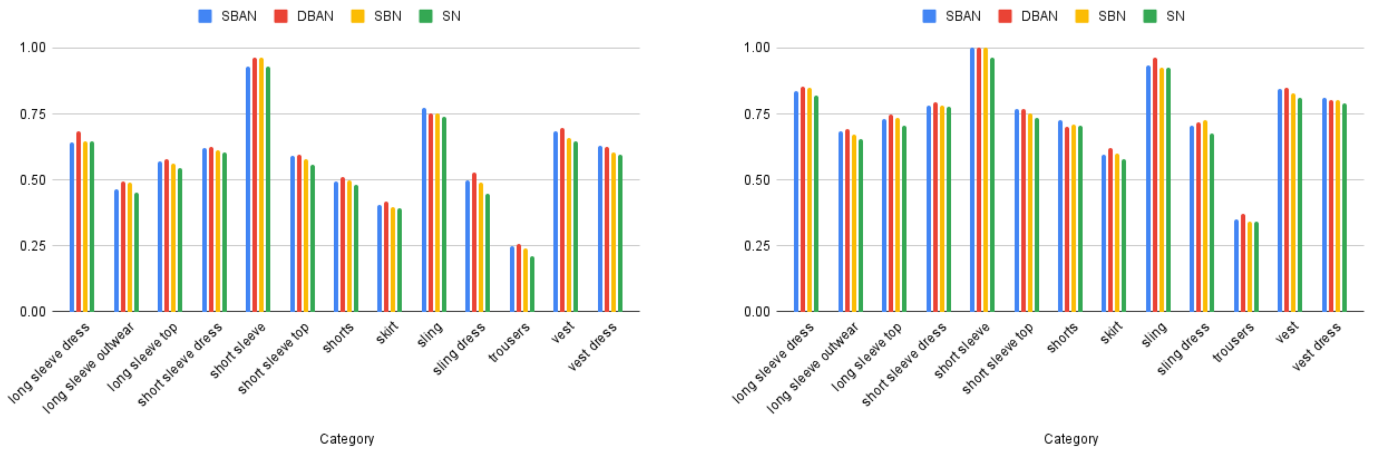
a) Simple Network(SN): It is a simple ResNet-50 [18] classification model pre-trained on Imagenet [25]. The output from $conv5\_block3$ of ResNet-50 [18] is followed by 1x1 convolutional layer, global average layer and a dense layer to extract a feature embeddings.



(a)

(b)

Fig. 3. a) and b) are the Top-1 and Top-5 Categories Retrieval Accuracy on DeepFashion [6] Validation Set. Each Model is Trained on Image Size of 320x320.



(a)

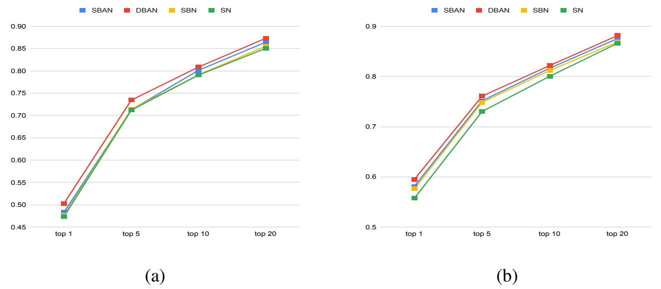
(b)

Fig. 4. a) and b) are the Top-1 and Top-5 Categories Retrieval Accuracy on DeepFashion2 [16] Validation Set. Each of the Model is Trained on Image Size of 256x256.

b) Single Branched Network(SBN): It is an extention to already existing Simple Network. A ResNet-50 [18] classification model coupled with a parallel branch. In the parallel branch, the output from $conv1\_relu$ is downsampled with a ratio of 4:1 and a series of convolutional layers is used. The output from the global average layer and the parallel branch is concatenated which is followed by a dense layer to extract the final feature embeddings.

c) Single Branched Attention Network (SBAN): This follows the architecture of Single branched network (SBN) here the convolutional layer in the parallel branch is followed by the CBAM [15] layer.

d) Dual Branched Attention Network (DBAN/PAResnet50): It is our final model, which has shown the best performance. It has two parallel branches with downsampling of 4:1 and 8:1, respectively. After downsampling on each branch, a series of convolutional and CBAM [15] layer is used which is followed by a flatten layer to extract low-level features. The outputs from the global average layer and the two parallel branches are concatenated and followed by a dense layer to extract the final feature embeddings. The architecture of PAResNet50 is shown in the Fig. 1.



