

## #1. Aesthetic-based Clothing Recommendation

By Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, Zheng Qin

① SIFT(scale-invariant feature transform algorithm) feature is an algorithm that extracts features that are invariant to the size and rotation of an image. The basic principle is that by extracting SIFT features from two different images and matching features that are most similar to each other, the corresponding part of the two images can be found.



### [Operation Principle of SIFT]

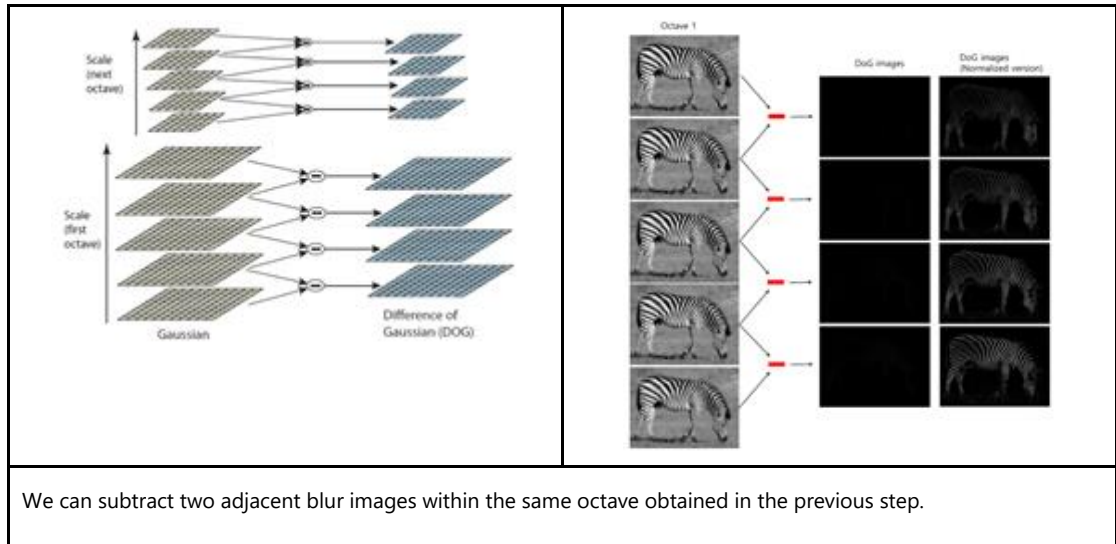
- Making "Scale space": In this blog, the original image is doubled, reduced in half, and then progressively blurred via DoF. The Scale space has 16 images including blur processing.

→ Scale space: Collect images of multiple scales(Large scale/Small scale).



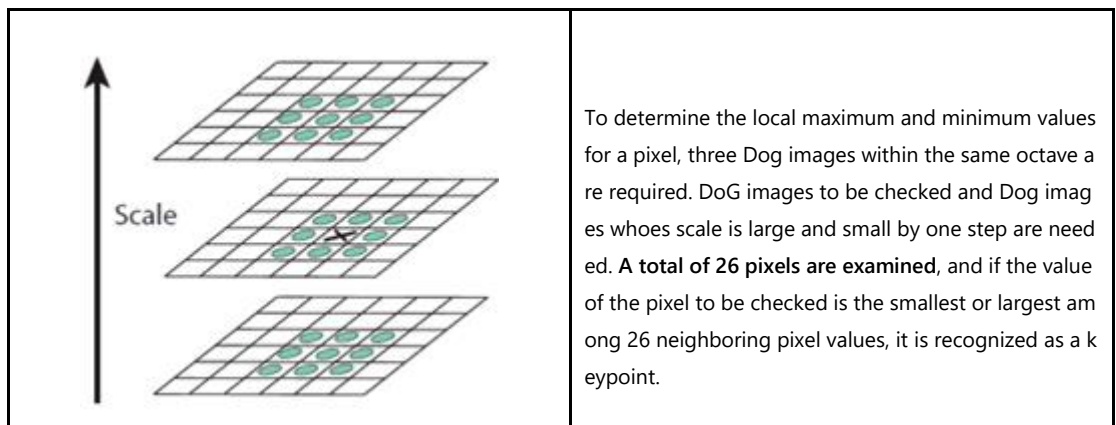
- Calculation of DoG(Difference of Gaussian)

: By blurring the image using Log(Laplacian of Gaussian) and then performing a second-order differentiation, the edges and corners in the image stand out. These edges and corners are useful to find keypoints. However, since LoG requires a lot of computation, the DoG, which is relatively hard and has similar performance, is used in this blog.



