ML MINI PROJECT REPORT

TEAM MEMBERS
Sideshwar Jb(151)
Siman Fathima(152)
Sneha Narayanan(155)
Sneha Priya M(156)

FASHION PRODUCT RECOMMENDER SYSTEM USING CNN

ABSTRACT

A fashion product recommender system is proposed which will be able to recommend the user to choose appropriate apparel and accessories that suit their preference. The necessity of this system is to reduce the product selection and purchasing time. This will also help to create tailor made outfits as per the personality traits. The system is based on two modules of processes; the first one is to recognize the type of fashion product image and categorize it into a class namely - watch, shirt or pants etc. and the second one is to recommend similar collection of products to motivate users' personal taste and make the most of their wardrobe thereby selecting their outfits based on their personality, profession etc.

MOTIVATION

Online shopping is a form of electronic commerce which allows consumers to directly buy services from a seller over the internet by using a web browser. Given the current pandemic situation, in-store shopping poses a threat to customers as they are at higher risk of getting exposed to the virus. Online shopping has now become inevitable option to purchase apparel and other accessories. Also, a survey by AC Nielson on global online consumers say that fashion products will continue to top the list for planned online purchases in the next six months. The proposed fashion product recommender system becomes a crucial part of all online shopping systems which will enhance customer experience by suggesting similar products that match with their preference.

OBJECTIVE

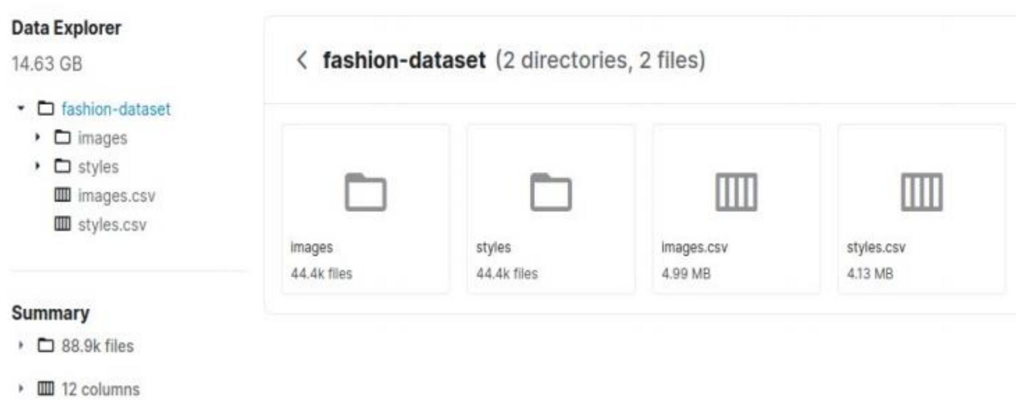
The objective of the proposed fashion product recommender system is to assist the users in selecting and purchasing a product by recommending products that suit the users' preferences. This helps in reducing the product selection and purchasing time. The recommender system is incorporated in two phases: the first phase categorizes the image product and the second phase recommends similar products to motivate users' personal taste and make the most of their wardrobe.

THE DATASET

DATASET

The chosen dataset containing two directories **(images and files each containing 44.4k files)**, and two files **(images.csv with a file size of 4.99MB containing 44446 images and styles.csv with a file size of 4.13MB)** is from Kaggle – https://www.kaggle.com/paramaggarwal/fashion-product-images-datase

The dataset contains professionally shot high resolution product images, in addition to multiple label attributes describing the product which was manually entered while cataloging. Also, descriptive text that comments on the product characteristics is also available.



METHODOLOGY

The proposed fashion product recommender system is implemented in two phases –

Phase 1: Category learning (Classification)