(a)

(b)

Fig. 5. a) and b) are the Comparison of Top-k(k=1,5,10,20) Retrieval Accuracy for Different Architecture on DeepFashion [6] and DeepFashion2 [16] Dataset Respectively. Each of the Model is Trained on Image Size of 256x256.

e) Architecture Comparison: From the Fig. 5, we observed that using low-level features directly from the branched network (SBN) slightly increases the model's performance compared to Simple Network (SN). The increase in performance is due to addition of low-level features. Adding a CBAM [15] layer in the branch ($SBAN$) further improves the performance since the attention mechanism suppresses the noises from low-level features. When two branches with an CBAM [15] layer ($PAResNet50$) are used to extract the low-level features, the model gets on additional fashion details (i.e., color, styles, and patterns) to learn, which significantly increases the retrieval performance. Therefore Dual Branched Attention Network(DBAN/PAResnet50) has higher retrieval accuracy in comparison to other architectures. We also applied attention mechanism on high level features by adding CBAM [15] layer on different blocks of ResNet-50 [18], but it didn't improve the performance. The attention mechanism didn't work well on high-level features.
To better analyze each architecture's performance, we evaluated the top-1 and top-5 retrieval accuracy for each category on both Deepfashion [6] and Deepfashion2 [16] datasets. Fig. 3 shows that PAResnet50 has improved the top-1 and top-5 retrieval accuracy on DeepFashion [6] for clothing, dresses, and tops while slightly improving in trousers. From the Fig. 4, we can see that on DeepFashion2 [16], PAResnet50 has the highest top-1 retrieval accuracy in all categories except sling and vest dress. In the top-5 retrieval accuracy, it has also performed well in the sling category.

H. Results on Deepfashion [6] and Deepfashion2 [16] Dataset

TABLE IV. Comparison of PAResNet50's [1] Top-k (k=1,5,10,20,50) Retrieval Accuracy on Deepfashion [6] and Deepfashion2 [16] Datasets with 320x320 Image Size

Datasets	mAP	top-1	top-5	top-10	top-20	top-50
DeepFashion [6]	0.771	0.503	0.733	0.810	0.873	0.936
DeepFashion2 [16]	0.813	0.617	0.774	0.834	0.895	0.943

We trained PAResNet50 on both DeepFashion [6] and DeepFashion2 [16] datasets with image size of 320x320. Table IV shows that DeepFashion2 [16] dataset has higher performance in comparison to DeepFashion [6] since we have removed the back-faced and highly occluded images in DeepFashion2 [16], which reduced the conflict invalid image pairs. The back-faced and occluded clothes might have different colors, patterns, and texture, so when paired together, it forms invalid pair and decreases the training performance.

Results from Table IV confirm that our proposed model PAResNet50 is suitable for fashion image retrieval on different e-commerce websites.

I. Query Results

To better understand the output quality of PAResNet50, we analyzed the query results on different category images as shown in Fig. 6. The output is categorized into three groups best, good, and bad. The top three rows are the best output, retrieving the corresponding shop image in the top-1 list. The fourth and fifth rows are the good outputs, retrieving the corresponding shop image in the top-3 list. The bottom row is the bad output where the pair shop doesn't occur within the top-3 list. We can observe that our model can retrieve perfect matching images by learning fashion details such as colors, styles, patterns, and textures. In the second row first query, our model has retrieved the exact shop image even if the cloth is not worn (shape deformed). With results from the first-row second query and second-row second query, we can see that even under different lighting conditions, our model has delivered the exact shop image in the top-1 list. Therefore, our model is robust to different lighting conditions. In the second last row of Fig. 6, although the exact shop image is not retrieved in the top-1 list, visually similar colors and pattern-styled clothes are retrieved, which is a more challenging task for a human being. In the bottom row, even though the exact shop image doesn't appear in the top-3 list, the retrieved images are significantly similar to the query image.

