International Core Journal of Engineering Volume 7 Issue 11, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

# Clothing Attribute Prediction with Projective Transformation

Ziyuan Qin[1,*], Kaipeng Wang[2,a], Lewen Sun[3,b], Yibo Ma[4,c], Shenyou Fan[5,d] and Zhijie Xia[6,e]

[1] Beijing University of Posts and Telecommunications, Beijing, China

[2] Henan University of Technology, Zhengzhou, Henan, China

[3] Wenzhou-Kean University, Wenzhou, Zhejiang, China

[4] University of Shanghai for Science and Technology, Shanghai, China

[5] Sichuan University, Chengdu, Sichuan, China

[6] University of California, Santa Barbara, Santa Barbara, CA, USA

*Corresponding author: ziyuanqin2020@163.com, [a] 1592622761@qq.com,

[b] 377188987@qq.com, [c] 15657563008@163.com, [d] fsy1633213927@163.com, [e] xiazj07@163.com

These authors have contributed equally to this work

## Abstract

**In this article, we introduce self-supervised learning based on the supervised learning model, which can help predict the attributes of clothing with the help of the self-information of clothing images. Specifically, we use the female clothing image dataset and the male clothing image dataset. We transform the image and feed the images before and after the transformation to the same CNN to obtain two sets of features. Then, the two sets of features are fed to the transform decoder module and the transform coefficients are solved. The coefficient is used as a label to enhance the ability of supervised learning. Finally, we use the enhanced CNN to learn the features of the original image and connect these features to the clothing attribute classifier. The experiment proves the advantage of this method on two clothing data sets.**

## Keywords

**Projective Transformation; CNN; Feature recognition.**

## 1. Introduction

The clothing attribute recognition is a fundamental task, and it can be used in many practical fields. For example, for e-commerce platforms, accurate clothing retrieval can facilitate and attract customers. At the same time, in the task of person re-identification (person Re-ID), the identification of clothes can assist in quickly finding the person we are looking for. In addition, there are also many researches on clothing attribute detection and recognition [1-4].

At present, there are mainly two solutions in the research on the problem of image attribute recognition. The first is the traditional method, which manually sets the features according to the image characteristics. For the task of clothing image attribute recognition, the features usually use SIFT, HOG, skin feature, and so on. The second method is based on deep learning, the most important is to use convolutional neural networks. Compared with traditional methods, features are based on learning, not artificial settings. At present, for the task of image attribute recognition, a lot of works

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

are conducted to improve the performance through exploring the convolutional neural network (CNN) model. The shortcoming of these studies is that they ignore the self-information of the image and explore too little information about the image itself. The motivation lies that the CNN model needs more supervision information to train a robust model, however, the annotations are time-consuming to obtain. Thus, introducing unsupervised self-information from images themselves is a good solution to help improving the performance of clothing attribute recognition.

In order to solve the above problems, we introduce self-supervised learning. Self-supervised learning is a type of unsupervised learning. Self-supervised learning uses the data information provided by the data itself as self-supervised label guiding learning. Our proposed model is shown in Figure 1. First, there are two input images, one is the normal image, the other is the transformed image. We perform projective transformation on the normal image to get the new transformed image. Secondly, the two images are fed into a shared CNN for feature extraction, and two sets of features are extracted. Then, the obtained convolutional feature and the transformed convolutional feature are fed into a transform decoder module. We can solve the transform coefficients through the decoder. In this step, we did not introduce new manual annotations, but excavated the image's own information, i.e., transform coefficients, as labels to enhance the ability of supervised learning. Finally, we use CNN enhanced by unsupervised learning to learn the features of the original image. And these features are connected to the classifier for clothing attribute classification. After our verification, the effect of introducing unsupervised learning is significantly enhanced. The experimental results prove the effectiveness of introducing an unsupervised training model for image attribute recognition tasks.

