April 2023

The FASHION Visual Search using Deep Learning Approach

Smita V. Bhoir
Department of Computer Engineering, K.J. Somaiya College of Engineering, Ramrao Adik Institute of Technology, Mumbai, Maharashtra, India, smitapatilbe@gmail.com

Sunita R. Patil
2Department of Computer Engineering, K.J. Somaiya Institute of Engineering & Information Technology, Mumbai, Maharashtra, India, spatil@somaiya.edu

The FASHION Visual Search using Deep Learning Approach

Smita Bhoir[1, a *], Sunita Patil[2,b]

[1]Department of Computer Engineering, K.J. Somaiya College of Engineering, Mumbai, Maharashtra, India

[2]Department of Computer Engineering, K.J. Somaiya Institute of Engineering & Information Technology, Mumbai, Maharashtra, India

[a]smitapatilbe@gmail.com, [b]spatil@somaiya.edu

Keywords: E-commerce; Visual Search; Deep Fashion Convolution Neural Network; Fashion Classification;

Abstract. In recent years, the World Wide Web (WWW) has established itself as a popular source of information. Using an effective approach to investigate the vast amount of information available on the internet is essential if we are to make the most of the resources available. Visual data cannot be indexed using text-based indexing algorithms because it is significantly larger and more complex than text. Content-Based Image Retrieval, as a result, has gained widespread attention among the scientific community (CBIR). Input into a CBIR system that is dependent on visible features of the user's input image at a low level is difficult for the user to formulate, especially when the system is reliant on visible features at a low level because it is difficult for the user to formulate. In addition, the system does not produce adequate results. To improve task performance, the CBIR system heavily relies on research into effective feature representations and appropriate similarity measures, both of which are currently being conducted. In particular, the semantic chasm that exists between low-level pixels in images and high-level semantics as interpreted by humans has been identified as the root cause of the issue. There are two potentially difficult issues that the e-commerce industry is currently dealing with, and the study at hand addresses them. First, handling manual labeling of products as well as second uploading product photographs to the platform for sale are two issues that merchants must contend with. Consequently, it does not appear in the search results as a result of misclassifications. Moreover, customers who don't know the exact keywords but only have a general idea of what they want to buy may encounter a bottleneck when placing their orders. By allowing buyers to click on a picture of an object and search for related products without having to type anything in, an image-based search algorithm has the potential to unlock the full potential of e-commerce and allow it to reach its full potential. Inspired by the current success of deep learning methods for computer vision applications, we set out to test a cutting-edge deep learning method known as the Convolutional Neural Network (CNN) for investigating feature representations and similarity measures. We were motivated to do so by the current success of deep learning methods for computer vision applications (CV). According to the experimental results presented in this study, a deep machine learning approach can be used to address these issues effectively. In this study, a proposed Deep Fashion Convolution Neural Network (DFCNN) model that takes advantage of transfer learning features is used to classify fashion products and predict their performance. The experimental results for image-based search reveal improved performance for the performance parameters that were evaluated.

1. Introduction

The fashion industry is expanding at a rapid pace in both the United States and internationally. In comparison to 2020, the domestic fashion market is predicted to grow by 8.8 percent in 2024, reaching USD (United States dollars) $26.288 million in value. [1][2]. The apparel and fashion online market are expected to reach USD 113 million in 2018, and it is growing at a rate of 18 percent per year, with mobile shopping accounting for approximately 40% of the total.

The fashion industry also includes accessories such as purses, shoes, and jewellery, which are categorized as such. The automation, development, and digitization of manufacturing and distribution have resulted in a new paradigm for the fashion industry. As the fashion industry becomes more digital, the system evaluates customer fashion desires and attempts to match those desires as quickly as possible. As a result, digital technology in the fashion industry is attracting the attention of a diverse range of consumers who, as a result of the shorter manufacturing cycle, have access to a greater variety of products. Customers' use of smart gadgets has increased to more than 10 million as a result of the introduction of a new e-commerce platform [3]. Traditional garment sales channels such as brick-and-mortar stores have seen sales decline, whereas non-store retailers such as online and mobile merchants have seen sales increase at the fastest rate. When compared to previous years, the online fashion market increased by 16 percent per year from 2011 to 2016, and from January to September 2017, the market increased by 20 percent over the previous year [4]. Eventually, it is expected to become a major channel in the fashion industry in the coming years.