J. Attention Visualization

To find the effect of the attention mechanism (CBAM [15]), we have visualized the attention map from PAResNet50. We

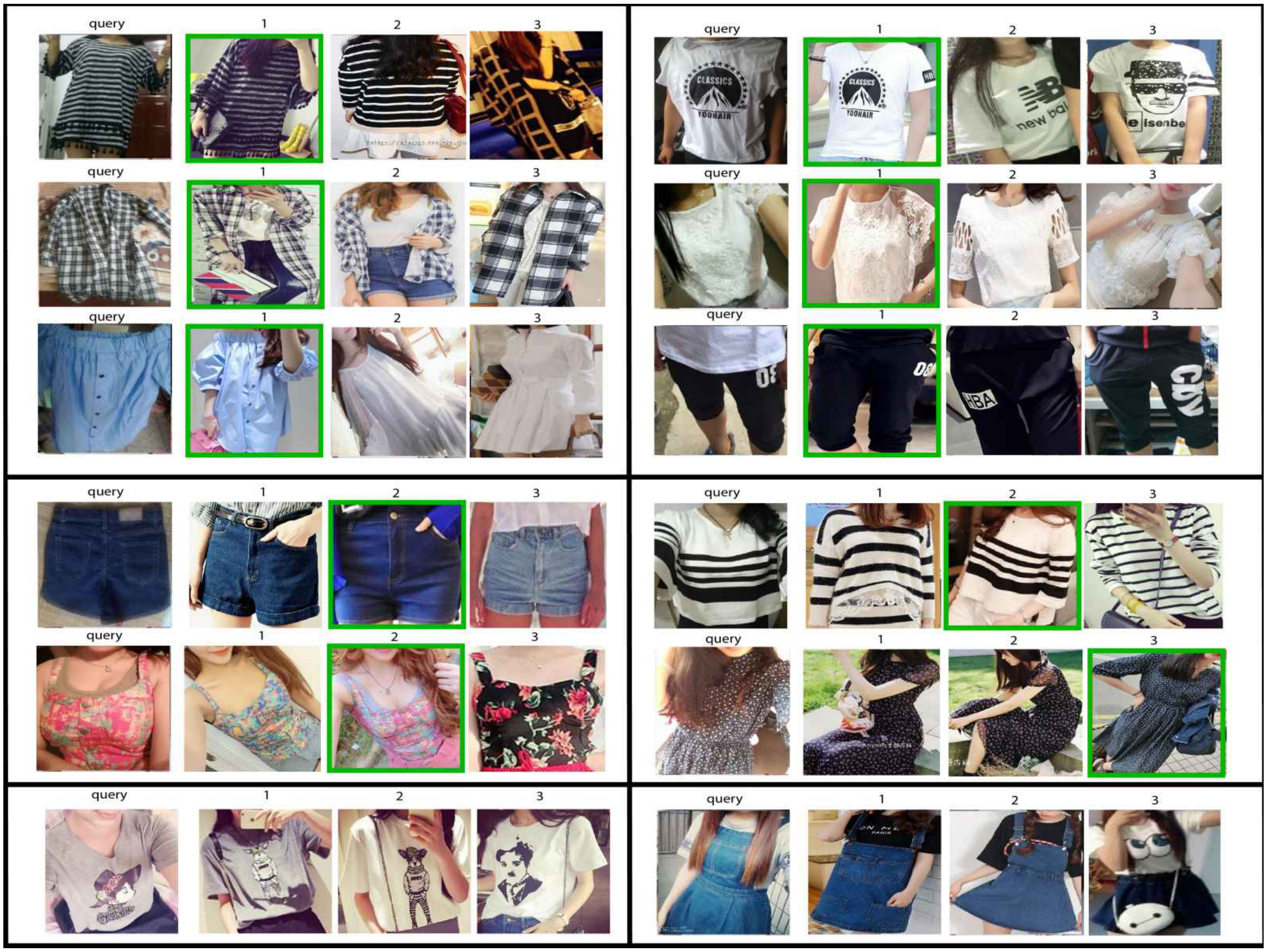


Fig. 6. Top-3 Retrieved Images for a Given Query on DeepFashion2 [16] Dataset. Green Box Indicates the Corresponding Shop Image.