Phase 2: Recommendation

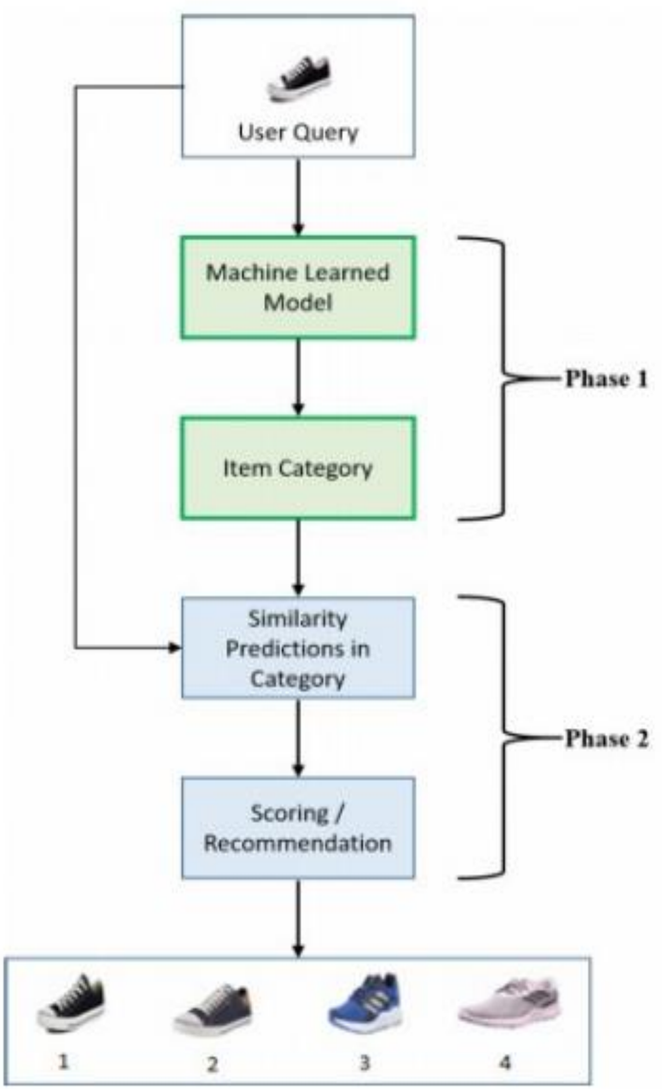
Phase 1: Category Learning (Classification)

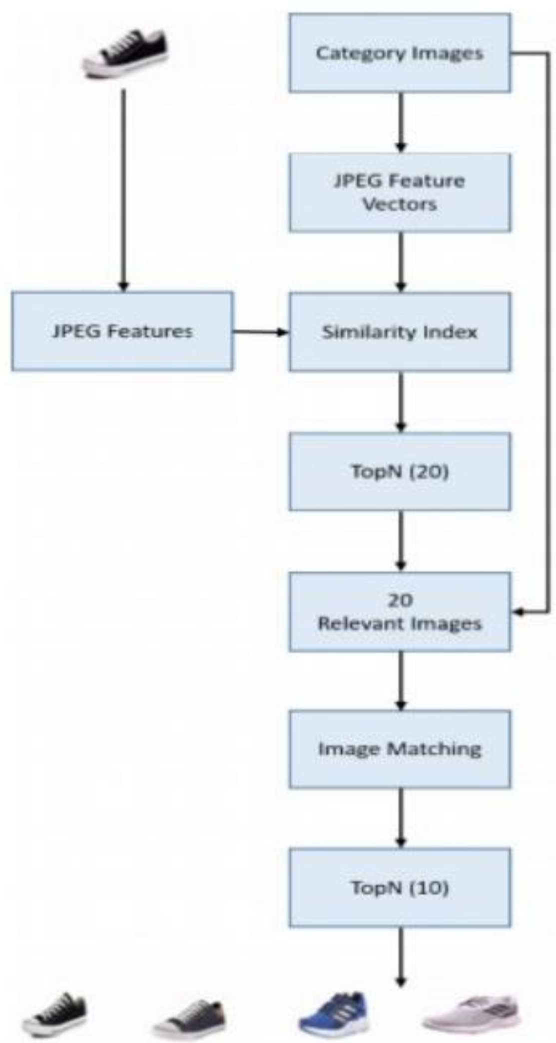
CNN (Convolution Neural Network) is used for product image recognition. A CNN can take an input image, assign learnable weights and biases to various aspects of the image and be able to differentiate from one another. CNN integrates the feature extraction and classification steps, and requires minimal preprocessing and feature extraction efforts. Hence, CNN is best suited for product image classification and recognition.

Phase 2: Recommendation

After the model classifies the query image into a category like a watch, or shirt etc., similar product images are found using Euclidean distance as a metric. The system calculated the Euclidean distance between the query image and every other image contained in the class. The product images are sorted based on similarity score or least Euclidian distance. The ten best similar product images are, then, displayed as recommendations.

METHODOLOGY





ARCHITECTURE

CNN ARCHITECTURE:

A basic convolutional neural network comprises three components, namely, the convolutional layer, the pooling layer and the output layer. The pooling layer is optional sometimes. It consists of the input layer, multiple hidden layers (repetitions of convolutional, normalization, pooling) and a fully connected and an output layer. In a CNN model, the input image is considered as a collection of small sub-regions called the “receptive fields”. A mathematical operation of the convolution is applied on the input layer, which emulates the response to the next layer. The response is basically a visual stimulus.

