Clothing Retrieval and Visual Recommendation for Fashion Images

James Le

Deep Learning For Vision (Fall 2018)



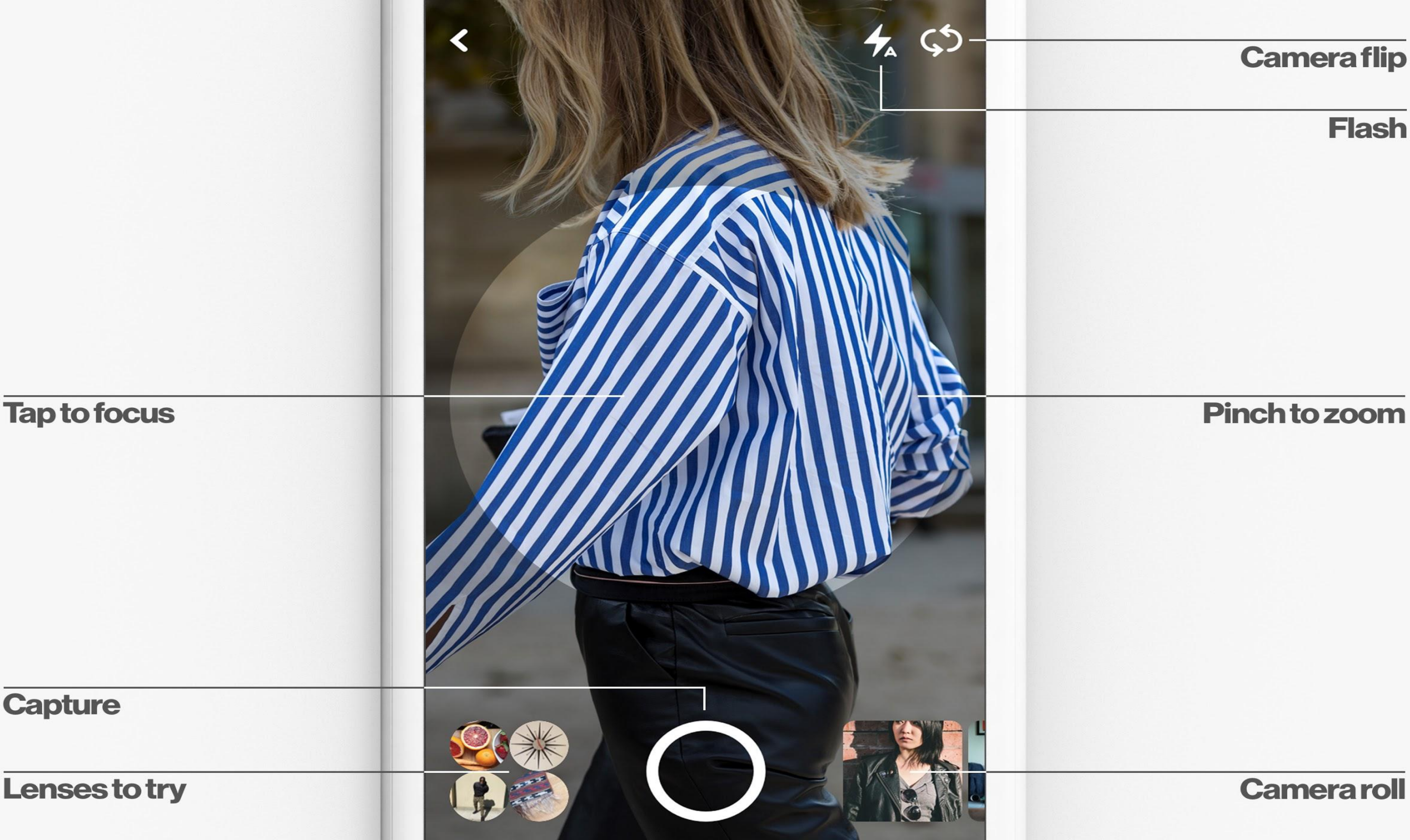



STITCH FIX





●●●○○ Verizon 2:48 PM 83%
RENT THE RUNWAY
GET RECOMMENDATIONS
WHAT'S NEW
IN LOVE WITH LACE
●●●○○ Verizon 2:49 PM 83%
8 OF 20
DONE
NOT FOR ME
SHORTLIST
BADGLEY MISCHKA
$85 Retail $495
●●●○○ Verizon 2:57 PM 81%
IN LOVE WITH LACE
FILTERS
CARMEN MARC VALVO
$100 Retail $725
ELIE TAHARI
$125 Retail $698
BAG
0
SHORTLISTS
MY ACCOUNT
DRESS MATCH
BROWSE
HOME
WHAT'S NEW
MOST POPULAR
DRESSES
ACCESSORIES
DESIGNERS