The advantage of introducing unsupervised learning is that it does not need to introduce new annotations, it comes with self-supervised information. At the same time, unsupervised learning has good feature learning capabilities. Feature recognition is very important in our clothing image attribute recognition task. Thus, our main purpose of introducing unsupervised learning is to improve the feature representation ability of supervised learning, so as to help complete the task of identifying attributes of clothing images.

In the following chapters, we will specifically introduce the related work of clothing attribute recognition, our experimental methods, and our verification results.

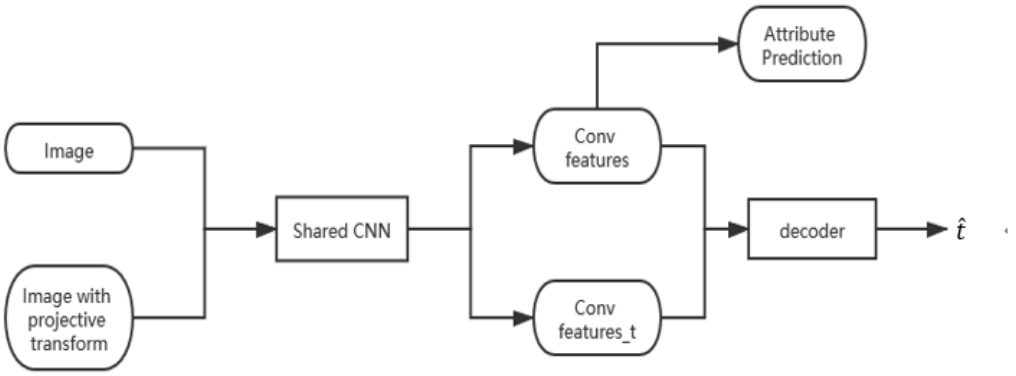


Figure 1. Projective Transform enhanced Clothing Attribute Network (PTCAN)

## 2. Related Works

Here we briefly review mainly related topics of clothing attributes and unsupervised learning.

### 2.1 Clothing attributes.

Lubomir Bouedev et al. [1] proposed to decompose the image into a set of parts, poselets, and used a feed-forward network for person attribute prediction. Zheng Song et al. [2] proposed to recognize human occupation via two properties of human-related images, human clothing and scene contexts. To further obtain the abstract occupation information, they proposed to refine features of human-

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

related images based on part based human alignment as well as the modeling of semantically meaningful patterns. Huizhong Chen et al. [3] presented a fully automatic system and created a dataset with annotated images that contain clothed people in unconstrained settings to evaluate the descriptive ability of system. Si Liu et al. [4] collected a large online shopping dataset and daily photo dataset for investigating the cross-scenario clothing retrieval task. What's more they proposed a two-step calculation to handle the cross-scenario discrepancies. Qiang Chen et al. [5] used deep learning approaches, considered attribute sub-categories that differ in subtle details, including many types of clothing color. Junshi Huang et al. [6] They created a dataset containing tens of thousands of online-offline clothing image pairs obtained from the user review pages. They proposed the Dual Attribute-Aware Ranking Network which simultaneously integrates the attribute learning and visual similarity constraint into the retrieval feature learning. Ziwei Liu et al. [7] fused rich attribute supervision, landmark prediction, and triplet loss incorporating pair correspondences into their full model FashionNet, respectively. They also conducted additional experiments on the in-shop clothes retrieval benchmark. Qi Dong et al. [8] formulated a novel Multi-Task Curriculum Transfer (MTCT) deep learning approach to model clothing attributes. Additionally, they implemented a novel curriculum transfer deep learning strategy that aims to explore target-domain knowledge from the source-domain attribute information.

