FASHION RECOMMENDATION SYSTEM DEVELOPMENT PROJECT

Doğa Gündoğar, Gülsüm İrem Baş
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{doga.gundogar, gulsum.bas}@yildiz.edu.tr

***Özetçe* —Online alışveriş günümüzde oldukça yaygın bir şekilde kullanılmaktadır. Sektörün büyümesiyle birlikte şirketler kullanıcıların ilgisini çekmek için diğer şirketlerle yarış haline girmiştir. Her bir firma, kullanıcıları kendi ürünlerinden almaya teşvik etmenin yollarını aramaktadır. Bu kapsamda öneri sistemleri büyük rol oynamaktadır. Kullanıcının ilgisini çeken ürünlerin benzerlerini göstererek onları sistemde tutma, zevkine göre seçtiği ürünlerin muadillerini sunarak daha fazla ürün almalarını sağlamak amaçlanmaktadır. Projemizde bu doğrultuda bir kıyafet öneri sistemi tasarlanmıştır. Potansiyel müşterinin yüklediği kıyafet resmine yakın benzerlikte ürünler müşteriye sunularak beğenisini ve ilgisini kazanmak amaçlanmıştır. Sisteme yüklenen her türden resmin işlem görmesi için öncelikle görüntü işleme yöntemleri kullanılarak özellik çıkarımı yapılmıştır. Sonrasında farklı benzerlik metrikleri denenerek en uygun örneklerin sunulması sağlanmıştır.**

Anahtar Kelimeler—Kıyafet Öneri Sistemi, Benzerlik Algoritmaları, Görüntü İşleme

***Abstract*—Online shopping has become very popular these days. As the industry grows, companies are competing with others for user attention. Every company is looking for ways to persuade users to buy their products. In this context, recommendation systems play an important role. Its purpose is to keep users in the system by showing them similar products that are of interest to them and to encourage them to buy more products by offering products that are similar to the ones they have selected according to their preferences. It is to get In our project, a fashion recommendation system was designed in this direction. The goal is to arouse the potential customer's appreciation and interest by presenting products that closely resemble the clothing image that the potential customer has uploaded. To process all kinds of images uploaded to the system, feature extraction was first performed using image processing techniques. A successful proposal presentation was then ensured by trying different similarity measures.**

Keywords—Fashion Recommendation System, Similarity Algorithms, Image Processing

I. INTRODUCTION

Today, the fashion industry is constantly differentiating and growing with new products. With hundreds of new styles and products being released every day, it is difficult for users to keep up with this growing industry and discover new products. This requires users to spend a large amount of time to find the right products. At the same time, fashion producers and sellers who cannot meet enough customers are negatively affected. In this context, garment recommendation systems are of great importance to recommend the most suitable products to consumers and provide more personalized services.

With the developing technology, many companies in the market have started to use garment recommendation systems. For example, Amazon uses advanced recommendation algorithms to provide customers with more personalized products [1]. These systems use technologies such as deep learning and image processing to enable customers to discover more products and personalize their shopping experience. At the same time, these systems increase customer satisfaction and provide companies with more targeted marketing and more efficient inventory management.

By providing personalized recommendations based on users' preferences and needs, fashion recommendation systems make the shopping process more efficient and enjoyable. This also offers fashion brands and retailers a unique opportunity to build customer loyalty, increase sales and improve customer satisfaction.

Fashion recommendation systems can solve many of the challenges facing the fashion industry. However, advanced and customized algorithms are required to effectively implement these systems. Moreover, users' different fashion preferences and shopping habits are important factors that need to be taken into account for these systems to make accurate and effective recommendations. Therefore, a comprehensive understanding and analysis of how to improve the user experience is a key part of successfully implementing this type of recommendation system.

This project aims to better understand users' fashion preferences and shopping habits in order to provide them with the most relevant and personalized clothing recommendations. Furthermore, this system aims to help users easily and effectively stay up-to-date with the latest trends in the fashion industry, in addition to saving their time.

