WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Fashion Recommendation System Using Deep Learning

Digambar Rathod [a], Akash Tundare [b], Piyush Shelke [c], Shantanu Muneshwar [d], Prof. Sachin L. Dawkhar [e]

[a,b,c,d] Student, Sinhgad College of Engineering Pune, Pune 411046, India
[e] Professor, Sinhgad College of Engineering Pune, Pune 411046, India

A B S T R A C T

Recommendation systems play a crucial role in predicting how an individual will rate an item or social entity. These systems are used for various items, including books, movies, and restaurants, where individual preferences differ. Two primary approaches are commonly used: content-based and collaborative filtering. The former considers item characteristics, while the latter leverages user behaviour to make recommendations. Selecting clothing is essential in an individual's life. To simplify this process, a fashion recommendation system with a virtual trial room has been proposed. Users can upload images, and the system will recommend outfits without the need for physical try-ons. Specifically, our focus is on upper body clothing images featuring human models. The system processes user-uploaded clothing images, identifies outfit types and colours, and suggests the most suitable attire for specific occasions based on the user's existing wardrobe. The virtual trial room allows users to visualize how clothes would look on their bodies without physically trying them on. To achieve this, we explore machine learning and deep learning techniques, including a content-based recommendation system using the ResNet-50 convolutional neural network.

Keywords: Fashion Recommendation System (FRS), Deep learning, Convolutional Neural Network (CNN), k-Nearest Neighbors (kNN)

1. Introduction

The world of fashion and e-commerce is rapidly evolving, with consumers seeking personalized clothing options that resonate with their unique preferences. However, the sheer abundance of choices can overwhelm users, making it challenging to find specific items or styles that truly align with their tastes. Moreover, the absence of physical trial rooms in the online shopping environment exacerbates this problem, leaving shoppers uncertain about how garments will fit and look on them. This disconnect between the digital shopping experience and the tactile nature of clothing has become a significant hurdle for online retailers.

To bridge this gap and enhance user satisfaction, we propose a sophisticated Fashion Recommender System coupled with a cutting-edge Virtual Trial Room feature. In this research our solution leverages image recognition, computer vision, and machine learning technologies to empower users.

2. Literature Survey

2.1 Intelligent Fashion Recommender System: Fuzzy Logic in Personalized Garment Design:

This paper proposes a new intelligent fashion recommender system to select the most relevant garment design scheme for a specific consumer in order to deliver new personalized garment products. This system integrates emotional fashion themes and human perception on personalized body shapes and professional designers' knowledge.

2.2 DeepPose: Human Pose Estimation via Deep Neural Networks:

This work proposes a DNN-based approach for human pose estimation, emphasizing holistic reasoning and simplicity. It employs a cascade of DNN regressors for precise joint localization, showcasing state-of-the-art performance on diverse academic benchmarks with real world images.

2.3 ImageNet Classification with Deep Convolutional Neural Networks:

The paper highlighted the importance of deep architectures in achieving these results, showing that removing any convolutional layer led to inferior performance. They discussed the potential for further improvements through unsupervised pre training and the use of even larger and deeper networks in the future, especially for tasks involving video sequences.

3. Methodology

3.1 Dataset Generation:

To train a robust and accurate fashion recommendation system, we need a large and diverse dataset of clothing images and metadata. We can either collect our own data from various online sources, such as e-commerce websites, social media platforms, fashion blogs, etc., or use existing datasets that are publicly available. Some examples of such datasets are DeepFashion, Fashion-MNIST, Fashion-Gen, and Fashionpedia. We also need to label our data with relevant attributes, such as category, style, color, pattern, occasion, etc., to enable fine-grained recommendations. Additionally, we need to preprocess our data to ensure consistent quality, size, format, and orientation.

So we use a fashion dataset from kaggle.com named Fashion Product Images Dataset. The size of the dataset is nearly 15 GB. It contains images of various clothes and fashion products. So we store that images in a folder and gave the path of that folder in the main code. So the images recommended by the model will be from this folder or so called Dataset.

3.2 Feature Extraction:

In order to identify and extract key characteristics from clothing images, we employed sophisticated methods from the fields of deep learning and computer vision. Convolutional Neural Networks (CNNs) were utilized to learn high-level features that capture the visual properties and distinct qualities of clothing articles. Additionally, techniques like semantic segmentation, pose estimation, and face detection were used to pinpoint and isolate the clothing regions within the images. This process also involved aligning these regions with the user's body shape and dimensions.

For extracting features from the images we used libraries in python named Tensorflow, Keras and used the ResNet-50 pre-trained model. We write a function that takes an image path and the pre-trained model as input. It loads the image, preprocesses it, and extracts features using the ResNet-50 model. The features are flattened and normalized. The normalized feature vector is returned. For each image we extract features using this function and listed that features in a file for building a fashion recommendation system.