### 2.2 Unsupervised learning.

Xiaolong Wang and Abhinac Gupta [9] designed a Siamese-triplet network with ranking loss function to train the CNN representation. Their key idea is that two patches connected by a video tracker should have similar visual representation in deep feature space since they belong to the same object. Carl Doersch et al. [10] presented a ConvNet-based approach to learn a visual representation from this task, which aims to provide a similar "self-supervised" formulation for image data: a supervised task involving predicting the context for a patch. They demonstrate that the resulting visual representation is good for both object detection, providing a significant boost on PASCAL VOC 2007 compared to learning from scratch, as well as for unsupervised object discovery / visual data mining. Deepak Pathak et al. [11] trained a convolutional neural network to regress to the missing pixel values. The model is called context encoder, it consists of an encoder capturing the context of an image with a compact latent feature representation and a decoder which uses that representation to produce the missing image content. Mehdi Noroozi and Paolo Favaro [12] proposed that solving Jigsaw puzzles can be used to teach a system that an object is made of parts and what these parts are. After the training, the features are highly transferrable to detection and classification. Spyros Gidaris et al. [13] followed the self-supervised paradigm and proposed to learn image representations by training ConvNets to recognize the geometric transformation that is applied to the input image. They defined a small set of discrete geometric transformations, then each of those geometric transformations are applied to each image on the dataset and the produced transformed images are fed to the ConvNet model that is trained to recognize the transformation of each image. Liheng Zhang et al. [14] proposed to learn unsupervised feature representations by Auto-Encoding Transformations (AET), which is focused on learning transformation coefficients rather than reconstructing the input data. They seek to train auto-encoders that can directly reconstruct the transformation coefficients from the learned feature representations between original and transformed images.

## 3. Methodology

### 3.1 Affine vs. Projective Transformation

Theorem: A mapping $h$: $P^2 \rightarrow P^2$ is a projectivity if and only if there exists a non-singular $3 \times 3$ matrix $H$ such that any point in $P^2$ represented by a vector $x$ is valid with the equation $x' = Hx$. Affine transformation has 6 degrees of freedom: 2 for scale, 2 for rotation, and 2 for translation. Affine transformation retains parallelism, ratios of parallel lengths, and ratios of areas, while full projective transformation (homography) has two more degrees of freedom for the line at infinity, which is 8 in total.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

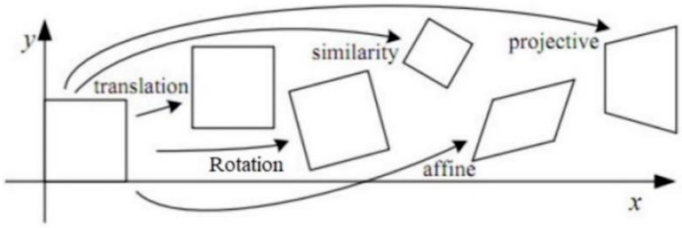


Figure 2. Comparisons among transformations

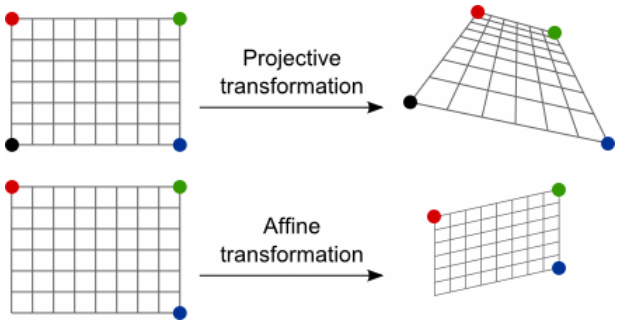


Figure 3. Full projective transformation vs. affine transformation

### 3.2 Image rectification and feature detection

Given the predefined formula x' = Hx, which is a linear mapping of homogeneous coordinates. The projective transformation technique is typically used to map the first image on the second one. This point-to-point mapping preserves lines: line in one plane is mapped to line in another.

$$\begin{matrix} x' \\ y' \\ 1 \end{matrix} = \begin{matrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & 1 \end{matrix} \cdot \begin{matrix} x \\ y \\ 1 \end{matrix}, \text{ in short form, } x' = \boldsymbol{H_p} x = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v^T} & \mathbf{1} \end{bmatrix} x.$$

Here, the $\mathbf{t}$ stands for translation vector, $\mathbf{A}$ stands for scaling and rotation vectors, and $\mathbf{v^T}$ is known as line at infinity.