+) **The local maximum and minimum values** of the scale-normalized log represent image features very stably, and these maximum and minimum values are candidates for key points.

- Find keypoints

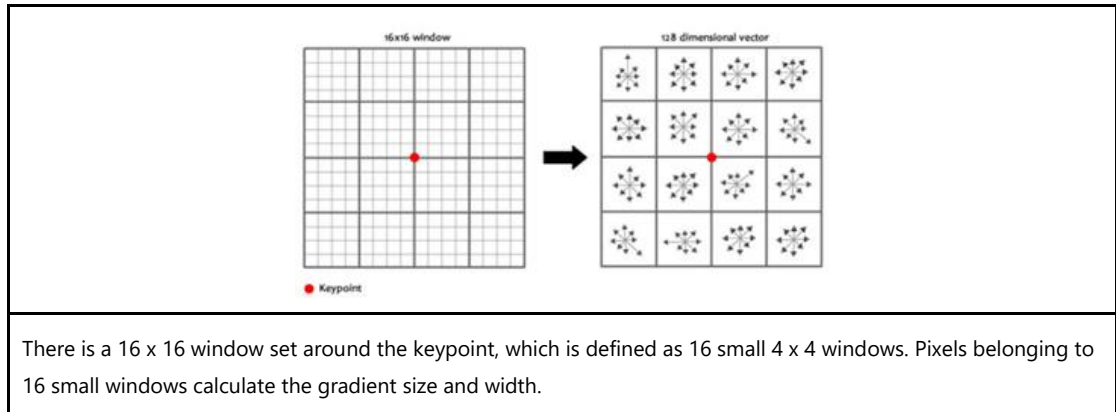


+) Remove bad key points: Removes those with low contrast or that are on the edge.

- Assign direction for keypoints

: When bad keypoints are removed and a suitable keypoint is found, these key points satisfy the scale invariance. Now, by collecting the gradient direction and size around each keypoint and assigning the direction to the keypoints, we make it have rotation invariance.

- Compute SIFT features finally  
: The characteristics of each keypoint are expressed as 128 numbers, and for this the trend of shape change around the keypoint is identified.