Since it provides a hassle-free shopping experience and delivery to the user, e-commerce has transformed the world of consumerism and triggered an increase in demand for goods. Because there is a greater selection of fashion products available on e-commerce platforms than in traditional retail stores, shopping for fashion apparel on e-commerce platforms differs significantly from shopping in traditional retail stores. As a result, a system that more efficiently assists consumers in searching for the goods they desire and recommending the desired product is becoming increasingly important to businesses. Therefore, if you want to communicate effectively in the fashion industry, you cannot rely solely on verbal communication because visual cues and design elements are so important. Although the existing fashion retrieval system employs a text-based retrieval approach that is based on product attribute information (product name, category, brand name, and so on), the system is not without its flaws (product name, category, brand name, etc.). A text-based search strategy has limitations when it comes to providing appropriate search results in the fashion industry, which includes a significant design component. Numerous "shopping how" and "smart lens" systems have been developed in recent years, but they have all failed to deliver good or noteworthy results when searching for products using fashion images as a starting point. Users must either find an image of the goods online or take their photographs to use these systems.

Two major issues affect the industry, which is discussed from both the seller's and buyer's perspectives in this study [5]. An e-commerce site requires sellers to upload photos of their products as well as relevant labels to the product's description for the product to be sold. Because of the involvement of humans, this process is prone to errors. Product misclassification can cause products to be missed in search results, resulting in lower or no sales for the company that makes them. By using machine learning models, the photographs can be classified with high accuracy, which in turn motivates the vendors to categorize them correctly. In addition, a customer's demand may be delayed because he does not understand the correct terminology [5]. When a customer is shopping on an e-commerce website, he or she typically enters the product's keywords into the search bar. Its search algorithm compares the keywords entered by the user with the product labels stored in its database, and it returns results that are relevant to the user's inquiry. When the user locates the product he or she is looking for, the user orders it directly from the search results page. Using a text-based search necessitates that the consumer has a thorough understanding of the product and is aware of the terms that should be entered into the search toolbar. It's important to note that this isn't always the case. We come into contact with a wide variety of things in our daily lives that we are completely unaware of. Occasionally, we are unable to conduct a product search on the e-commerce website due to technical difficulties. By employing visual search techniques, you can overcome this difficulty. Shoppers who conduct visual searches look for products that include images or other visual cues rather than

products that contain keywords. Customers can search for similar products by simply taking a picture of what they want and uploading it to the visual search engine.

Any visual search algorithm can benefit from machine learning models, which can be used to learn attributes about new photographs and search for comparable products. To provide more relevant search results, additional images with similar features will be added to the visual search engine once the target image has been uploaded to make the search results more relevant as well. It is possible to generate images' latent properties using autoencoders and other visual search methods, for example. To further enhance the retrieval of image embedding features, deep neural network models that have already been trained can be applied to the problem [4][6][7].

A large number of images from all over the Internet are available to users through social media and respectable smartphones, which means that users have access to a large number of images from all over the Internet at once. In these circumstances, the ability to search for, filter, and organize photos becomes increasingly important. Manually searching for the images you want is an option if you have a small number of images. As the number of things grows, this becomes impossible to accomplish. To keep up with this rapid expansion, it is necessary to develop picture retrieval systems that are capable of operating across a wide range of platforms. It is our goal to develop a retrieval system that is accurate in handling and querying a database of photographs [8][9].

To summarize, this paper will pursue three broad objectives:

Image Labeling: In E-commerce, while purchasing products appropriate labeling of products is important. Many images over the web are unlabeled [10].

Image Classification: To design and train different neural network models to learn from large sets of images of products from an e-commerce website [11].

Image Search: To use autoencoders and cosine similarity to identify similar images [12][13].