INPUT LAYER:

The input is stored and loaded here. It also describes the height, width and number of channels of the input image (RGB information).

HIDDEN LAYER:

They perform a feature extraction process where a series of convolution, pooling, and activation functions are used. The distinct features of the product image are identified here. This layer is the backbone of CNN.

CONVOLUTION LAYER:

It is the first layer placed above the input image. It is used for extracting features of an image. The main contributors of the convolutional layer are receptive field, stride, dilation and padding.

POOLING LAYER:

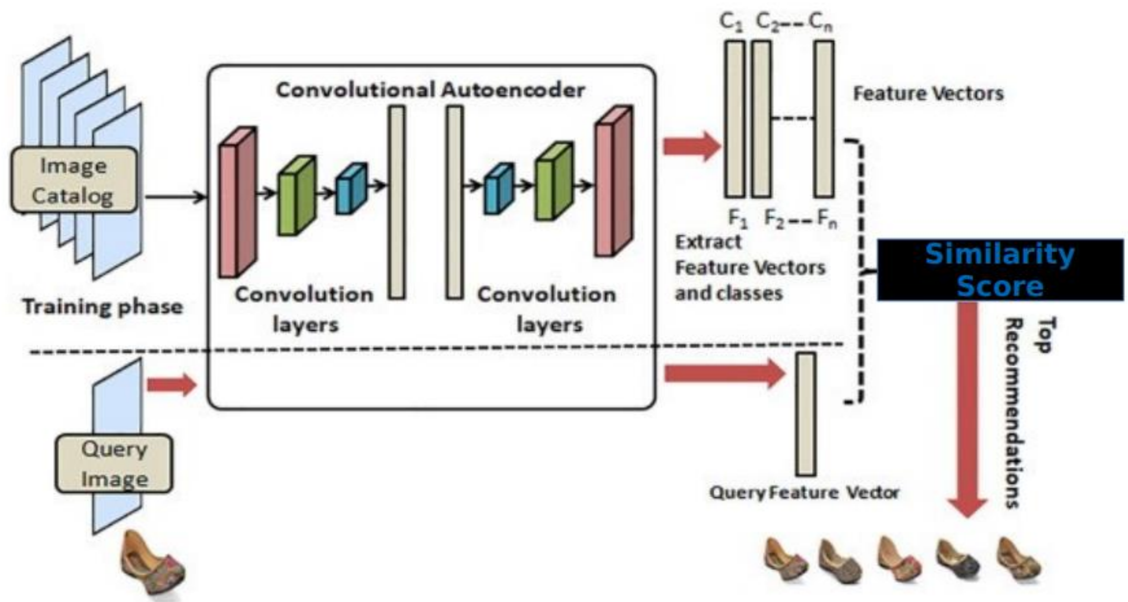
It is placed between two convolutional layers to reduce input dimensionality which in turn reduces the computational complexity. Pooling allows the selected values to be passed to the next layer while leaving the unnecessary values behind. The pooling layer also helps in feature selection and in controlling overfitting. The common types of pooling operations are max-pooling and avg-pooling (where max and avg represent maxima and average, respectively).

ACTIVATION LAYER:

It contains activation functions to introduce non linearity in the system. The sigmoid function, rectified linear unit (ReLu) and Softmax are some activation functions that can be used. The activation function used in the present work is the non-linear rectified linear unit (ReLu) function, which has output 0 for input less than 0 and raw output otherwise.

CLASSIFICATION LAYER:

It is the last layer in the CNN architecture. It is a fully connected feed forward network, mainly adopted as a classifier, where the neurons in the fully connected layers are connected to all neurons of the previous layer. This layer calculates predicted classes by identifying the input image, which is done by combining all the features learned by previous layers. In this project, the classification layer uses the 'softmax' activation function for classifying the gestures of the input image from the previous layers into various classes based on training data.



EVALUATION METRICS

EVALUATION OF PHASE 1: CLASSIFICATION

1. ACCURACY:

Accuracy is defined as the ratio of Number of correct predictions to the total number of predictions.

$$\boxed{\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

LOSS - SPARSE CATEGORICAL CROSS ENTROPY:

Each predicted class probability is compared to the actual class desired output 0 or 1 and a score/loss is calculated based on how far it is from the actual expected value.

The penalty is logarithmic in nature. In sparse categorical cross-entropy, truth labels are integer encoded, for example, [1], [2] and [3] for 3-class problem.

Cross-entropy loss is used when adjusting model weights during training. The aim is to minimize the loss, i.e, the smaller the loss the better the model.