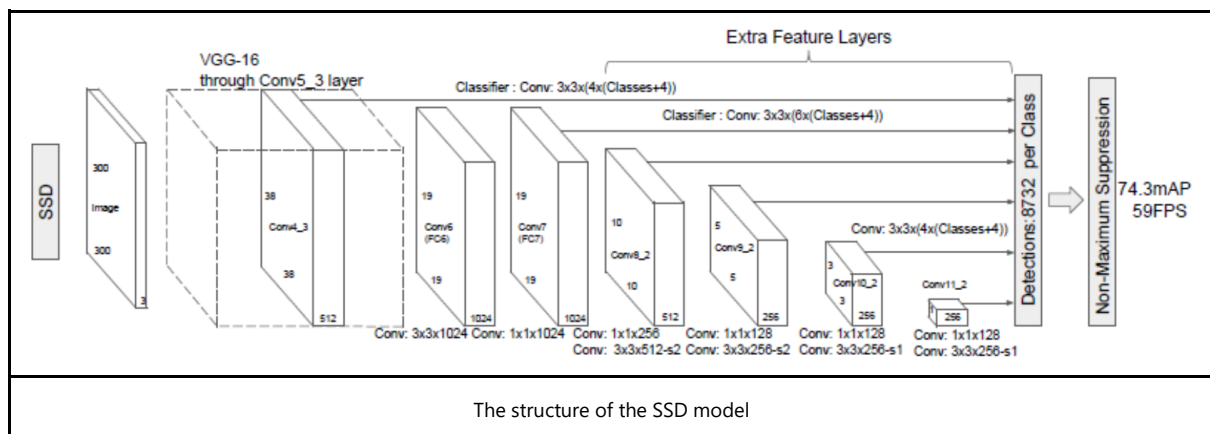
<https://bskyvision.com/21>

// This reference explain only keypoint detail and easily based on practice

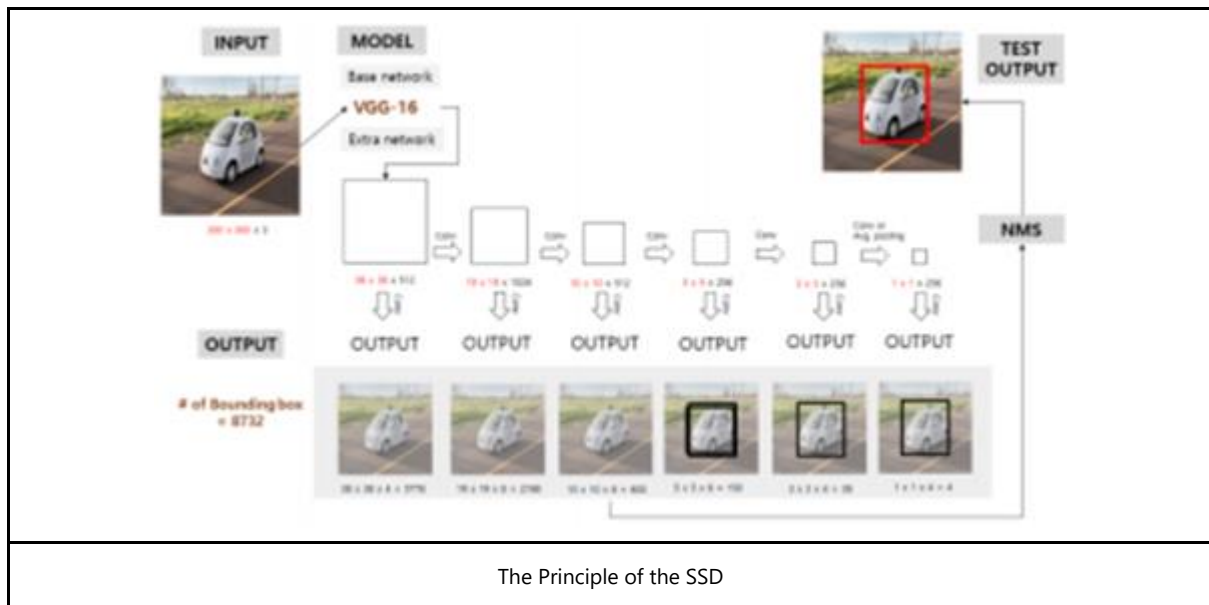
## #2. Deep Learning for Fast and Accurate Fashion Item Detection

By Evgeny Smirnov, Anton Kulinkin, Karina Ivanova and Michael Pogrebnyak

① MultiBox → SSD(Single Shot Multibox Detector, 2016) discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location(multiple feature map)



[https://pytorch.org/hub/nvidia\\_deeplearningexamples\\_ssd/](https://pytorch.org/hub/nvidia_deeplearningexamples_ssd/)



- First, when processing the input image basically, it takes the VGG-16 model and applies up to Conv4-3 (3 4 x 4 Convs) to the base network.
- The 38 x 38, 19 x 19, 10 x 10, 3 x 3, and 1 x 1 multi-feature maps highlighted in the original text are **feature maps** that are directly connected to the output.
- In each feature map, we get the class score and offset of the bounding box we want to predict with the appropriate transform operation.  
→ conv filter size = 3 x 3 x #bounding box x (class score + offset), stride = 1, padding = 1
- The output of 8732 bounding boxes comes out, but after calculating the IOU of the default bounding box by applying a different cross section for each feature map, only those boxes with a **value of 0.5 or more in advance are included as 1**, but the rest are considered as 0.

