COSE474-2022F: Final Project Report
Shoes Recommender System Model using CNN-based image similarity

Minjeong Ban

1. Introduction

These days there are too much information in every domain such as watching media, listening to the music, buying clothes etc. Thus, recommender system become mandatory system for many situations. Netflix and Melon recommend movies and music for users by their own recommender system. They usually recommend by content-based filtering and collaborative (user-based) filtering. However, collaborative filtering has "cold start" problem that users cannot get good recommendation because of lack of user data for recommender system. Also, content-based filtering usually uses text data.

In this project, using image data for recommender system could solve "cold start" problem and new trial for content-based filtering. Calculate the similarity of images and recommend similar products. This recommender system is useful for domain where the visual feature is important such as fashion(shoes). Also, just comparing the pixels of images is not accurate and effective to calculate similarity. Thus, I use CNN models to extract image features and calculate similarity by these features.

1.1 Motivation

I have been study recommender system this semester such as content-based filtering and collaborative filtering of Netflix. So, I want to make another recommender system myself. However, collecting user-data is so difficult and only using content-based(only use text data) filtering cannot make great recommender system. Thus, I want to use deep learning in recommender system and decide to use image data.

1.2 Problem definition

In fashion items, the visual features are very important. The mood, style of fashion is come from visual features such as shape, color etc. By deep learning, we can extract feature of images and compare with others. Thus, in this project I decide to make shoes recommender system. The goal is to recommend similar style of shoes by comparing image features extracted by CNN model not just simple pixel distance similarity and text data.

1.3 Concise description of contribution

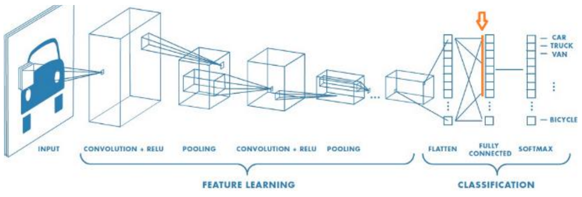
I try to use deep learning to recommender system to improve two things. One is that get out of limitation of using text data such as category, color of shoes and user-data. The another is that calculate image similarity by pixel-to-pixel distance is not accurate and inefficient. So, extracting feature by CNN, calculate image similarity by feature-to-feature distance.

2. Methods

The method is simple. In each shoes images, extract the feature vector by CNN models. In this project, I used pretrained Resnet-18 model and VGG-16 model. Like this, we can express image data to vector. Then calculate similarity between images feature vectors. We can calculate similarity with Euclidean distance and cosine similarity. Finally, we can recommend shoes that have such high cosine similarity (or low distance) with user's favorite shoes.

2.1 Feature extraction

We can extract feature vector from the image by CNN models. 'feature vector' is a list of numbers taken from the output of a neural network layer. Feature vector is a dense representation of the input image and can be used for ranking, classification, clustering etc.



Above of the picture, We can extract feature vector at orange arrow point from neural network. Get vector before fully connected layer and SoftMax. We don't need to classify image by SoftMax, just extract feature vector.

2.2 Cosine similarity

Cosine similarity is a measure of similarity between two vectors (sequences of number). It always belong s to the interval [-1,1]. Two proportional vectors have a cosine similarity of 1, two orthogonal vectors have 0, and two opposite vectors have -1.

$$similarity = \cos(\Theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

Cosine similarity can be calculated with above calculation. It is also used for text similarity too. We can calculate similarity with Euclidean distance (L2-norm), but cosine similarity is better in vector space.

3. Experiments

3.1 Dataset

I crawled Musinsa Codimap website (https://www.musinsa.com/app/codimap/lists) to get shoes images sorted by clothes coordination. I crawled 60 shoes for each clothes coordination with Selenium and BeautifulSoup. There are 8 clothes coordination for man and 11 clothes coordination for woman. Thus, I crawled total 1140 shoes images.

3.2 Computing resource

I worked in google colab and ran with CPU. Used libraries are Tenserflow and Keras, Pytorch, Pandas, numpy, Scikit-Learn, bs4(BeautifulSoup), Selenium.

3.3 Experimental design/setup

For 1140 images, I extracted feature vectors and calculate cosine similarity. I try to compare image similarity computed by Euclidean distance pixel-to-pixel, cosine similarity feature-to-feature based on CNN, and Euclidean distance feature-to-feature based on CNN. Also, I used two different pretrained CNN models, VGG16 and Resnet18. Thus, I compared total 4 results.