Camera flip
Flash
Tap to focus
Pinch to zoom
Capture
Lenses to try
Camera roll

Problem Formulation

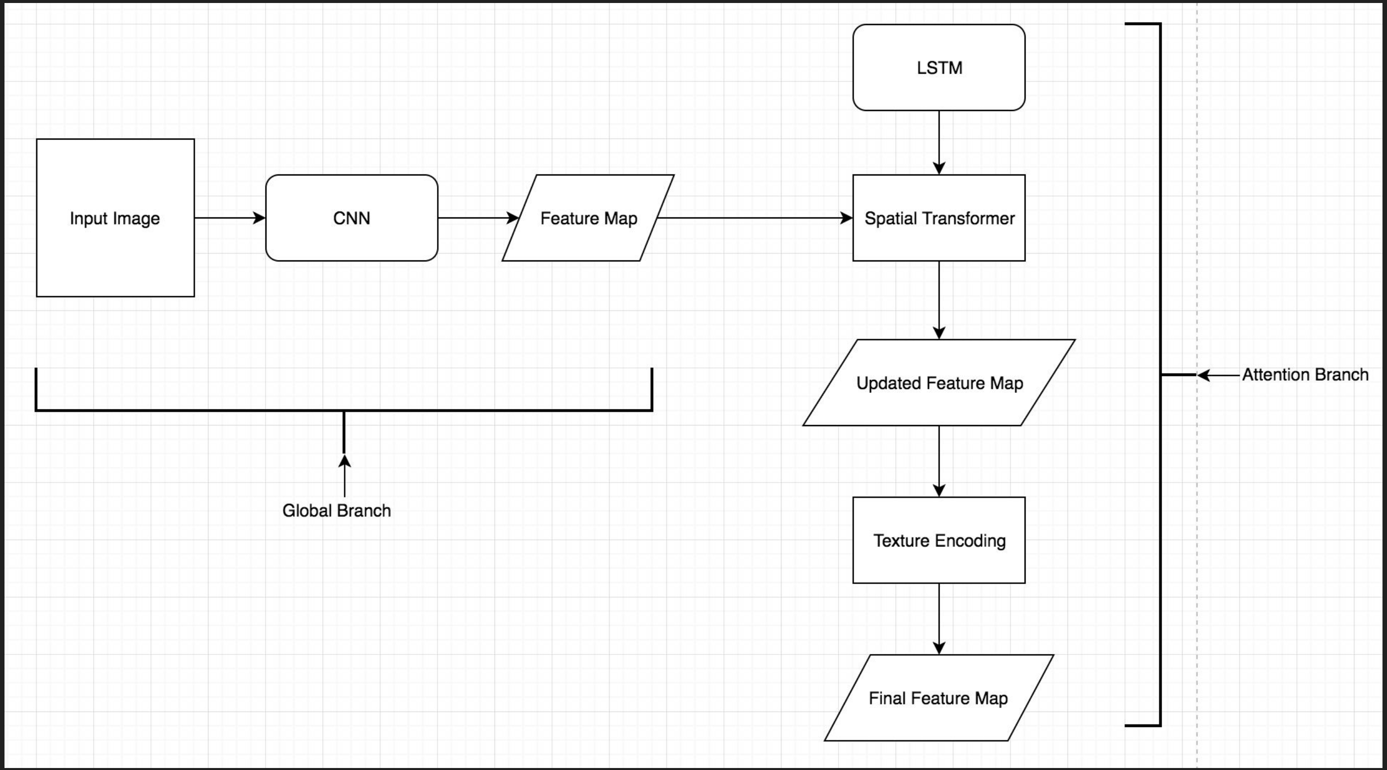
Main Contributions

- End-to-end solution for large-scale clothing retrieval and visual recommendation.
- Learn the important regions in an image.
- Generate diverse recommendations based on semantic similarity.
- Evaluate my method on in-shop retrieval task.

Literature Review

- Clothing Attribute Recognition
- Clothing Image Generation
- Clothing Item Retrieval
- Fashion Recommendation System

Proposed Method



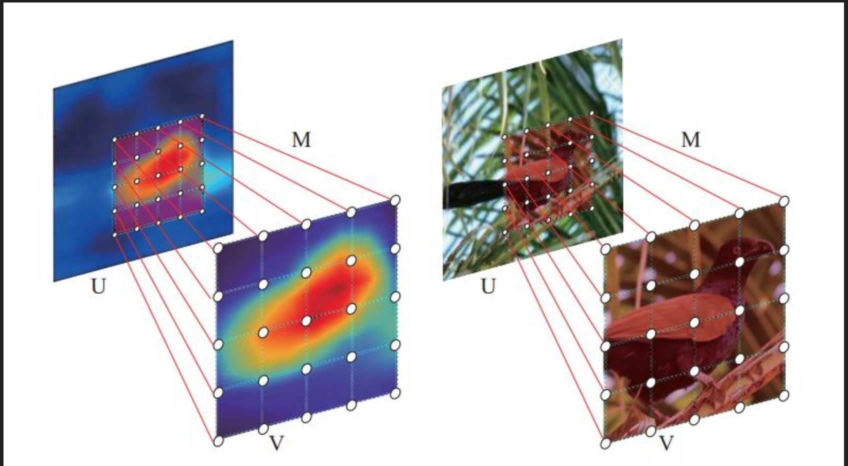
Global Branch

- All convolutional layers have 1 x 1 padding.
