Fashion recommendation systems aim to personalize and facilitate the shopping experience by suggesting outfits that match users' personal fashion preferences. Providing clothing recommendations based on users' tastes and preferences can increase user satisfaction with online shopping sites and strengthen customer loyalty. The recommendation system designed in this study will not only personalize the user's shopping experience, but also save them time and make the shopping process more efficient. Assuming that users speed up the process of choosing among thousands of products by receiving recommendations that suit their personal taste and style, personalized recommendations may encourage users to purchase more products. This, in turn, increases the revenues of online shopping sites and supports business growth. Therefore, fashion recommendation systems provide great benefits to both users and online shopping sites. We have already mentioned that offering clothing recommendations based on users' tastes and preferences can increase user satisfaction. It can also strengthen the customer loyalty of online shopping sites. Therefore, recommending a similar product is the main topic of this study.

In addition to the literature review in the Proposal Report, an additional research on recommendation systems was conducted in the first week. This research can be generally characterized as a continuation of the Proposal Report. In addition, a summary of our work was made to give us an idea of what will be included in the study. In the second week of the approximately one-month study period, designing a model from scratch with CNN and testing this model was discussed. The CNN mentioned in the literature review is intended to be used in the design of the recommendation system, but it is intended to be learned in detail, and no product recommendation has been studied. The study was conducted on the Cifar10 dataset. [1]

PyTorch is one of the most popular and widely used deep learning libraries – especially within academic research. It's an open-source machine learning framework that accelerates the path from research prototyping to production deployment and we'll be using it today in this article to create our first CNN. We will be using the CIFAR-10 dataset. The dataset has 60,000 color images (RGB) at 32px x 32px belonging to 10 different classes (6000 images/class). The dataset is divided into 50,000 training and 10,000 testing images. The steps of loading the dataset and designing the model are described in more detail in the notebook with the code. The loss values and accuracy values of the model at each epoch are given Figure 1 and Figure 2.

```
Epoch [1/20], Loss: 1.8847
Epoch [2/20], Loss: 1.5313
Epoch [3/20], Loss: 1.7422
Epoch [4/20], Loss: 1.3382
Epoch [5/20], Loss: 1.9132
Epoch [6/20], Loss: 1.0077
Epoch [7/20], Loss: 1.3437
Epoch [8/20], Loss: 0.4538
Epoch [9/20], Loss: 1.4301
Epoch [10/20], Loss: 1.1297
Epoch [11/20], Loss: 1.1042
Epoch [12/20], Loss: 0.6595
Epoch [13/20], Loss: 0.8148
Epoch [14/20], Loss: 0.7682
Epoch [15/20], Loss: 0.8232
Epoch [16/20], Loss: 0.8483
Epoch [17/20], Loss: 0.9479
Epoch [18/20], Loss: 0.8178
Epoch [19/20], Loss: 0.3819
Epoch [20/20], Loss: 0.6994
```

Figure 1: Table of changes in loss values depending on the number of epochs

```
Accuracy of the network on the 50000 train images: 83.356 %
```

Figure 2: Accuracy value after testing the model on the test dataset

While the success of the CNN model designed at the basic level was at a good level, we had the opportunity to discover many notebooks on CNN on the Kaggle platform. In Week 3, we examined and worked on the first of these notebooks. This notebook was about designing a recommender system. In this notebook, where we learned the concepts of transfer learning and pre-trained models, we used VGG16, a deep learning model widely used in image classification tasks. [2] The model has a 16-layer (weighted layers) architecture and is therefore named VGG16. VGG16 is a model trained on the ImageNet dataset. This model is used as a basis for various image processing and image classification problems. As a result of feature extraction with VGG16, cosine similarity, which is basically a distance measure that allows us to measure the similarity of vectors by calculating the cosine of the angle between two vectors, is calculated and similar products are listed as an example can be seen in Figure 3.