+) Ground Truth Box: The correct box we should predict

+) NMS(Non-Max-Suppression): A technique to process data by removing or ignoring relatively insignificant points or data based on certain criteria. It is widely used in neural networks related to object recognition.

★★★ <https://taeu.github.io/paper/deeplearning-paper-ssd/>

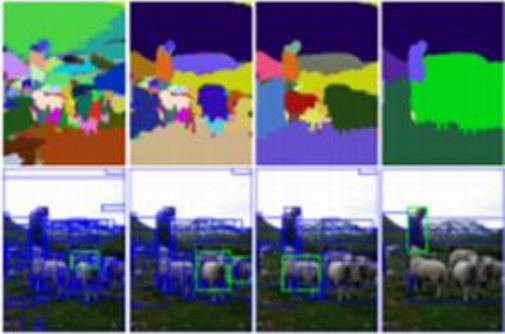
(Original Paper) <https://arxiv.org/pdf/1512.02325.pdf>

<https://m.blog.naver.com/sogangori/221007697796>

<https://www.slideshare.net/HyunKyuJeon3/15ssdsingle-shot-multibox-detector-124450000>

② Selective Search(2012) uses a hierarchical grouping algorithm to search based on super pixels(patch that expresses the edge well) that finds bonding boxes. It recommends candidate regions by combinin

g the two methods, segmentation to guide the sampling process using image structure and Exhaustive Search to locate all objects.

	
<p>It finds many objects at different scales.</p>	<p>It necessarily finds the objects at different scales as the girl is contained by the tv.</p>

<https://mainpower4309.tistory.com/27>

<https://go-hard.tistory.com/33>

<https://donghwa-kim.github.io/SelectiveSearch.html>

<https://m.blog.naver.com/laonple/220918802749>

(Original Paper) <http://www.huppelen.nl/publications/selectiveSearchDraft.pdf>

### ③ How the Backpropagation is calculated

: Understanding backpropagation could be helpful to fine-tune models and identify the cause of errors.

<https://gomguard.tistory.com/182>

<http://jaejunyoo.blogspot.com/2017/01/backpropagation.html>

### ④ L2 Loss function

	L1	L2
loss	<p>Take the absolute value of the difference between the actual value (y) and the predicted value (f(x)) and find the loss in the direction that minimizes the sum of the error. = LAD (Least Absolute Deviations)</p>	<p>The loss is calculated by summing the squared errors between the actual and predicted values. = LSE (Least square error)</p>
	$L = \sum_{i=1}^n  y_i - f(x_i) $	$L = \sum_{i=1}^n (y_i - f(x_i))^2$

Regularization	$\text{cost}(W, b) = \frac{1}{m} \sum_i^m L(\hat{y}_i, y_i) + \lambda \frac{1}{2} \ w\ ^2$	$\text{cost}(W, b) = \frac{1}{m} \sum_i^m L(\hat{y}_i, y_i) + \lambda \frac{1}{2} \ w\ ^2$
Normalization	$\ \mathbf{w}\ _1 =  w_1  +  w_2  + \dots +  w_N $	$\ \mathbf{w}\ _2 = ( w_1 ^2 +  w_2 ^2 + \dots +  w_N ^2)^{\frac{1}{2}}$
Characteristic	<ul style="list-style-type: none"> <li>- Robust</li> <li>- Unstable solution</li> <li>- Possibly multiple solution</li> <li>- Less affected by outliers</li> </ul>	<ul style="list-style-type: none"> <li>- Not very robust</li> <li>- Stable solution</li> <li>- Always one solution</li> <li>- Very sensitive to outliers</li> </ul> <p>* outlier: Data that is too high or too low compared to other values in the data</p>