- All other layers have a 1 x 1 stride.
- All max pooling layers have a 4 x 4 stride.
- 3 x 3 kernels for the convolutional filters.
- Batch Normalization layers.
- Dropout layers.

type	kernel size	output size
convolution	3 x 3	384 x 256 x 64
convolution	3 x 3	384 x 256 x 64
dropout (25%)		384 x 256 x 64
max pooling	4 x 4	96 x 64 x 64
batch normalization		96 x 64 x 64
convolution	3 x 3	96 x 64 x 128
convolution	3 x 3	96 x 64 x 128
dropout (25%)		96 x 64 x 128
max pooling	4 x 4	24 x 16 x 128
batch normalization		24 x 16 x 128
convolution	3 x 3	24 x 16 x 256
convolution	3 x 3	24 x 16 x 256

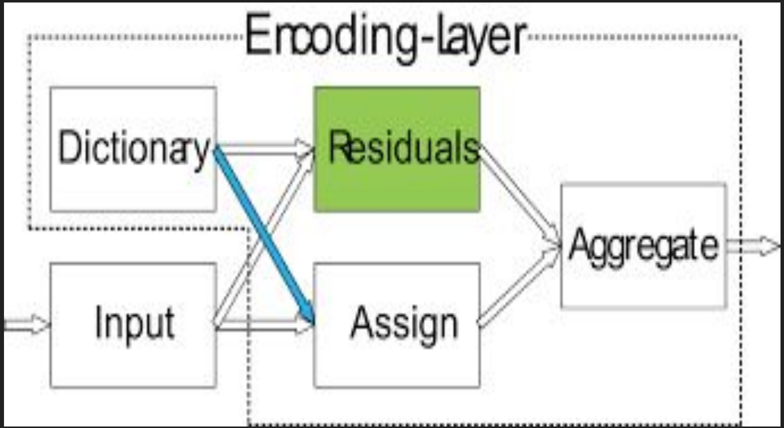
Attention Branch

Spatial Transformer Layer



(a) (b)