First of all, we want to input a pair of images with common overlapping regions, and we wish to receive an output of a pair of resampled images such that the epipolar lines in two images are horizontal (parallel with the x-axis), and as a result of that the corresponding points in the two images are as close to each other as possible.

1. Identify a seed set of image-to-image matches $x_i \leftrightarrow x_i'$ between the two images. A minimum of seven points is needed, preferable with more corresponding pairs. It is possible to find such matches by automatic means.

2. Compute the fundamental matrix F and find the epipoles $e$ and $e'$ in the two images.

3. Select a projective transformation $H'$ that maps the epipole $e'$ to the infinity point $(1,0,0)^T$.

4. Find the matching projective transformation $H$ that minimizes the least-squares distance:

$$\sum_i d(Hx_i, H'x_i')$$

Efficient methods exist to compute this robustly.

Resample the first image according to the projective transformation $H$ and the second image according to the projective transformation $H'$.

While computing homography and fundamental matrix, a ubiquitous problem is known as the correspondence problem, which is to identify pixel locations in two or more images that belong to the same scene point. There are several requirements constituted on an excellent feature point: uniqueness, computability, and similarity. Corners are generally good points: they remain stable over

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

different viewpoints, are points with significant curvature changes, are points correspond to junctions of three or more planes in 3D scenes. An image patch noted as $I(u,v)$ together with a window of pixels $\{(u,v)\}$ is defined, shifting this patch by $(x,y)$. The difference between the shifted and the original image formulates the equation:

$$E(x,y) = \sum_{u,v} (I(u+x, v+y) - I(u,v))^2$$

Substituting the above equation using approximation:

$$E(x,y) \approx \sum_{u,v} w(u,v)\big(I_x(u,v)x + I_y(u,v)y\big)^2$$

The local structure matrix is defined to be $A$:

$$A = \sum_{u,v} w(u,v)\begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}$$

For a 2*2 matrix $A$, we can consider the following as the Corner Response Function (CRF):

$$\det(A) - k(trace(A))^2$$

Where $k$ is the sensitivity of the detector, and we prefer small values of $k$, typically from 0.04 to 0.06, to be considered as highly sensitive.

The above corner measure is close to zero for nearly flat regions and provides larger values for "corner" type regions.

### 3.3 CNN convolution network

Given image input $u(m,n)$ the output is $v(m,n) = f\big(u(m,n)\big), where\ v(m,n) = \sum_k \sum_l u(k,l)h(m-k, n-l)$, and $h(m-k, n-l)$ is the convolution kernel. In many CNN discussions, the "valid" convolution refers to all filter locations where the filter is completely inside the image input. Though this terminology is different from standard signal processing, we will use "valid" convolutions to refer to the case where the filter kernel fully contains the input data (image). Moreover, zero-padding is used to extend the signal dimensions to result in the output signal with the desired dimensions. The convolutional filters will also have to work with the 3rd dimension corresponding to the depth (channels), and typical convolutional layers are followed by "pooling layers" whose main goal is to extract the useful responses in the feature maps. Given the input is $W_1 \times H_1 \times C$, and the number of filters $K$, filter size $F$, stride $S$, zero padding $P$, this will generate an output of $W_2 \times H_2 \times C$, where $W_2 = \frac{(W_1 - F + 2P)}{S} + 1, H_2 = \frac{(H_1 - F + 2P)}{S} + 1$. The total number of parameters will be $F^2 * C * K$ with $K$ biases. The entire design has 5 layers in total and is designed as follows:

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | [3×3, 64; 3×3, 64] ×2 | [3×3, 64; 3×3, 64] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 | [1×1, 64; 3×3, 64; 1×1, 256] ×3 |
| conv3_x | 28×28 | [3×3, 128; 3×3, 128] ×2 | [3×3, 128; 3×3, 128] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×4 | [1×1, 128; 3×3, 128; 1×1, 512] ×8 |
| conv4_x | 14×14 | [3×3, 256; 3×3, 256] ×2 | [3×3, 256; 3×3, 256] ×6 | [1×1, 256; 3×3, 256; 1×1, 1024] ×6 | [1×1, 256; 3×3, 256; 1×1, 1024] ×23 | [1×1, 256; 3×3, 256; 1×1, 1024] ×36 |
| conv5_x | 7×7 | [3×3, 512; 3×3, 512] ×2 | [3×3, 512; 3×3, 512] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 | [1×1, 512; 3×3, 512; 1×1, 2048] ×3 |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

Figure 4. Construction and design of NN layers

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

The usage of first layer is to visualize filters. The second layer responds to the intersection of corners and other edges/colors. The third layer has more complex immutability and can capture similar textures (e.g. grid patterns, text). The fourth layer shows obvious changes and is category-specific, such as dog's face, bird's legs. The fifth layer shows that the posture of the entire object changes greatly (e.g. keyboard and the dog).

The multiclass SVM (Support Vector Machine) loss is defined using the shorthand for the scores vector $s = f(x_i, W)$.

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Figure 5. SVM loss

# 4. Experiment

## 4.1 Dataset and processing

Two datasets are used, one is the woman clothing image dataset and man clothing image dataset. Both these two datasets own 13,500. To make recognition more efficiently, we positioned the clothing with a bounding box that contains four-dimensional coordinates, including the position of the upper left point and the lower right point. For the clothing attribute prediction task, we use 10,200 images for training and 3300 images for testing. A total of 10 attributes with different values are predicted, including color, style, collar, color style, sleeve style, sleeve length, zipper, belt, button, and overall length.

Table 1. Clothing attribute values in woman and man clothing

| Attribute | | Attribute Values |
|---|---|---|
| Color | *Woman and man clothing* | Red, orange, yellow, green, blue, purple, black, white, gray, brown, multi-color |
| Collar | | V-shape, round, pile, collar, turndown collar/POLO, stand collar, irregular |
| Style of color | | Pure color, round dot, cell, irregular |
| Length of sleeve | | Sleeveless, short, long |
| Style | | Skinny, straight, loose, irregular |
| Style of sleeve | | Normal sleeve, puff sleeve, shirt sleeve, pile sleeve, irregular |
| Zip | | Exist, without |
| Belt | | Exist, without |
| Button | | Exist, without |
| Length of whole | | median, short, long |
| 10 attribute | | **color, style, collar, style of color, style of sleeve, length of sleeve, zip, belt, button, length of whole** |

## 4.2 Baseline

We choose the AlexNet backbone to conduct the baseline. We set the base learning rate as 0.001 and then reduce the learning rate by one-tenth every 40 epochs. Furthermore, we adjusted all input images to the form of 224*224*3. After that, we got the loss of our classification and calculated the precision for each attribute of our work. We get precision, average precision, and mean average precision and use them to evaluate our work.

For precision, we use the formula $P = \frac{tp}{tp+fp}$ (Where P is precision, $tp$ denotes the number of correct predictions, $fp$ denotes the number of mispredictions)

For average precision, we use $AP = \frac{1}{T}\sum_{t=1}^{T} P_t$ (where T is the set of all attribute tasks, $P_t$ denotes the t-th precision)

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

For mean average precision, we use $mAP = \frac{1}{|L_t|} \sum_{i=1}^{|L_t|} P(y_{t,i}, \hat{y}_{t,i})$ (where $L_t$ is the attribute value set of t-th attribute, P denotes the precision score, $y_{t,i}$ denotes the samples predicted as i-th class of t-th attribute, $\hat{y}_{t,i}$ denotes the labels corresponding to $y_{t,i}$)

We did the next task using unsupervised learning to optimize the pre-set parameters we used in classification. We have already got the image x and transformed image $\hat{x}$, so we input the original image and the transformed image into our proposed model to calculate the $\hat{t}$. The way we did this can be display as the following calculation.

$$[x' \quad y' \quad w'] = [x \quad y \quad 1] \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

Afterwards, we used $\hat{t}$ to compare with t in order to analyze the precision of the neural network, improving its learning hyperparameters.