<https://junklee.tistory.com/29>

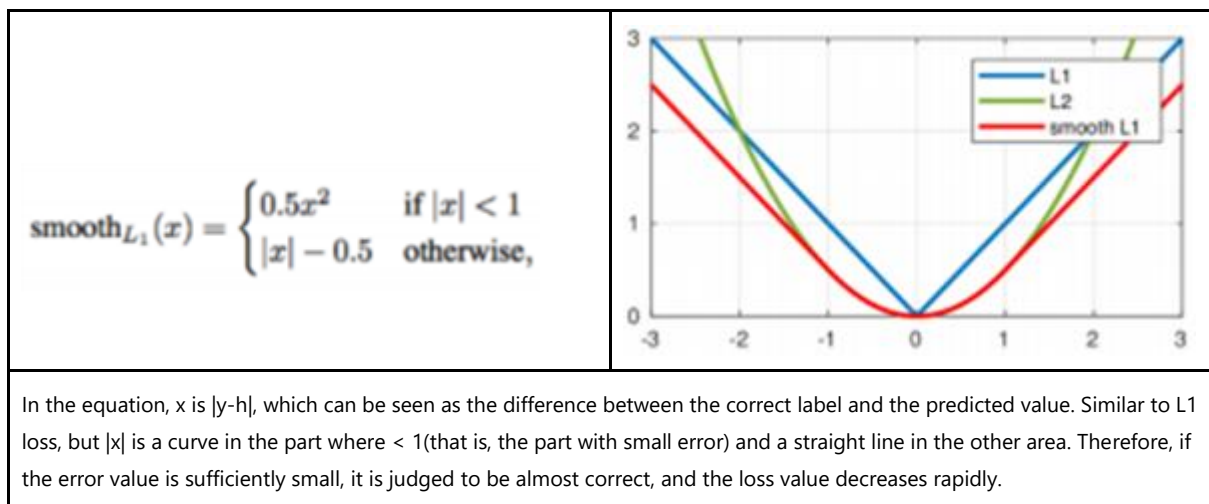
<https://www.stand-firm-peter.me/2018/09/24/1112/>

<https://89douner.tistory.com/19>

<https://light-tree.tistory.com/125>

[https://seongkyun.github.io/study/2019/04/18/11\\_12/](https://seongkyun.github.io/study/2019/04/18/11_12/)

#### ⑤ Smooth L1 Loss function



<https://ganghee-lee.tistory.com/33>

#### ⑥ ROI Pooling layer

ROI(Region of Interest)

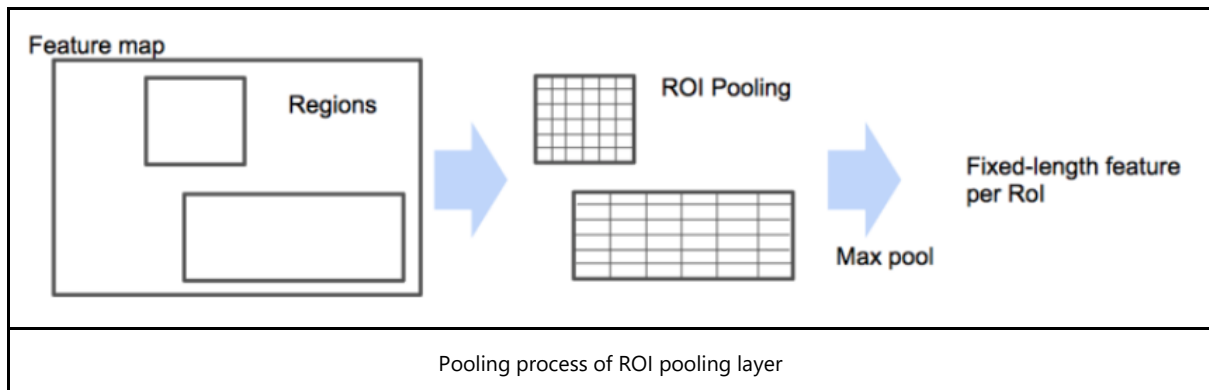
: Region of interest within the image

- In R-CNN, several regions are cropped through the region proposal algorithm, and these regions are called ROI. In addition, the size of each region is different in Fast R-CNN.