Texture Encoding Layer



k-Nearest Neighbor

Euclidean Distance

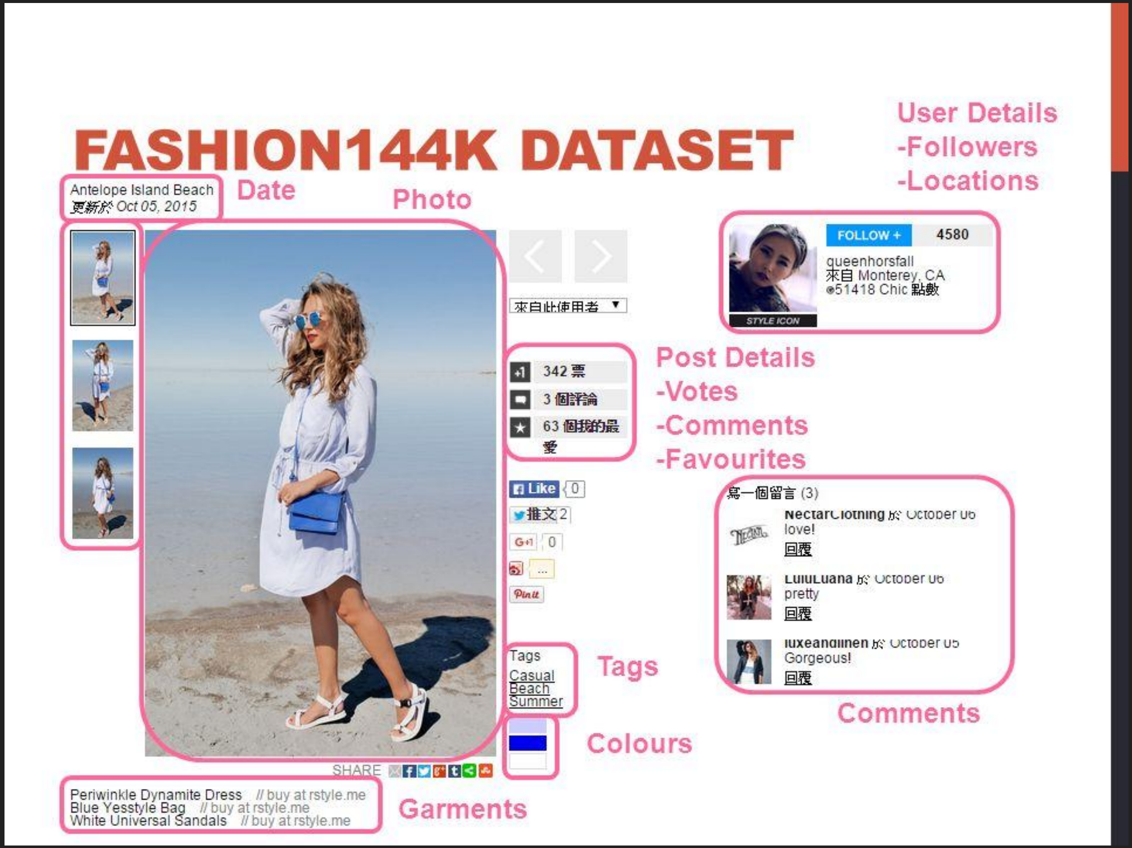
$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$$

Conditional Probability

$$P(y = j | X = a) = \frac{1}{K} \sum_{i \in A} I(y_i = j)$$

Fashion144k

- 90,000 training images.
- 128 classes.
- 384 x 256 image resolution.
- Multi-label annotations.
- Fashionability scores.