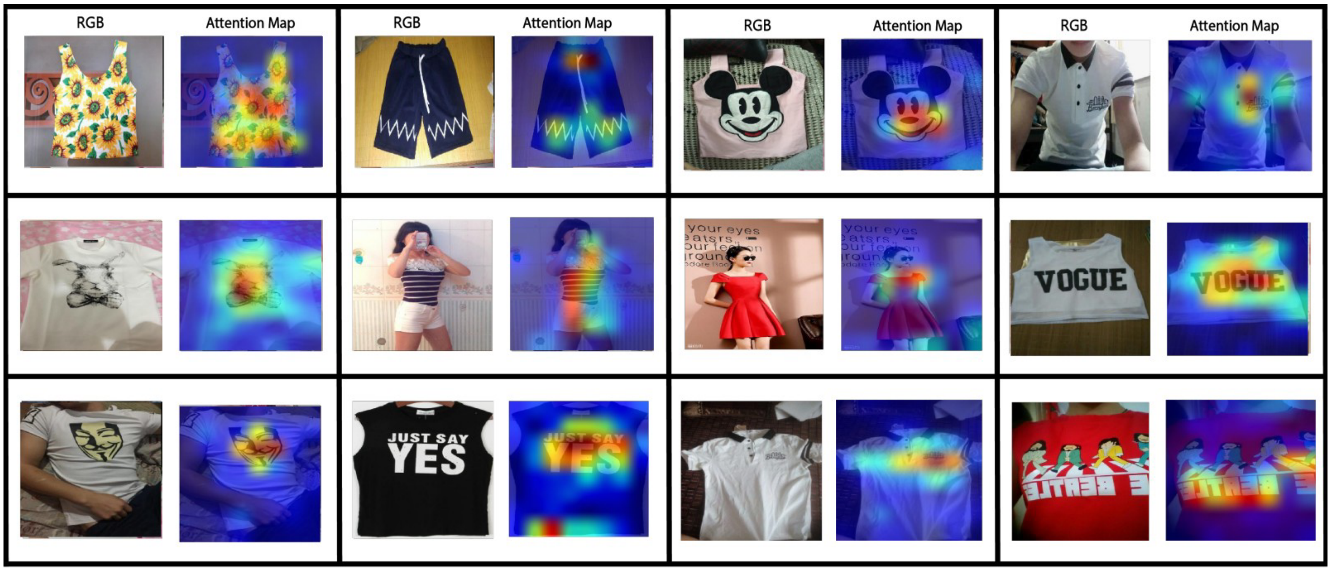


Fig. 7. Visualization of Attention Map in the Query Images from PAResNet50 1. Red Indicates Higher Important Region while Blue Indicates Lower Important Regions.

used Grad-CAM [26] for visualization. From Fig. 7, we can clearly observe that our model has mainly focused on the local discriminative regions (e.g. logos, pictures, patterns, and text) in an input image while ignoring non-discriminative regions (e.g. background, plain region, and hand). Therefore only the discriminative features are used to find the matching images, which increases the retrieval performance of the model. Attention mechanism on the branch layer helps the network focus on only the important features while ignoring the less significant ones.