Instead of warp in R-CNN, 'Spatial Pyramid Pooling', which is a method using max pooling, is used to make output images of the same size. That is, this pooling is called ROI Pooling.



+) warp: cropping each region proposal region from the image and making them the same size



<https://ganghee-lee.tistory.com/33>

<https://blog.lunit.io/2017/06/01/r-cnns-tutorial/>

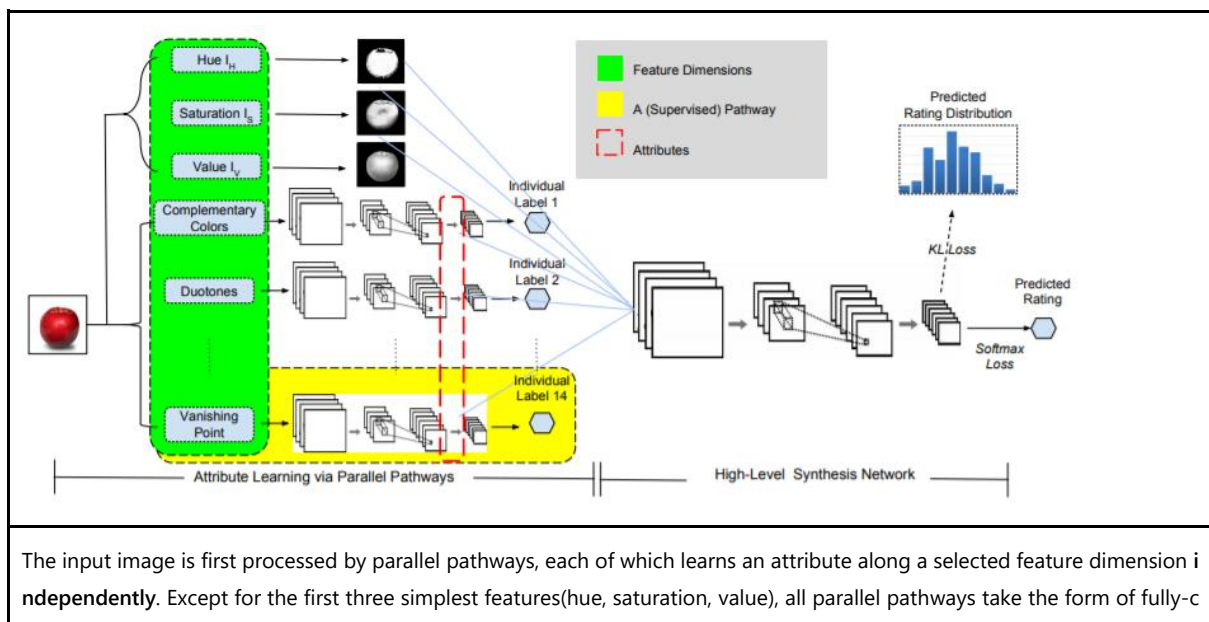
### #3. Aesthetic-based Clothing Recommendation(2018)

by Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong and Zheng Qin

① BDN(Brain-inspired Deep Network)

: The whole training process is divided in two stages, based on insight.

→ This network first learns attributes through parallel(supervised) pathways, over the selected feature dimensions. It then combines those “pre-trained” pathways with the high-level synthesis network, and jointly tunes the entire network to predict the overall aesthetics ratings.



onvolutional networks, supervised by individual labels; their hidden layer activations are utilized as learned attributes. We then associate those “pre-trained” pathways with the high-level synthesis network, and jointly tune the entire network to predict the overall aesthetics ratings. In addition to the binary rating prediction, we also extend BDN to predicting the rating distribution, by introducing a Kullback-Leibler(KL)-divergence based on loss of the high-level synthesis network.

<https://arxiv.org/pdf/1601.04155.pdf>

## ② BPR(Bayesian personalized ranking)

: In general, Bayesian optimization is find the parameter ‘ $\Theta$ ’ that maximizes the posterior probability. This is called ‘Maximum A Posteriori Estimation’.