3.3 Model Implementation:

To create an effective fashion recommendation system, we rely on machine learning algorithms that learn from user preferences and feedback. Here are the key approaches we can use:

1. ***Collaborative Filtering****:*
    - Collaborative filtering recommends clothing items based on similarities between users' preferences.
    - If a user has liked or purchased certain items, we look for other users with similar tastes and recommend items they have enjoyed.
    - This method leverages collective wisdom and community preferences.
2. ***Content-Based Filtering****:*
    - Content-based filtering recommends clothing items based on their characteristics and features.
    - For instance, if a user uploads an image or provides text describing their desired style, we can recommend similar clothing items.
    - By analyzing the content (such as colour, pattern, fabric, etc.), we match it to existing items in the system.
3. ***Hybrid Filtering****:*
    - Hybrid filtering combines collaborative and content-based approaches.
    - It aims to overcome limitations of each method by leveraging their strengths.
    - For example, we can use collaborative filtering to provide personalized recommendations and content-based filtering to enhance diversity.

So, we collected sample images and pre-processed it for building a CNN (Convolutional Neural Network) model. We processed the images by making dataframes which are the entities from pandas library of python. A CNN model is constructed with three convolutional layers, max-pooling layers, dropout layers, and fully connected layers. The output layer has units corresponding to the number of unique article types. The model is compiled with sparse categorical cross-entropy loss and the Adam optimizer. It is trained using the training data (X_train and Y_train) for 150 epochs. The trained model is saved to a file. Later, the model is loaded back into memory.

3.3.1 Algorithm:

- Step 1: Import necessary libraries

- Step 2: Define dataset path
- Step 3: List contents of the main directory
- Step 4: Locate the "images" subdirectory
- Step 5: List contents of the "images" subdirectory
- Step 6: Collect sample images
- Step 7: Display sample images
- Step 8: Create augmented Dataframe
- Step 9: Build CNN model
- Step 10: Compile and train the model
- Step 11: Save and load the model

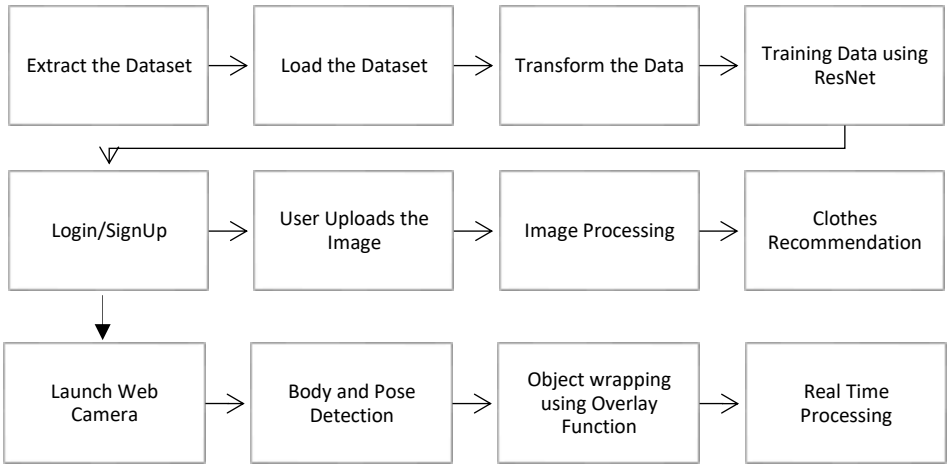
3.4 Main Implementation:

The code begins by importing necessary libraries such as streamlit, os, pickle, numpy, tensorflow, shutil, cv2, and others. It sets up the ResNet50 model (pre-trained on ImageNet) for feature extraction. The feature list and filenames are loaded from pickled files. A custom feature extraction model is created by adding a global max-pooling layer to the ResNet50 base model. The uploaded image is saved and features are extracted from that image. K-Nearest Neighbors (k-NN) is used to recommend similar images based on extracted features. The top five images are displayed which are recommended, using streamlit columns. The user can try that clothes (recommended images) virtually in the virtual trail room, which is the feature of our application built using OpenCV python library. The user is redirected to the virtual trial room if he want to try the clothes.

3.4.1 Algorithm:

- Step 1: Image Upload: Users can upload an image file containing the desired clothing item.
- Step 2: Image Display: The uploaded image is presented for user verification.
- Step 3: Feature Extraction: Key characteristics are extracted from the image using advanced algorithms.
- Step 4: Similar Item Recommendation: Based on the extracted features, similar clothing items are recommended to the user.
- Step 5: Recommendation Display: The top recommended clothing items are presented for user review.
- Step 6: Real-Time Video Capture: The system utilizes a framework like OpenCV to capture video frames from the user's webcam in real-time.
- Step 7: Pose Estimation: Pre-trained models like MediaPipe Pose or OpenPose are employed to identify key body points on the user in real-time.
- Step 8: Clothing Virtual Try-On: Image processing techniques facilitated by OpenCV are used to virtually overlay chosen clothing items onto the user's body in real-time.
- Step 9: Continuous Rendering: The system continuously captures video frames, estimates pose, and overlays clothing in real-time, providing a seamless virtual try-on experience.