For Euclidean distance, the smallest distance is the most similar image and for cosine similarity, the biggest number up to 1 is the most similar image.

4. Results

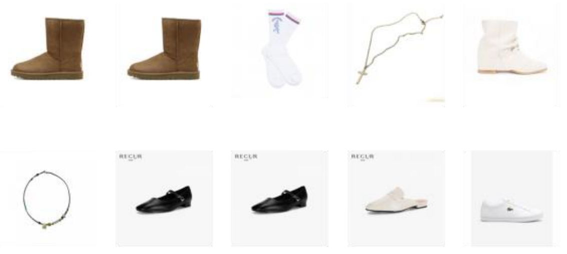
Because it is hard to collect label data such as user's like and buy data, it is hard to compute quantitative result such as accuracy, precision etc. So, I could only get qualitative results.

4.1 Qualitative results

I tested shoes recommender system by get 9 output of similar images for one particular image.

First, I input boots image. The images next to first boots image is output of recommender system and the number list below the 10 shoes images is the Euclidean distances or cosine similarity.

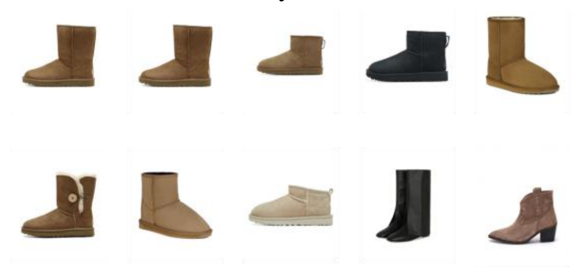
Euclidean distance (L2-norm) – RGB space



[0.0, 22172, 22780, 22892, 22979, 22998, 22998, 23011, 23074]

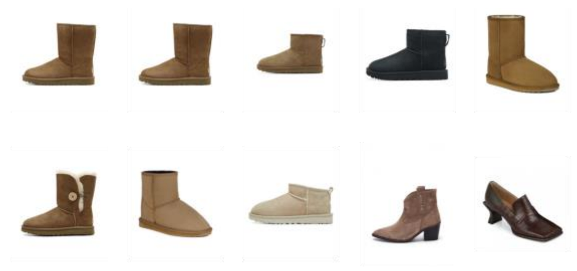
Output of Euclidean distance pixel-to-pixel (RGB space) is awful. It found same image, but it could not find similar images.

VGG16 – cosine similarity



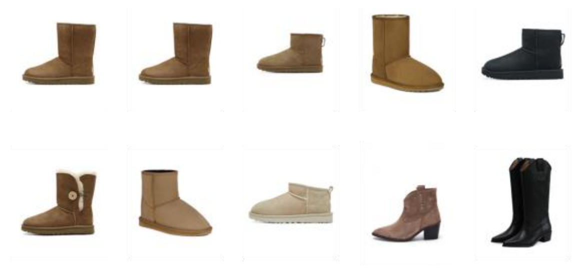
[1.0, 0.82, 0.80, 0.77, 0.75, 0.74, 0.71, 0.69, 0.68]

Resnet18 – cosine similarity



[0.99, 0.92, 0.91, 0.91, 0.89, 0.88, 0.87, 0.84, 0.83]

Resnet18 – Euclidean distance



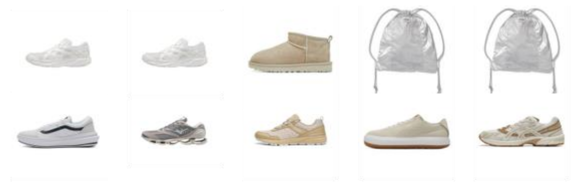
[0, 241, 269, 279, 292, 308, 329., 350, 385]

Outputs of VGG16(cos similarity) and Resnet18(cosine similarity) and Resnet18(Euclidean distance) are almost same. They found same image and similar images sufficiently because their outputs are all boots not sneaker or high heels. Also, color, shape and shoes material are similar.

Thus, VGG16 and Resnet18 have similar performance. Also, after feature extraction, calculate similarity with Euclidean distance and cosine similarity are also have similar performance.

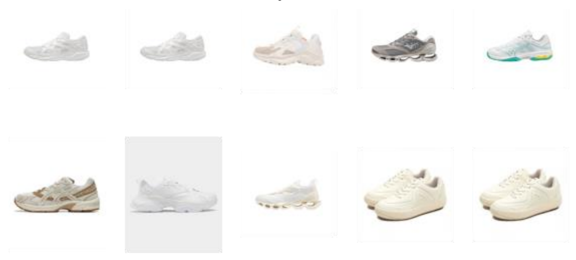
I will show another example.