- Posteriori Estimation: This is the probability of a parameter for which some information is considered. For example, user preference information( $>_u$ ). A contrasting concept is is ‘prior probability’ ( $p(\Theta)$ ), which is prior information about a parameter. Since the **prior probability** is given information, unlike the posterior probability,, it depends only on parameters.

$$p(\Theta | >_u) \propto p(>_u | \Theta) p(\Theta)$$

$$\therefore p(\Theta | >_u) = \frac{p(\Theta, >_u)}{p(>_u)} = \frac{p(>_u | \Theta)p(\Theta)}{p(>_u)} \propto p(>_u | \Theta) p(\Theta)$$

<https://leehyejin91.github.io/post-bpr/>

<https://arxiv.org/ftp/arxiv/papers/1205/1205.2618.pdf>

## ③ Amazon clothing dataset

<https://jmcauley.ucsd.edu/data/amazon/>

<https://data.world/promptcloud/fashion-products-on-amazon-com>

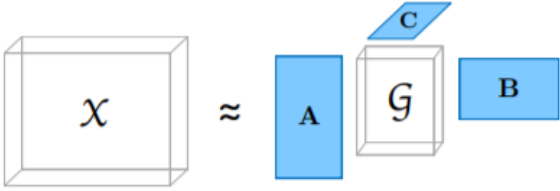
## ④ sparsity problem & cold start problem

- Data sparsity arises from the phenomenon that users in general rate only a limited number of items.
- Cold start refers to the difficulty in bootstrapping the Recommender systems(RSs) for new users or new items.

[https://www.researchgate.net/publication/241770998\\_Resolving\\_Data\\_Sparsity\\_and\\_Cold\\_Start\\_in\\_Recommender\\_Systems](https://www.researchgate.net/publication/241770998_Resolving_Data_Sparsity_and_Cold_Start_in_Recommender_Systems)

⑤ Tucker decomposition(TD) decomposes a tensor into a so-called core tensor and multiple matrices which correspond to different core scalings along each mode. Therefore, the Tucker decomposition can be seen as a higher-order PCA.



 <p style="text-align: center;"><b>Figure 8: Tucker Decomposition</b></p>	<p>In the 3-way tensor case, we can express the problem of finding the Tucker decomposition of a tensor <math>X \in \mathbb{R}^{I \times J \times K}</math> with <math>G \in \mathbb{R}^{P \times Q \times R}</math>, <math>A \in \mathbb{R}^{I \times P}</math>, <math>B \in \mathbb{R}^{J \times Q}</math>, <math>C \in \mathbb{R}^{K \times R}</math>.</p>
<p><math>\mathbb{R}^{P \times Q \times R}</math>, <math>A \in \mathbb{R}^{I \times P}</math>, <math>B \in \mathbb{R}^{J \times Q}</math>, <math>C \in \mathbb{R}^{K \times R}</math> as follows:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\min_{\hat{X}} \ \mathcal{X} - \hat{\mathcal{X}}\  \quad \text{with} \quad \hat{\mathcal{X}} = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R g_{pqr} \mathbf{a}_p \otimes \mathbf{b}_q \otimes \mathbf{c}_r</math> <math display="block">= \mathcal{G} \times_1 A \times_2 B \times_3 C</math> <math display="block">= [\![\mathcal{G}; A, B, C]\!]</math> </div> <p style="text-align: right;">(32)</p>	<p>In this setting, G is the core tensor, which expresses how and to which extend different tensor elements interact with each other. The factor matrices A, B, and C are often referred to as the principal component in the respective tensor mode.</p> <p>We can already see, that if we pick <math>P &lt; I</math>, <math>Q &lt; J</math>, and <math>R &lt; K</math>, this will result in a compression of X, with G being the compressed version of X.</p>

+) PCA(principal component analysis) is to transform the preservation of a high-dimensional space into a low-dimensional space with no linear association by finding the basis(axes) that are orthogonal to each other while taking full advantage of the variance of the data.