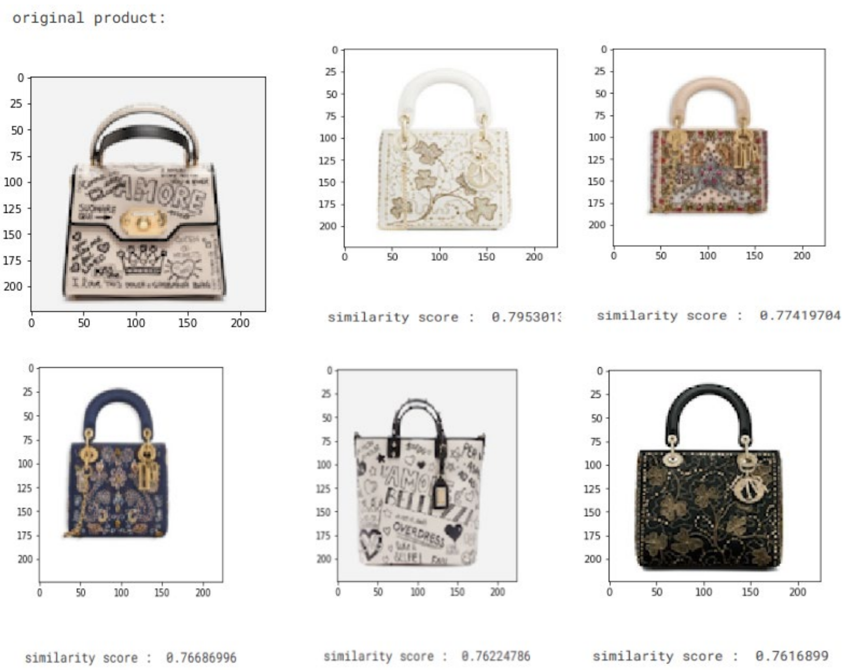

Figure 3: Recommendation system lists 5 products similar to the original product

The following week, we worked on another recommendation system one level above the VGG16 modeled recommendation system we had been working on. In this system, we had the opportunity to research on the definitions of recommendation systems in addition to VGG16. [3] The first of these, Content-Based Recommendation Models, examines the content of products and user preferences and creates recommendations. This approach uses content features such as descriptions, labels and categories of products and performs similarity calculations for these features used throughout the process. Furthermore, by analyzing the user's past preferences and behaviors, it can provide customized recommendations tailored to the user's interests. [4]

Collaborative Filtering, another recommendation system model, uses users' past evaluations and related data. Thus, this method makes recommendations by identifying user similarities. This designed model utilizes the ability to predict users' opinions about other products when users review similar or identical products in order to provide recommendations. [4] The last recommendation system we investigated in this study, Hybrid Recommendation Models, aims to provide more comprehensive and accurate recommendations by combining various recommendation techniques. By combining the strengths of each technique, hybrid models improve the user experience, identify user preferences, and generate product recommendations. This approach evaluates the suitability of products by comparing product features and user needs, and thus creates a system that identifies products that best meet users' needs based on the user's past preferences and behavior. [4]

In the notebook we worked on last week, we considered Content-Based filtering based on the description of an item and a profile of the user's preferred options. Unlike the similarity calculation with Cosine Similarity, in this system where KNN algorithm is used, given the number of images in the dataset and the features extracted, an output matrix of 10000 images x 512 features is obtained. Before giving this matrix to the KNN algorithm, PCA is used and product recommendation is realized as shown in Figure 4.

Figure 4: Products recommended by the recommendation system

3. In the suggestion system we worked on in the last week, which is not expected to be evaluated like the suggestion system worked on in the 3rd week, the success of the system is expected to be measured only through customer feedback.

References

[1] Web Source: https://github.com/ogulcanakcaa/Fashion-Recommendation-System/blob/main/Codes/Writing_CNNs_from_Scratch_in_PyTorch.ipynb

[2] Web Source: https://github.com/ogulcanakcaa/Fashion-Recommendation-System/blob/main/Codes/product-recommendation-based-on-visual-similarity.ipynb

[3] Web Source: https://github.com/ogulcanakcaa/Fashion-Recommendation-System/blob/main/Codes/visually-similar-product-recommendation.ipynb

[4] Oğulcan Akça, Öneri Sistemi ile Ürün Tavsiyesi, YBS Ansiklopedi, v. 12, is. 1, 2024