A. Project Motivation and Scope

The Fashion Recommendation System project aims to attract the attention of potential customers and increase their purchase rates by providing them with personalized recommendations. The technologies used in the realization of the project are as follows:

In the project, feature extraction is performed from the image given by the user. For this process, the resnet50 architecture, which is popular and has achieved the highest appreciation in its comparisons, is used. ResNet-50 is a popular model based on the Residual Networks (ResNet) architecture, which is an important breakthrough in the field of deep neural networks and image recognition. ResNet has achieved particularly high success in tasks such as image classification and object recognition [2]. After extracting image features, KNN, Pearson coefficient, and cosine similarity algorithms were run on the data. The algorithm with the highest success rate was used for the recommendation presentation. Finally, a simple interface was designed so that the user can easily operate the system.

B. Challenges

The challenges that may be encountered during the construction phase of the project are as follows:

- Data Set Preparation and Preprocessing: Dealing with large data sets requires time. It may be necessary to clean the data and correct missing or erroneous data. The pre-processing process can be made difficult by differences in the size of the images.
- Feature Extraction Process: Extracting features from images is a processor and memory-intensive process. If working on a very large dataset, it takes hours to extract these features using deep learning and save them as pickle files.
- Model Training: Selecting appropriate hyperparameters for deep learning models and optimizing the model can take a long time. The process also requires high-performance hardware as it requires high processing power. Selecting the
- Measurement Metrics: Choosing the appropriate similarity algorithm is crucial for the performance of the model.
- Memory Management: Working with high-dimensional data sets can cause performance issues.
- User Interface and Usability: Creating a simple and easy-to-use user interface is important for user satisfaction.
- Server Configurations: Technical issues such as running the application in a live environment and server configurations can be encountered.

II. Literature Review

Fashion and product recommendation systems use emerging technologies such as Deep Learning and Knowledge-Based Systems to recommend products that users might like and that match their personal tastes and desires. With the rise of e-commerce platforms, people are less likely to go to the store to shop, which has led to increased investment and interest in this area. As stores and e-commerce platforms began to compete in the online space, they began to offer more personalized recommendations by collecting users' previous purchases and personal preferences. In this way, the emergence and acceleration of clothing recommendation systems and the use of different technologies in this context have been realized in a short period of time. One of the studies in this field is developing a fashion recommendation system that can understand fashion products and styles using a convolutional neural network in 2016 [3]. In another developed work, the fashion product understanding system was implemented using an LTSM and an auto coder [4]. Before this 2018 study, a fashion recommendation system was developed in 2017, again using LTSM. In this work, we worked on finding products that match existing products and obtaining views that resemble the products or descriptions entered by users [5]. One of the studies that enable the development of such systems is "VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback". This study emphasizes that visual information in recommendation systems will bring more accurate recommendations and a better understanding of the personal preferences of users [6]. In conclusion, as can be understood from the literature, visual information and deep learning algorithms have gained importance in order to provide personalized, accurate, and comprehensive recommendations to users.

III. Preliminary Review

With the ever-changing and diversifying fashion industry, fashion recommendation systems are needed for users to access the products they want more easily. This project, it is aimed to create a Fashion Recommendation System that shows different options in line with the selected product in order to meet the needs of users.

A. Need for the Project

1) Providing fashion recommendations based on user's personal preferences and tastes
2) Making the shopping experience more convenient and efficient for users
3) Increasing customer satisfaction, increasing sales, and customer confidence in brands

B. Scope

The fashion recommendation system intended to be built in this project consists of the following components:

1) Collection and processing of product metadata: Collecting and organizing metadata such as attributes, images, and branding of products
2) Recommendation algorithm: Developing an algorithm that provides users with the most appropriate clothing recommendations using product metadata
3) User interface: Designing a simple interface where users can see products similar to their previous preference

C. Road Map

1) Determination of project requirements and scope
2) Research and organization of dataset for user profile and product metadata

3) Investigate and select the necessary machine learning and data analytics techniques for the recommendation algorithm
4) User interface design and development
5) Integration and testing of all components
6) System improvements

IV. Application

In this project, image processing and deep learning methods are used to create a fashion recommendation system. Using the pre-trained ResNet50 model, feature vectors are extracted from product images, and similar product recommendations are provided with these feature vectors. The application interface was implemented using the Python Flask library. The implementation parts of the project are shown below.