<https://arxiv.org/pdf/1711.10781.pdf>

[https://www.alexejgossmann.com/tensor\\_decomposition\\_tucker/](https://www.alexejgossmann.com/tensor_decomposition_tucker/)

<https://ratsgo.github.io/machine%20learning/2017/04/24/PCA/>

⑥ PITF(Pairwise Interaction Tensor Factorization) is a special case of the **TD model with linear runtime both for learning and prediction**. PITF explicitly models the pairwise interactions between **uses, items and tags**. The model is learned with an adaption of the **Bayesian personalized ranking(BPR) criterion** which originally has been introduced for item recommendation. Empirically, they show on real world datasets that this model outperforms TD largely in run-time and can even achieve better prediction quality.

[http://videolectures.net/wsdm2010\\_rendle\\_pit/](http://videolectures.net/wsdm2010_rendle_pit/)

<https://pdfs.semanticscholar.org/bf2b/10e9a3bb73499666facf376918e8c275734e.pdf>

(Original\_Author\_Steffen Rendle, Google, Inc)

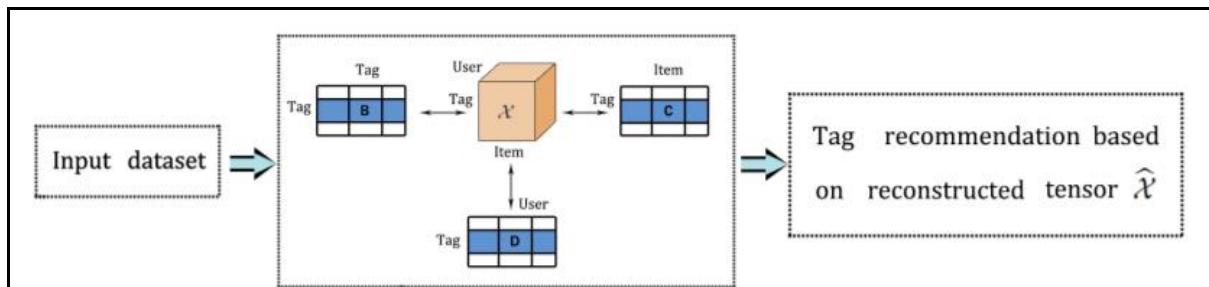
<http://www.wsdm-conference.org/2010/proceedings/docs/p81.pdf>

<https://github.com/yamaguchiyuto/pitf>

[://github.com/yamaguchiyuto/pitf](https://github.com/yamaguchiyuto/pitf)

⑦ CMTF(Coupled Matrix and Tensor Factorization)

: In order to address data sparsity, missing value, and overfitting problems in a social tagging system, a coupled matrix and tensor factorization(CMTF) method named Tagrec-CMTF for tag recommendation is proposed.



**FIGURE 1.** The proposed CMTF-based tag recommendation approach.

In the CMTF method, they decompose the **tag-item-user tensor joint with tag graph** and two auxiliary matrices by using the CMTF, optimize the learning parameters with an alternating direction method of multipliers algorithm, and recommend the tag according to the predicted tensor.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8506367>

<https://arxiv.org/pdf/1105.3422.pdf>

⑧ AVA(Aesthetic Visual Analysis) dataset contains **over 250,000 images** along with a rich variety of meta-data including a large number of aesthetic scores for each image, **semantic labels for over 60 categories** as well as labels related to photographic style for high-level image quality categorization.

<https://academictorrents.com/details/71631f83b11d3d79d8f84efe0a7e12f0ac001460>

## #4. Convolutional Neural Networks for Fashion Classification and Object Detection

by Brian Lao and Karthik Jagadeesh

→ It assessed accuracy, precision, recall, and F1-Score on the ACS dataset using the standard AlexNet Convolutional network which had been pre-trained using ImageNet.

- Clothing Type Classification is the multiclass classification problem of predicting a single label that describes the type of clothing within an image.(ACS, Apparel Classification with Style Data set)
- Clothing Attribute Classification is the problem of assigning attributes such as color or pattern to an article of clothing.(CA, Clothing Attribute Dataset)
- **Clothing Retrieval** encompasses the task of **finding the most similar clothing items** to query clothing items.