To improve the whole work's effectiveness, we did not create the transform coefficient one by one but have 500000 transformations. While in training stage, we randomly choose one transformation implemented on the input image, the transformation coefficients will act as the label of the unsupervised learning branch.

## 5. Achievement

The total loss of our framework is

$$L_{total} = L_{att} + \alpha L_{AET},$$

Where $L_{att}$ is the sum loss of all clothing attribute tasks, each clothing attribute task adopts the standard cross entropy loss. $L_{AET}$ is the mean square error loss. The parameter $\alpha$ is set as 2.

Table 2. The comparison results of AlexNet baseline and adding AET part.

| Method | Color | Style | Collar | Styleof color | Styleof sleeve | Lengthof sleeve | Zip | Belt | Button | Lengthof whole | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline_ Alexnet | 73.96 | 50.81 | 53.54 | **82.58** | 45.09 | **92.43** | 86.67 | 78.22 | 88.86 | 71.11 | 72.33 |
| AlexNet_AET ($\alpha = 1$) | **74.47** | 52.71 | 53.86 | 76.86 | **51.58** | 92.25 | 87.17 | **84.65** | 88.96 | 69.79 | 73.23 |
| AlexNet_AET ($\alpha = 2$) | 74.29 | **53.23** | **55.17** | 81.17 | 50.89 | 91.93 | **88.73** | 81.66 | **89.68** | **72.52** | **73.93** |

As shown in Table 1, we can find that the methods AlexNet_AET obtain higher mAP score than the baseline AlexNet. While the method AlexNet_AET with different value α, the AlexNet_AET(α=2) achieves 0.7% improvements than AlexNet_AET (α=1) in terms of mAP score. Thus, we choose α=2 in the following for the reason of good performance.

Table 3. The comparison results of VGG16 baseline and adding AET part.

| Method | Color | Style | Collar | Styleof color | Styleof sleeve | Lengthof sleeve | Zip | Belt | Button | Lengthof whole | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline_ VGG16 | **74.33** | 54.84 | 57.78 | 81.47 | 43.84 | 95.33 | **92.55** | **85.77** | 92.05 | 71.39 | 74.94 |
| VGG16_AET ($\alpha = 2$) | 73.85 | **56.09** | **61.17** | **82.92** | **50.16** | **95.85** | 91.85 | 81.36 | **92.74** | **74.24** | **76.02** |

In addition, we also test the influence on other popular CNN backbone VGG16, the results are shown in Table 2. The method VGG16_AET obtains 1.08% improvements than the baseline with VGG16 in terms of mAP metric. And the performance of VGG16_AET is also better than the AlexNet_AET on account of better CNN backbone.

In Table 3, we show the comparison results with current state-of-the-art methods in terms of AP and mAP scores. The proposed VGG16_AET can achieve the best mAP score than other state-of-the-art

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

methods, and it can obtain the best AP scores on most clothing attribute tasks. The best performance shows the proposed unsupervised AET part can give us assistance for the clothing attribute recognition task.

Table 4. The comparison results on the testing set of woman clothing dataset with state-of-the-art methods.