A. Feature Extraction with the RestNet50 Model

RestNet50, part of the Residual Networks family, is a deep learning model widely used to solve large-scale image classification problems. RestNet50 is pre-trained on millions of images from the ImageNet dataset and is used especially in transfer learning. It generally shows high performance.

1) RestNet50 Architecture: RestNet50 consists of a total of 50 layers. The model includes residual blocks consisting of several convolutional layers. A two-stage sampling process is used to minimize the architecture and a global average pool layer is used. Softmax is used as the activation function. The RestNet architecture consists of convolutional, activation, pooling and fully-connected layers [7].

2) Usage of RestNet50: RestNet50 is used in transfer learning to extract features from image data. Pre-trained models save time and resources. In this context, RestNet50 was chosen to be used in the project.

First, $'include\ top = False'$ option is used to remove fully connected layers and only convolutional layers of the model are used. Then the model is organized by adding the GlobalAveragePooling2D layer.

B. Creating Pickle Files

Pickle is a data serialization and reverses serialization module for Python. The process of converting the state of an object into a byte stream is known as serialization. This byte stream can be converted to the original object by the same program or by another program. Pickle "freezes" the internal structure of a Python object by saving it to disk. This object can later be restored and used in the same state. The image feature vectors and image file paths are saved in pickle files named $'image\_features\_embedding.pkl'$ and $'img\_files.pkl'$ respectively. These files are then used for the recommendation system. In this way, the model training phase, which requires a long processing time, is not repeated continuously.

C. Creating the Image Feature Extraction Function

The function $'extract\_img\_features'$ is used to take a given image file path and model and extract the features of the image. It first fits the image to the ResNet50 model and extracts the predicted features of the model. Finally, it returns a normalized version of the features.

D. Suggestion of Similar Products

After feature extraction on the image, similarity algorithms were tried and tested for the part of the system that shows products similar to the uploaded image to the user. The algorithms tested in this context are Cosine similarity, Pearson similarity, and KNN algorithms. Among the algorithms used, the Pearson algorithm is the most successful similarity algorithm. As a result, it was decided to use Pearson similarity as a criterion for recommendation system development.

E. User Interface Development

Flask, a Python web framework, was used in this project to create a dynamic web application. Flask provided the flexibility and simplicity needed to create a functional web interface quickly and simply. The interface designed for the application consists of two main pages. The first screen is the screen where the user can upload images. Here the user uploads a picture of the outfit they want to receive similar outfit suggestions. After uploading, the system switches to the page where the results are shown. On the result page, the 5 most similar pictures to the picture uploaded by the user are displayed on the screen with their features.

1) Upload Page: The upload page is the page ('uploads.html') that allows the user to select a product image from their computer and upload it. The "request" module of the Flask library checks whether the uploaded file is in an acceptable format (i.e. 'jpg', 'jpeg' or 'png') and verifies that the file has been successfully uploaded to the server. Below is an image of the upload page.

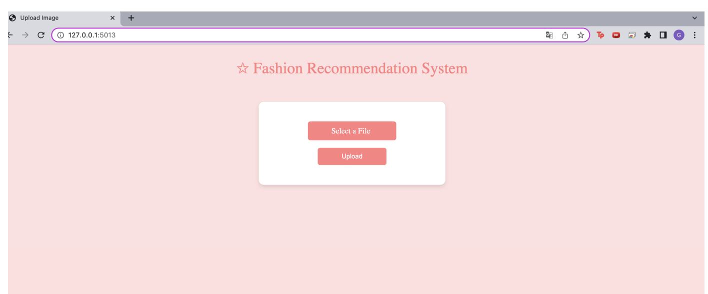

Figure 1 Upload Page

In Figure 1 while the 'Select a File' button allows uploading images to the system, the 'Upload' button takes you to the page where the results are shown.

2) Results Page: If the upload process is successful in the previous stage, the features of the uploaded product are extracted through the ResNet50 model and 5 products similar to the uploaded product are displayed on the results page ('results.html) using the Pearson similarity algorithm. Each recommendation is accompanied by additional information such as the brand, model, and id number of the product. In this way, the user can easily access a similar product they like.