DeepFashion

- 800,000 images.
- Annotations about landmarks, categories, pairs etc.
- In-Shop Clothes Retrieval Benchmark:
  - 52,000 images.
  - 8,000 clothing items.

Experiments

- Trained the model on Fashion144k with 59 item labels, excluding color labels.
- Evaluated the model for in-shop retrieval task on DeepFashion.
- Experimental Setting:
    - PyTorch
    - Adam Optimizer
    - Batch Size 64
    - Learning Rate 0.00001
    - Momentum 0.9
    - 40 Epochs

Multi-Label Classification Loss

$$L_{cls} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}(p_i^c - \hat{p}_i^c)^2,$$

where, N is the number of training samples, C is the number of class labels, ^p_i is the ground-truth probability vector of the i-th sample, and p_i is the predicted label vector of that sample i.

Diversity Loss

$$L_{div} = \frac{1}{K-1}\sum_{k=2}^{K}\sum_{i=1}^{H\times W} l_{k-1}, i.l_{k,i},$$

where, K is the total steps of recurrent attention, H x W is the height and width of attention maps, l_k is the k-th attention map, and l_k,i is the i-th attention value of the attention map after conducting softmax on H x W locations at time step t.

Localization Loss

$$L_{loc} = L_S + \lambda_1 L_A + \lambda_2 L_P,$$

Anchor Constraint:

$$L_A = \frac{1}{2}\{(t_x^k - c_x^k)^2 + (t_y^k - c_y^k)^2\},$$

Scale Constraint:

$$L_S = L_{s_x} + L_{s_y},$$

Positive Constraint:

$$L_P = max(0, \beta - s_x) + max(0, \beta - s_y),$$

Combined Loss

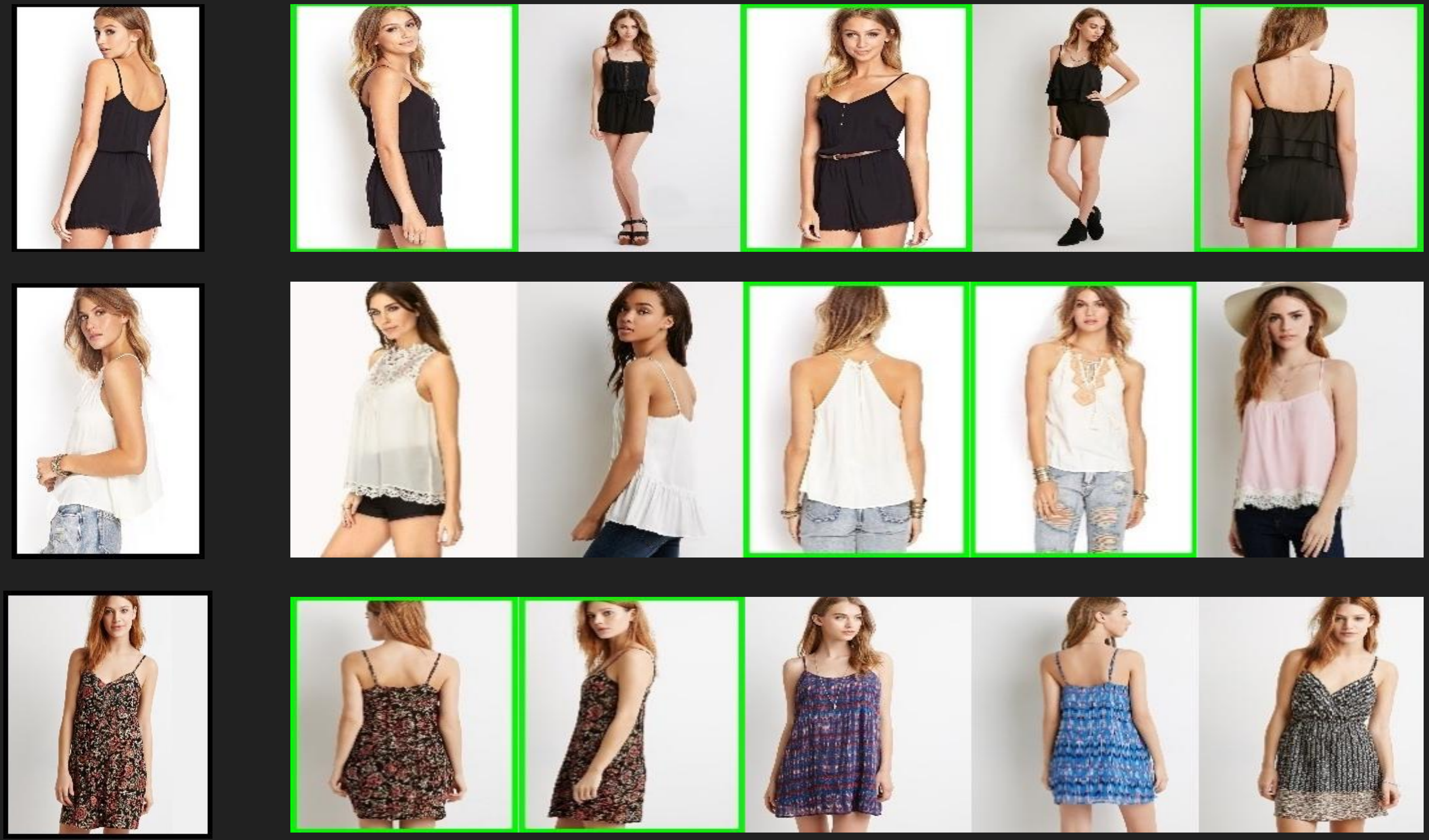
$$L = L_{cls} + \gamma_1 L_{div} + \gamma_2 L_{loc},$$