Euclidean distance (L2-norm) – RGB space



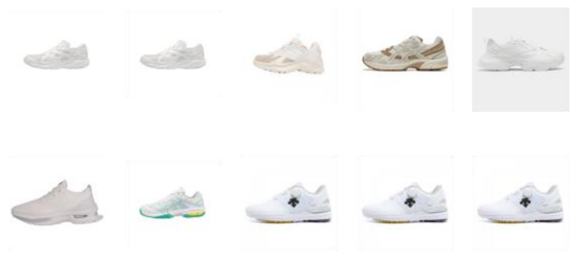
[0.0, 13688, 17211, 17211, 17483, 17612, 18102, 18312, 18992]

VGG16 – cosine similarity



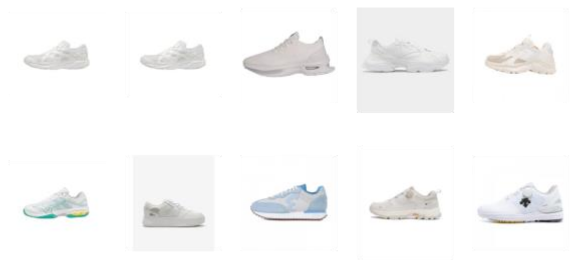
[1.0, 0.87, 0.85, 0.84, 0.83, 0.82, 0.81, 0.79, 0.79]

Resnet18 – cosine similarity



[0.99, 0.87, 0.87, 0.86, 0.86, 0.86, 0.85, 0.85, 0.85]

Resnet18 – Euclidean distance



[0, 295, 296, 298, 309, 338, 338, 344, 346]

Second example also has same result with first example.

5. Conclusion and Future direction

I tried to adapt deep learning to recommender system. Especially I used image data which I directly crawled famous Korea fashion website Musinsa. Also, compare several outputs from different methods.

Between simple Euclidean distance in RGB space and feature extraction based on CNN has explicit performance difference. Using CNN has better performance. I think when we calculate Euclidean distance after detecting and crop the shoes images, the performance would improve. I will try this experiment later.

However, between CNN models (VGG16, Resnet18) and between different similarity calculation after feature extraction don't have explicit difference. Thus, I could get result that the feature extraction based on CNN has a biggest contribution for searching similar image.

Visual recommender system in this project is too simple model. So, later I want to try more complex model like SOTA such as style2Vec(2017) – combining multiple CNNs, Generative Image model with GAN(2017), explainable outfit recommendation with comment generation(2019).

As a result, the domain where the visual component is important could use recommender system with image data transformed by CNN or other techniques. When we add image data with text, user data for recommender system, we can have better recommender system. Also, 'Serendipity' is being important in recommender system because there are too many only similar things.

References

[1] Shankar, Devashish & Narumanchi, Sujay & Ananya, H & Kompalli, Pramod & Chaudhury, Krishnendu. (2017). Deep Learning based Large Scale Visual Recommendation and Search for E-Commerce.

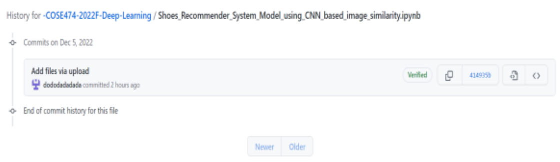
[2] 강주희 and 이윤정. (2020). A Recommender System Model Using a Neural Network Based on the Self-Product Image Congruence. 한국의류학회지, 44(3), 556-571.

[3] 서민지(Minji Seo),and 이기용(Ki Yong Lee). "이미지 유사도 측정을 위한 CNN 기반 이미지 임베딩 모델 비교 분석." 한국정보과학회 학술발표논문집 2019.6 (2019): 751-753.

[4] 변정. "개인 선호이미지를 이용한 딥러닝기반 이미지 추천시스템 설계와 구현." 국내석사학위논문 상명대학교 일반대학원, 2017. 서울

Appendix

https://github.com/dododadadada/-COSE474-2022F-Deep-Learning



History for -COSE474-2022F-Deep-Learning / Shoes_Recommender_System_Model_using_CNN_based_image_similarity.ipynb

Commits on Dec 5, 2022

Add files via upload
dododadadada committed 2 hours ago

Verified 4149359

End of commit history for this file