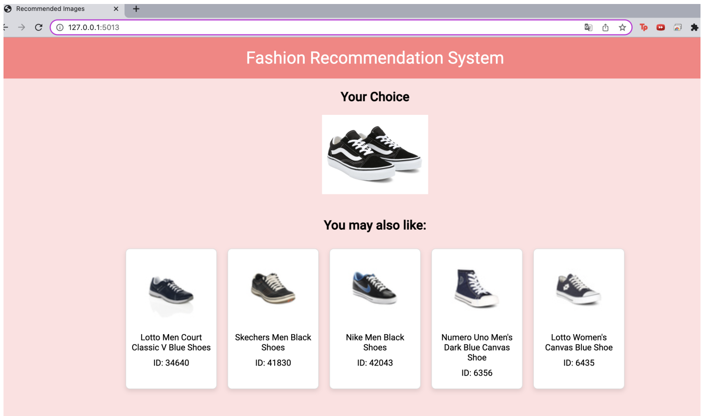

Figure 2 Load Page

As can be seen in Figure 2, the images, brands, and models of the 5 products most similar to the given shoe image are listed.

V. Performance Analysis and Results

The algorithms chosen during the development phase of the project are analyzed in this section and the results are discussed in detail. Then the methodology and testing procedures used to evaluate the success of the implementation are shown.

A. Model Selection and Comparison

In this project, ResNet50 model is used for feature extraction with image processing. In addition, for the recommendation development part of the application, three different algorithms were tested on test data, and the algorithm with the highest success rate was used for system recommendation. Below is a detailed analysis of the selection analysis of ResNet50 and the algorithms used for system recommendation.

1) ResNet50 Model Analysis: The base model used in this study is ResNet50. ResNet50 is a pre-trained image classification model distinguished by its depth (50 layers) and residual connections. The main factors for choosing this model are that it was trained on the ImageNet dataset and its ability to recognize 1000 different object classes [8]. These features provide a solution to the gradient vanishing problem often encountered in deep learning models. ResNet50 is often used in combination with pre-trained weights because it can perform well even with fewer data. This feature has been very useful especially when data is limited [9]. Before selecting ResNet50, other comparable models were also examined. These models are Xception, InceptionV3, VGG16, and VGG19. However, the main reason for choosing ResNet50 is the advantages offered by residual connections and depth [10]. To briefly explain the reasons for not choosing different models, the VGG16 and VGG19 models have lower accuracy rates compared to ResNet50 and require more parameters. InceptionV3 and Xception models have higher accuracy rates but are more complex and have higher computational costs. A table comparing the models is given below. Based on this information, ResNet50 was chosen as the most suitable image processing model for this project [11].

Table 1 Comparison of Models

Model Name	Depth	Parameters	Top-1 Acc.	Top-5 Acc.
VGG16	16	138M	71.5%	90.1%
VGG19	19	144M	72.7%	91.2%
InceptionV3	48	23.8M	78.8%	94.4%
Xception	71	22.9M	79.0%	94.5%
ResNet50	50	25.6M	76.2%	92.9%

2) Analysis of Recommendation Development Algorithms: Recommendation systems use algorithms that aim to recommend products or services to users based on their goals. The Fashion Recommendation System recommends similar but different products to users based on their previous preferences. In this context, various recommendation algorithms that can bring the highest success to the project have been examined. In this study, K-Nearest Neighbors (KNN), Pearson Correlation Coefficient, and Cosine Similarity algorithms are examined. This section describes these algorithms and compares their performance.