| Method | Color | Style | Collar | Styleof color | Styleof sleeve | Lengthof sleeve | Zip | Belt | Button | Lengthof whole | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DDAN [1] | 67.07 | 49.97 | 50.99 | 73.69 | 40.00 | 88.44 | 82.34 | 76.88 | 86.29 | 66.87 | 68.25 |
| DARN [2] | 60.54 | 45.38 | 39.76 | 55.18 | 40.4 | 86.92 | 64.47 | 67.17 | 80.40 | 64.88 | 60.51 |
| FashionNet [3] | 61.00 | 55.81 | 54.35 | 70.70 | 37.09 | 89.16 | 81.41 | 83.33 | 88.92 | 71.33 | 69.31 |
| MTN[4] | 59.01 | **61.25** | 45.95 | 67.32 | 47.13 | 89.06 | 86.59 | 71.48 | 85.06 | 73.38 | 68.62 |
| VGG16_ TAN[5] | 65.74 | 58.34 | 60.06 | 75.05 | 52.31 | 93.1 | 88.45 | **85.46** | 90.40 | 74.17 | 74.33 |
| VGG16_AET ($\alpha = 2$) | **73.85** | 56.09 | **61.17** | **82.92** | **50.16** | **95.85** | **91.85** | 81.36 | **92.74** | **74.24** | **76.02** |

The qualitative results of the proposed VGG16_AET are shown in Figure 6. The method VGG16_AET can predict clothing attribute values well.

blue
skinny
V-shape
irregular
normal sleeve
long
without zip
without belt
exist button
median

Gray (Brown)
skinny
round
pure color
normal sleeve
long
without zip
without belt
without button
median

red
skinny
turndown collar/POLO
cell
normal sleeve
short
without zip
without belt
exist button
median

red
loose
round (irregular)
irregular
normal sleeve
long
without zip
without belt
without button
median

Figure 6. The clothing attribute recognition results of the proposed VGG16_AET method. The values with red color are the wrong predictions. From top to down are the results of color, style, collar, style of color, style of sleeve, length of sleeve, zip, belt, button, length of whole.

## 6. Conclusion

In this article, we introduce unsupervised learning on the basis of supervised learning to enhance the feature extraction ability of supervised learning. After transforming the image, the features are extracted and the decoder is used to calculate the coefficient of change, and the coefficient is used as a label to enhance the ability of supervised learning. This way we can better use the information of the picture itself. Through experimental results, we prove that the introduction of unsupervised learning can better recognize clothing images.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0077

## References

[1] Lubomir Bourdev, Subhransu Maji and Jitendra Malik. Describing People: A Poselet-Based Approach to Attribute Classification. In ICCV, 2011.

[2] Zheng Song, Meng Wang, Xian-sheng Hua and Shuicheng Yan. Predicting Occupation via Human Clothing and Contexts. In ICCV, 2011.

[3] Huizhong Chen, Andrew Gallagher and Bernd Girod. Describing Clothing by Semantic Attributes. In ECCV, 2012.

[4] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu and Shuicheng Yan. Street-to-Shop: Cross-Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set. In CVPR, 2012.

[5] Qiang Chen, Junshi Huang, Rogerio Feris, Lisa M Brown, Jian Dong and Shuicheng Yan. Deep Domain Adaptation for Describing People Based on Fine-Grained Clothing Attributes. In CPVR, 2015.

[6] Junshi Huang, Rogerio Feris, Qiang Chen and Shuicheng Yan. Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network. In ICCV, 2015.

[7] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang and Xiaoou Tang. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In CPVR, 2016.

[8] Qi Dong, Shaogang Gong and Xiatian Zhu. Multi-Task Curriculum Transfer Deep Learning of Clothing Attributes. In WACV, 2017.

[9] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations using Videos. In ICCV, 2015.

[10] Carl Doersch, Abhinav Gupta and Alexei A. Efros. Unsupervised Visual Representation Learning by Context Prediction. In ICCV, 2015.

[11] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell and Alexei A. Efros. Context Encoders: Feature Learning by Inpainting. In CPVR, 2016.

[12] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In ECCV, 2016.

[13] Spyros Gidaris, Praveer Singh and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In arXiv: 1803. 07728[cs.CV].

[14] Liheng Zhang , Guo-Jun Qi, Liqiang Wang and Jiebo Luo. AET vs. AED: Unsupervised Representation Learning by Auto-Encoding Transformations rather than Data. In arXiv:1901. 04596v1 [cs.CV].