where gamma1 and gamma2 are the weighted parameters, and they

are set as 0.01 and 0.1, respectively.

Results

Method	Top-5	Top-10	Top-20	Top-30	Top-50
FashionNet	0.678	0.725	0.764	0.781	0.796
WTBI	0.425	0.470	0.506	0.514	0.540
DARN	0.548	0.624	0.675	0.701	0.719
VAM	0.836	0.887	0.923	0.936	0.947
Mine	**0.683**	**0.728**	**0.775**	**0.802**	**0.834**



Conclusion

- Using clothing parts for recommendation gives much variability in the recommendation results.
- Attention model can be used to learn discriminative features and semantic regions from the images.
- Texture-based features are important for learning different regions.

References

- Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. **Deepfashion: Powering robust clothes recognition and retrieval with rich annotations**. In CVPR, 2016.
- E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. **Neuroaesthetics in Fashion: Modeling the Perception of Fashionability**. In CVPR, 2015.
- Z. Wang, Y. Gu, Y. Zhang, J. Zhou, and X. Gu. **Clothing retrieval with visual attention model**. In IEEE, 2017.
- E. Simo-Serra and H. Ishikawa. **Fashion Style in 128 Floats: Joint Ranking and Classification using Weak Data for Feature Extraction**. In CVPR, 2016.
- Z. Wang, T. Chen, G. Li, R. Xu, and L. Xin. **Multi-label image recognition by recurrently discovering attentional regions**. In ICCV, 2017.
- M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. **Spatial transformer networks**. In NIPS, 2015.
- H. Zhang, J. Xue, and K. Dana. **Deep TEN: Texture encoding network**. In CVPR, 2017.
- B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan. **Diversified Visual Attention Networks for Fine-Grained Object Classification**. In IEEE Transactions on Multimedia, 2017.
- S. Verma, S. Anand, C. Arora, and A. Rai. **Diversity in fashion recommendation using semantic parsin**g. In IEEE, 2018.