K. Experimental Summary

Overall, multiple experiments were conducted to find the best settings for image retrieval tasks. Table II, which is the comparison of different classification models (VGG-16, MobileNet, and ResNet-50), shows that ResNet-50 outperforms other models with a minimum margin of 10 percent in mAP metrics. Likewise, Table I clearly depicts that using ResNet-50 architecture with Global Average as embedding layer has performed the best with top-k (k=1, 5, 10, 20, 50) accuracy as 0.576, 0.747, 0.812, 0.863, 0.927 respectively. Further, to show the importance of low-level features and attention mechanisms in image retrieval tasks, we experimented with different architectures. Experimental results from Fig. 5, clearly indicate that Dual Branched Attention Network (DBAN) has achieved the highest retrieval accuracy. Analyzing the Fig. 3 and Fig. 4 demonstrates that DBAN works best in almost all categories. Also, the experiment concluded to observe the influence of different image sizes displays that higher resolution increases the model retrieval performance. As shown in Table III, an image size of 320x320 works best for DBAN. The query output from Fig. 6 helps to better understand the quality of DBAN which shows that this model retrieves visually similar colors and pattern-styled clothes and is robust to different lighting conditions. To further show the attention region of the DBAN, we have visualized the attention map in Fig. 7. We observed that the model has primarily focused on discriminative features. Therefore, it confirms that the attention mechanism ignores the noisy background.

V. Conclusion

In this paper, we have designed the PAResNet50 architecture to present the importance of the low-level features with an attention mechanism for image retrieval tasks. We found that two coupled attention branches in Dual Branched Attention Network(DBAN/PAResNet50) learn low-level fine details and effectively locate the local discriminative regions while ignoring non-significant areas. From various experiments, it can be inferred that incorporating low-level discriminative features along with high-level features improves retrieval performance. The query results exhibit the usability of PAResNet50 in a variety of categories for different e-commerce purposes. Experiments with different architectures(SN, SBN, SBAN, and DBAN) on two public datasets, DeepFashion, and DeeFashion2, demonstrate that DBAN(PAResNet50) outperforms other architectures with fewer or no attention branches. This result leaves room for the possibility of future enhancement in the retrieval accuracy by experimenting with a greater number of such multiscale attention branches.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. Aryal, M. Gupta, and M. Abdelsalam, "A survey on adversarial attacks for malware analysis," *arXiv preprint arXiv:2111.08223*, 2021.

[3] A. Dhakal, C. McKay, J. J. Tanner, and J. Cheng, "Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab476, 2022.

[4] S. M. Buddhacharya, R. Adhikari, and N. Maharjan, "Monocular depth estimation using a multi-grid attention-based model," *Journal of Innovative Image Processing*, vol. 4, no. 3, pp. 127–146, Aug. 2022. [Online]. Available: https://doi.org/10.36548/jiip.2022.3.001

[5] H. Diao, Y. Zhang, L. Ma, and H. Lu, "Similarity reasoning and filtration for image-text matching," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1218–1226.

[6] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1096–1104.

[7] H. Jun, B. Ko, Y. Kim, I. Kim, and J. Kim, "Combination of multiple global descriptors for image retrieval," *arXiv preprint arXiv:1903.10663*, 2019.

[8] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, and J. Gu, "A strong baseline and batch normalization neck for deep person re-identification," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2597–2609, 2019.

[9] M. Wieczorek, A. Michalowski, A. Wroblewska, and J. Dabrowski, "A strong baseline for fashion retrieval with person re-identification models," in *International Conference on Neural Information Processing*. Springer, 2020, pp. 294–301.

[10] Y. Yuan, W. Chen, Y. Yang, and Z. Wang, "In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 354–355.

[11] T.-T. Do, T. Tran, I. Reid, V. Kumar, T. Hoang, and G. Carneiro, "A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 404–10 413.

[12] Z. Zhang, C. Lan, W. Zeng, Z. Chen, and S.-F. Chang, "Rethinking classification loss designs for person re-identification with a unified view," *ArXiv abs/2006.04991*, 2020.

[13] S. Zhou, J. Wang, J. Wang, Y. Gong, and N. Zheng, "Point to set similarity based deep feature learning for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3741–3750.

[14] M. Wieczorek, B. Rychalska, and J. Dabrowski, "On the unreasonable effectiveness of centroids in image retrieval," in *International Conference on Neural Information Processing*. Springer, 2021, pp. 212–223.

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 3–19.

[16] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5332–5340.

[17] 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 *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[20] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.

[22] J. Huang, R. Feris, Q. Chen, and S. Yan, "Cross-domain image retrieval with a dual attribute-aware ranking network," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1062–1070.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv*, vol. abs/1704.04861, 2017.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[26] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.