2. Transfer Learning

Deep learning [14] is a machine learning method that belongs to a large family of machine learning methods. [15] Deep learning classifiers, as opposed to traditional neural network classifiers, develop classifiers with numerous hidden layers to identify the salient low-level features of a picture, as opposed traditional neural network classifiers. Concerning Deep Learning, a technique known as transfer learning makes use of an artificial neural network's ability to use features learned from a previous problems to solve a new problem within the same domain. It is advantageous to learn through transfer[16][17][18] for a variety of reasons.

- First, it saves computational time by reusing information from a previous training process rather than starting from scratch with a new model, rather than starting from scratch with a new model [16].
- The second advantage is that it builds on the knowledge and experience gained through the use of previous models [17].
- The third point to mention is that when the new training dataset is small, transfer learning is extremely beneficial [18].

Transfer learning has the potential to benefit a wide range of applications, including computer vision, audio categorization, and natural language processing. Many attempts have been made to automate the classification of images, either to speed up the process or to improve accuracy. To solve the picture categorization problem, the convolutional neural network (CNN) was developed as one of the first approaches. Because of the work of Krizhevsky et al.[19], CNN was able to outperform all other methods for solving the picture classification problem by the year 2012. They achieved state-of-the-art performance in the ImageNet Large Scale Visual

Recognition Challenge (ILSVRC) competition, outperforming other commonly used machine learning algorithms and achieving superior results. In all of these scenarios, training the weights of the deep network from scratch requires a significant amount of time and a large amount of data (hundreds of thousands of images). The presence of these constraints makes deep learning algorithms extremely difficult to implement in the domain of image data, where there are frequently only a few images available [20]. A significant amount of time and knowledge is required to annotate fashion images; this is where transfer learning may be beneficial to students. Utilizing an architecture that has already been trained eliminates the need to start from scratch. According to several studies, CNN has been used to identify fashion images, either through transfer learning or the introduction of novel architectural structures. According to [24], deep CNN-based fashion image categorization models are presented in [21–35] of the literature reviewed. Some critical open questions are presented here to improve the utilization of transfer learning in the fashion industry.

3. CNN Architectures used in Transfer Learning

The main CNN architectures used in transfer learning are discussed in detail in this section. According to [36], the rise of deep learning in image classification began in 2012 with the introduction of AlexNet [14], which included the ReLU activation layer as well as other deep learning techniques. It was discovered that using a CNN to classify images improved its accuracy while also eliminating the need to manually feature engineer each image. Following the introduction of AlexNet, a slew of new architectures, including VGG16, VGG19, ResNet, GoogLeNet, DenseNet, and Xception, were introduced, each with additional features to more effectively classify images.

A. VGG Network Architecture

The Visual Geometry Group at Oxford University introduced two new structures in 2014, which were designated VGG16 [37] and VGG19 [37]. The VGG16 architecture is comprised of five convolution blocks and three thick layers, with a total of 138,355,752 parameters in the architecture. Each block contains many convolutional layers, followed by a max pool layer, which helps to reduce the block output size and remove noise. In total, six convolutional layers are used in the final three blocks, with two being used in each of the first and second blocks. One stride in the kernel is used by this network, and it is stride 1. Last but not least, an additional layer was created that would transform each block into a single-dimensional object that could then be inserted into the various levels that were connected. Total connectivity is achieved by connecting 4096 neurons in the first two layers and 1000 neurons in the third and final layer in the third and final layer. In addition to the fully linked layers, a softmax layer is added to ensure that an output probability summation is a positive number. To emphasize the difference between the two, VGG19 has 19 convolution layers, whereas VGG16 only has 16. As a result of the addition of layers, the number of parameters increases from 138,357,544 to 143,667,240. In the authors' opinion, the addition of additional layers increases the system's robustness and ability to learn complex architectures.

The sequential blocks of this network's sequential convolutional layers allow for a significant reduction in the amount of spatial information that must be stored in the network. The majority of the time, this network suffers from a lack of feature extraction weights, which is a significant flaw in the overall design. A large number of additional variables must be taken into consideration.