- **K-Nearest Neighbors** - The K-Nearest Neighbors (KNN) algorithm is used to determine how similar users are when they rate an item. The KNN algorithm examines other users' reviews to predict how a user will rate an item. Typically, the similarity is based on users' previous preferences. KNN is particularly effective in large data sets with a large number of items and users. The equation for the KNN algorithm is given in (1).

$$P(A|B) = \frac{I(A,B)}{n(B)} \tag{1}$$

- **Pearson Correlation Coefficient** - Pearson Correlation Coefficient is used to find the linear relationship between two variables. It is often used in recommender systems to determine the relationship between users' ratings of an item. If two users have similar habits of rating the same items, they will have a high Pearson Correlation Coefficient. The Pearson Correlation Coefficient is often low except when two people are very similar. Its equation is given in (2).

$$r = \frac{\sum(x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum(x_i - \mu_x)^2 \sum(y_i - \mu_y)^2}} \tag{2}$$

- **Cosine Similarity** - Cosine Similarity is a metric that measures the angle between how two people rate an item. This metric gives a high value when the way two users rate an item is very similar. Cosine Similarity can calculate similarity quickly and efficiently, especially for large data sets. Its equation is given in (3).

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| ||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{3}$$

VI. Measurement of System Success

In order to measure the success of the system, some features of the dataset used in the project are utilized. Some of the metrics used to measure the success of the system measure the accuracy of the proposed items and their similarity with the given product. Figure 3 shows the features used in the dataset.

	id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName
24922	12532	Women	Apparel	Topwear	Jackets	Black	Fall	2011.0	Casual	Puma Women Solid Black Jackets
35553	3469	Unisex	Accessories	Headwear	Headband	Black	Fall	2010.0	Sports	ADIDAS Unisex Black White Headband
31700	24742	Women	Accessories	Bags	Handbags	Green	Winter	2015.0	Casual	Murcia Women Casual Handbag
42694	28477	Women	Apparel	Topwear	Kurtas	Purple	Summer	2012.0	Ethnic	Urban Yoga Women Printed Purple Kurta

Figure 3 Dataset

Based on the features given above, the similarity of the product given by the user with the recommended products is measured by how many of the features are the same. After dividing the dataset into two parts training and testing, the system was implemented using the 'cosine', 'pearson' and 'knn' similarity algorithms respectively. Each image in the separated test data was entered into the system as a user product and the suggestions were observed. The extent to which the features of each input in the dataset matched the features of the proposed products was kept and averaged over the entire test data. The success rate determines how close certain attributes of the recommended item (e.g. 'gender', 'masterCategory', 'subCategory', 'articleType', 'baseColour', 'season', 'year', 'usage') are to the attributes preferred by the user. A separate success rate is calculated for each attribute and the average gives the overall success rate. Figure 4 shows the code block where this process is implemented.

```
def compute_success_rates(recommended_items, input_img_path, styles_df, features_to_check, model):
    input_img_features = styles_df[styles_df['image'] == input_img_path][features_to_check].values[0]
    counters = [0] * len(features_to_check)

    for item in recommended_items:
        item_image_path = "/Users/gulsumirembas/Downloads/project/" + item['image_path']
        item_features = styles_df[styles_df['image'] == item_image_path][features_to_check].values[0]
        for i in range(len(features_to_check)):
            if item_features[i] == input_img_features[i]:
                counters[i] += 1

    success_rates = [count / len(recommended_items) for count in counters]
    return success_rates


def compute_overall_success_rate(success_rates):
    return sum(success_rates) / len(success_rates)
```

Figure 4 Success Measurement

A. *Measurement of System Success*

The algorithm used for the success of the system is described in section 7.2. In this context, K-Nearest Neighbor, Pearson Correlation, and Cosine Similarity algorithms were tested on the test data. The graphs of the results are given in Table 2 and Table 5.

Figure 5 shows the overall success rates of the algorithms on the same test data. Looking at the overall success rates,