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log(p_i), \quad \text{for n classes,}$$

where t_i is the truth label and p_i is the Softmax probability for the i^{th} class.

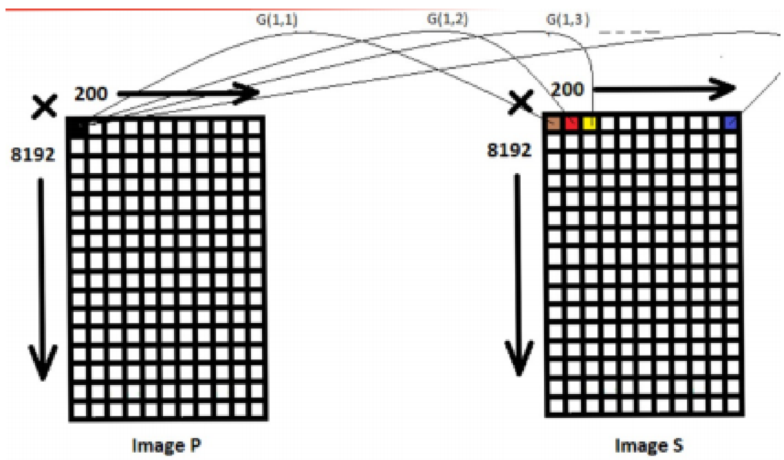
EVALUATION OF PHASE 2: RECOMMENDATION

The evaluation of Phase 2, is not straight forward as that of Phase 1. This is due to several reasons. First, there is no baseline for comparison. Second, even if the data is labeled for Phase 2 of the recommendation step, it is still labeled by humans, and thus the results are subjective. Therefore, we adopted the vector based euclidean distance for evaluation of Phase 2.

EUCLIDEAN DISTANCE BETWEEN IMAGES:

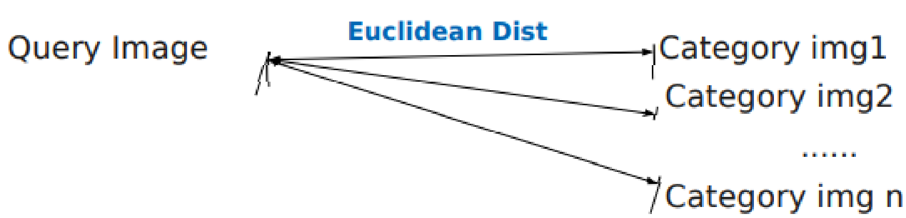
D(x,y)= sqrt((I1G-I2G)^2+(I1B-I2B)^2+(I1R-I2R)^2) where I1, I2 are RGB images, I1G is the green channel of the I1 RGB image, I1B is the blue channel and I1R the red channel.

For Grayscale Images, D(x,y)= sqrt((I1-I2)^2)



Euclidean distance is used as the similarity and scaled between 0 and 1, where 0 represents 0% similarity, and 1 means 100% similarity.

All the other values between 0 and 1 indicate the corresponding percentage of similarity.



The similarity scores are then sorted in the ascending order, and 10 images with the MOST SIMILARITY SCORE or the LEAST EUCLIDEAN DISTANCE are selected as the possible candidates for the recommendations.

IMPLEMENTATION: STEPS INVOLVED

MODULES USED:

- OpenCV
- Tensor Flow
- Keras
- Sklearn
- Matplotlib
- Numpy
- Pandas

STEPS INVOLVED:

- Importing the libraries
- Prepare, Train and Test Data
- Train Test Split
- CNN Model
- Load the saved model and pass the test data
- Find the class/ category of a given product image
- Find similar images
- Calculate Euclidean distance between query image and similar images
- Sort the images in ascending order based on least Euclidean distance
- Display the ten best similar images as recommendations

EXAMPLE RETRIEVALS

CONCLUSION

Fashion Product Recommendation using images has made tremendous progress over the years. Online shopping websites are hugely benefitted by this. As research in this field continues, more and more interesting methods have come to light. Work once started using text-based methods, turned to visual methods with image processing and use of neural networks, convolutional neural networks and now transfer learning with deep neural networks. Hence, it can be seen that there is a common theme in the recent research carried out in the field of clothing recommendation. This theme analyzes images, finding out features in the images and classifying pieces of clothing in the image. It can be understood that this methodology works for most systems. The system proposed by us recommends or guides the user to choose appropriate clothes or products suiting their choice. The guidelines for selection of their respective products are based on Euclidean distance. A fashion product recommender system is hence implemented using a convolutional neural network model for classifying the data-set of fashion product images and recommending the best similar products using machine learning libraries with optimal accuracy and efficient performance through suitable parameters.