B. ResNet Network Architecture

ResNet, which is an abbreviation for residual network, was invented by He et al. [38] in 2015. There are a total of 25,000,000 variables in use. When comparing ResNet to other

architectures, you'll notice that it has a single connection between the convolutional layers and the ReLU activation layer, which is referred to as a "residual connection." This connection is what makes ResNet unique among the rest. It is ensured by the residual connection that weights learned from preceding layers do not disappear during the backpropagation of the network. The three iterations of this network are designated as ResNet50, ResNet101, and ResNet152 (each with a different number of layers). The use of residual connections in this network has several advantages, not the least of which is that it allows for the use of multiple levels. Furthermore, increasing the depth of the network rather than the width of the network makes it easier to reduce the number of unnecessary parameters. The fact that the summation that occurs in each residual block keeps the filter size constant is one of the main disadvantages. Furthermore, to successfully train this network, large datasets are required, resulting in a time-consuming training phase that is expensive in terms of computing resources.

C. GoogLeNet Network Architecture

According to Google researchers, the GoogLeNet network, also known as the IncpetionV1 architecture, was first implemented in 2014 [39]. Following the success of InceptionV1, the writers went on to create sequels, including InceptionV2 and InceptionV3. To capture different aspects of images, GoogleLeNet's fundamental idea is to use multiple convolution layers in the same block to go deeper and wider; these blocks are referred to as Inception blocks in the GoogLeNet architecture. In the GoogLeNet designs, InceptionV1 and InceptionV3 are the two most frequently encountered. A total of six convolution layers are used in the InceptionV1 inception blocks, while a total of seven convolution layers are used in the InceptionV3 inception blocks (see figure). This network's inception module enables the convolution layers to be used in parallel, allowing it to collect images with different aspect ratios at the same time as it collects images with the same aspect ratio. The most significant weakness of this network is the amount of computing effort required to train it, which is a result of the network's extensive layer structure.

D. AlexNet Network Architecture

As a starting point, CNN selected AlexNet [40] as the architecture for its participation in the 2012 ImageNet competition. The AlexNet architecture is made up of five convolution layers and three dense layers, for a total of nine layers. In AlexNet, two novel features were introduced: reLU activation (instead of sigmoid activation) and dropout, both of which were intended to combat the overfitting that can occur as a result of the deep architecture's deep learning. The main advantage of this network is that it is computationally efficient when compared to other networks, which is particularly important during the training stage. A common misconception is that because of the depth of the network, fine details in images cannot be detected.

E. DenseNet Network Architecture

DenseNet architecture [41] [42]refers to convolutional networks that are densely connected and have a large number of connections. Thick blocks, rather than residual connections, were recommended as a replacement for ResNet by the authors. Convolution layers are added to the dense block progressively, but each layer has a direct relationship to the layers that come after it. All of the information from the layers that came before it is fed into each subsequent layer of convolution. The DenseNet contains a total of 8,062,504 variables. The fact that all layers are connected prevents information loss, which is a significant advantage of this network (especially the deep layers). The following are the most significant disadvantages: it needs a large amount of data to work with during the training phase if you want to achieve good results.

F. Xception Network Architecture

Architect Chollet was inspired by the InceptionV3 architecture when he created the Xception (which stands for extreme inception)[43][44]. Xception's design makes use of two key concepts: convolution in-depth and pointwise separability. These concepts are used to replace the inception module, which was previously present. In this 71-layer network, there are a total of 22.9 million parameters. Comparing this network to other deep networks, its main advantage is that it has a deep architecture but only a small number of parameters. This is an important distinction. The network requires a large amount of data from a variety of sources to train all of the network's parameters properly.

To better understand the significance of transfer learning, experimental work is carried out to train models such as the VGG16, VGG19, ResNet, GoogLeNet, DenseNet, and Xception, among others. Table I presents a summary of the results obtained using the number of parameters and accuracy of the parameters over the ImageNet dataset. The accuracy of a classification system is determined by dividing the number of correctly classified observations by the total number of observations. Accuracy over predicted labels y is measured as top-k accuracy, with the top 5 accuracies representing accuracy over five classes, and the top 1 accuracy representing accuracy for a single-class classification.

Table I: Top 5 accuracy, top 1 accuracy, and the number of parameters of AlexNet, VGG, Inception, and ResNet in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) challenge.