Table 2 Test Results of Algorithms

Algorithm	Value
Cosine	0.675
Pearson	0.725
Knn	0.03125

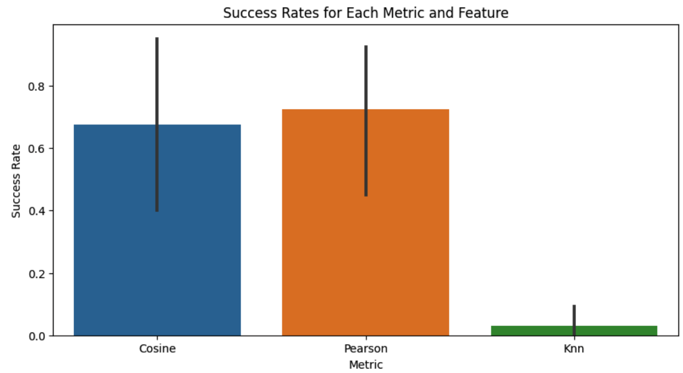

Figure 5 Comparison of Algorithms

it is seen that the Pearson Recommendation metric shows a higher success rate than the Cosine and KNN metrics. As can be seen in Figure 2, Pearson achieved a high result of 0.725 for the success rate. In line with these results, it is seen that Pearson recommendation metric is more consistent than other algorithms and recommends products that are more similar to the product given by the user.

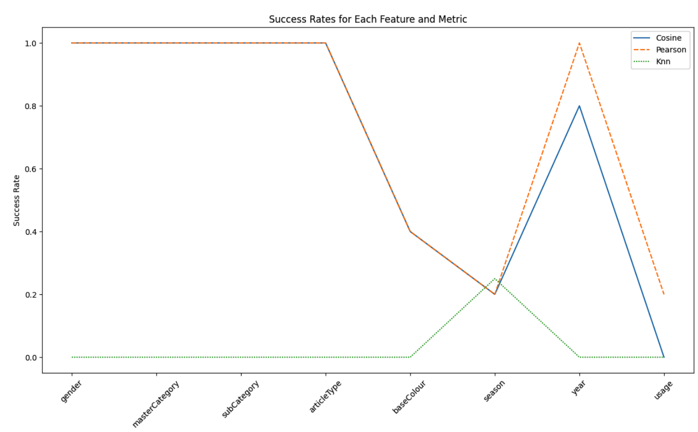

Figure 6 Comparison of Algorithms for Each Feature

Looking at Figure 6, it is observed that for each feature, Pearson and Cosine algorithms generally recommend similar products, but KNN does not achieve a high accuracy rate for most features, while Pearson algorithm has an above-average success rate for all features. It is understood that this algorithm is well-matched with various features in the dataset and makes more balanced and comprehensive recommendations. For these reasons, the Pearson Similarity algorithm was found to be the best choice for this recommendation system.

VII. Results

In this project, a deep learning-based system is developed to extract features from fashion product photos and recommend similar products. The system extracts features from images based on various measurement metrics and recommends similar products using a deep-learning model called ResNet50. During the project, data collection, pre-processing, model training, feature extraction, and recommendation were performed. Working with high-dimensional data and the need to extract features from this data caused difficulties during the construction phase. It is also important to select and apply the right metrics to evaluate the performance to be successful.

The developed system can be very useful for the fashion industry. It can help brands to customize the shopping experience by tailoring their products to customers' interests and tastes. However, to make the system more efficient and effective, the recommendation engine needs to be tested with real data and user feedback. The use of more sophisticated deep learning models, experimentation with different measurement metrics, and improvements in scalability and memory management are suggested topics for future work. In addition, the interface could be made more interactive.

Finally, this project is an example of how deep learning and machine learning can be used in the fashion industry. With the continuous development of technology, further development of modeling and algorithms will make a significant contribution to the industry.

References

[1] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," pp. 1096–1104, 2016.

[4] M. Vasileva, B. Plummer, A. Kovashka, D. P. Kumar, A. Farhadi, and R. Kumar, "Learning type-aware embeddings for fashion compatibility," pp. 390–405, 2018.

[5] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "Learning fashion compatibility with bidirectional lstms," pp. 1078–1086, 2017.

[6] R. He and J. McAuley, "Vbpr: Visual bayesian personalized ranking from implicit feedback," 2016.

[7] T. Muhammed, "Meme kanseri histopatalojik görüntülerinin konvolüsyonal sinir ağları ile sınıflandırılması," *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 31, no. 2, pp. 391–398, 2019.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. [Online]. Available: http://arxiv.org/abs/1512.03385

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, 2017. [Online]. Available: http://arxiv.org/abs/1610.02357

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016. [Online]. Available: http://arxiv.org/abs/1512.00567