Fig. 1 - Design



So we built a Streamlit app using python providing a simple GUI for the user to interact with it. The user can select the image of a cloth and upload it to get the similar cloths which are recommended by the model and then try it if he/she wants in the virtual trial room. The above figure shows the design of our system. We also give the user an option to choose between the genders. So both male and female can try our application for cloth recommendation.

4. Results

The project aims to develop a Fashion Recommendation System using Deep Learning techniques. When the code is executed, it displays a Python GUI interface. Users can upload fashion images, and the system provides personalized fashion recommendations based on those images. Additionally, the system offers a virtual try-on feature, simulating a trial room experience where users can virtually try recommended clothing items. The recommendations are generated in real-time, making the FRS fully functional.

In a comprehensive review, the focus was on fashion recommendation-based articles published over the last decade. These articles explicitly described their frameworks, algorithms, and filtering techniques. The search for relevant articles was conducted using topic-related keywords. Despite this, the article extraction methodology remained unaffected, as the authors included and studied all research papers relevant to the research focus. Future researchers could conduct a systematic literature review on the same topic. Furthermore, reviewing the datasets used in fashion recommendation research articles and evaluating the performance of recommender systems using proposed algorithms would be valuable for future studies.

5. Conclusion

The introduction of the Virtual Trial Room feature enhances user experience by enabling virtual outfit trials. Although recommendations depend on existing wardrobe items, our system successfully bridges the gap for fashion novices. Looking ahead, potential enhancements, such as detecting diverse designs and accommodating various occasions, promise an even more versatile and user-friendly platform, meeting the diverse fashion needs of users in the dynamic world of style. It achieves our goal of providing online shopping system a prominent feature that will help a lot of people who loves shopping online. Also, our Fashion Recommendation System is like a helpful guide, guiding people through the world of fashion even if they aren't experts. It suggests outfits based on what they already own, making sure the style matches their personality. Our system keeps up with fashion trends and grows alongside them, enhancing user’s fashion sense. The Virtual Trial Room feature makes online shopping feel real, allowing users to virtually try on outfits and boosting their confidence. In the future, we plan to make our system even better, understanding different styles and suggesting outfits for various occasions. Our goal is to create a simple and friendly platform that caters to everyone's fashion needs, making the world of style accessible and exciting for all.

Acknowledgement

We owe our deep gratitude to our project guide Prof. S. L. Dawkhar, who took keen interest on our project work and guided us all along, till the completion of our research work by providing all the necessary information for developing a good system. We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of Information Technology Department which helped us in research and design of our project.

References

1. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. NIPS, 1(1):795–825, November 2012. [3] M. F. e. a. Stephan M. Parameterized shape mod.

2. Apparel classification with style. http://people.ee.ethz. ch/˜lbossard/projects/accv12/index.html.

3. Bossard, Lukas, Matthias Dantone, Christian Leistner, Christian Wengert, Till Quack and Luc Van Gool.

4. Frédéric Cordier, WonSook Lee, HyeWon Seo, Nadia Magnenat-Thalmann, "Virtual-Try-On on the Web," pp. 1-11, May 2001. [Online]. Available: https://pdfs.semanticscholar.org/bbc0/39122b37934488c32b0fe713e3ed1a6d36d6.pdf

5. Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, Ping Luo, "Towards Photo Realistic Virtual Try-On by Adaptively Generating↔Preserving Image Content. IEEE international Conference on computer vision.

6. Krizhevsky, Alex, Ilya Sutskever and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." Communications of the ACM 60 (2012): 84 – 90.

7. Mohamed Elleuch, Anis Mezghani, Mariem Khemakhem, Monji Kherallah "Clothing Classification using Deep CNN Architecturebased on Transfer Learning", 2021DOI:10.1007/978-3- 030-49336-3_24.

8. M. F. e. a. Stephan M. Parameterized shape models for clothing. http://www.cs.berkeley.edu/˜pabbeel/papers/ MillerFritzDarrellAbbeel ICRA2011.pdf.

9. Saurabh Gupta, Siddartha Agarwal, Apoorve Dave. "Apparel Classifier and Recommender using Deep Learning." (2015).

10. Shih-En Wei, Varun Ramakrishna, Takeo Kanade, Yaser Sheikh, "Convolutional Pose Machines," pp. 1-9, April 2016. [Online]. Available: https://arxiv.org/pdf/1602.00134.pdf