Architecture	**Number of Parameters**	**Top-5 Accuracy**	**Top-10 Accuracy**
Xception	22,910,480	84.60%	63.30%
VGG-19	138,357,544	91.90%	74.40%
GoogLeNet	23,000,000	92.2%	74.80%
ResNet-152	25,000,000	94.29%	78.57%
DenseNet	8,062,504	93.34%	76.39%
AlexNet	62,378,344	94.50%	79.00%

4. Proposed Work and Methodology

The research proposal uses structure, the framework to work on the technology. Our approach incorporates three modules. Background removal and extraction of valuable features from the image are the first two modules in the process of image processing. After that, the deep CNN model is trained to examine field-specific feature representations. Finally, the third module uses the features acquired from the trained model to find images that are comparable to the query image. The algorithm depicts the general workflow of the project.

Algorithm:

Step 1: Import the necessary libraries to process the data like (NumPy, pandas, os, OpenCV, matplotlib, etc.)

Step 2: Loading Dataset

Step 3: Process CSV file

Step 4: Pre-process the data by applying data pre-processing techniques

Step 5: Split the dataset

Step 6: Train the model

Step 7: Evaluate performance of model

Authentic fashion images from the most popular Indian e-commerce web application, myntra.com, are used in this work. Because the Fashion Product Images (Small) dataset is readily available on Kaggle, there is no need to scrape images from the website to complete the task. It also has the advantage of having color images, which makes it more similar to a real-world business problem than other types of information. Google Collaboratory, which imports Kaggle data seamlessly, provides free GPU/TPU resources for convolutional neural network (CNN) algorithms for image classification. These algorithms can be used to classify images. Python modules such as TensorFlow and others are already installed, allowing for quick and simple CNN image modeling in Python. During the cataloging process, the dataset includes a variety of label attributes that were manually entered to describe each object, as well as high-quality images taken of the objects themselves. Because the low-resolution images present several computational and run-time issues, we chose the smaller image collection, which contains the same set of images but at a lower resolution, as an alternative.

Step 1: Import Libraries

Import the necessary libraries to process the data like (NumPy, pandas, os, OpenCV, matplotlib, etc.)

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
import cv2
import matplotlib.pyplot as plt
```

Step 2: Loading Dataset

There are 44,424 color channels in the CSV file. Styles.csv can be used to map an image's metadata to its numeric id. Images IDs and metadata IDs were merged and we found 44,419 instances where they match. These photo sets are being considered for use in modeling. Gender, Master Category, Sub-Category, Article Type, Season, etc. are all completely labeled in the dataset. We use three schemes for product classification namely - a concatenated Gender and Master category, Sub Category, and Article Type; as they are clearly labeled without any missing values. Load the dataset from the given directory path for further processing and display some samples of images from the dataset.

Step 3: Process CSV File

Information about the product is contained in a CSV file (id, gender, masterCategory, subCategory, articleType, color, and so on). As demonstrated in Figure 1, Figure 2, Figure 3, and Figure 4, we read the CSV file and count the number of items based on several categories such (masterCategory, subCategory, articleType) and produce the bar graph.

Step 4:Pre-process the data

a) NumPy array Conversion

- Images are converted into NumPy Array in Height, Width, Channel format.
- The main advantage of NumPy array Conversion, e.g., You can load the images into a NumPy array so that you can save computation time. Images can be easily loaded.

b) Generate pixelate form

- Convert NumPy array back to Image format in pixelated form.

c) Image Resize

- Image resizing refers to image scaling. This tends to minimize no. of pixels from the image. For example, that has other effects; for example, it can minimize the training time of neural network as more is no. of pixels in the image more is no. of input nodes that improve the model's complexity.

d) Data Filtration

- Limitations- Existing research papers surveyed considered categories of products having maximum images, neglected categories with less number of products [45]
- To give scope for searching newly available products in a category with less number of products proposed model does not filter any class or images with less number of images.
- Data Filtering is one of the most frequent data manipulation operations. We have used to filter the images based on some conditions (ratio of the images, images that are not present in the dataset), during image pre-processing, images are automatically rejected [46].

e) Gray Scaling

- Gray scaling is the conversion process of an image from other color spaces, for example, RGB, CMYK, HSV, etc. Dimension reduction and Reduction in model complexity are benefits of gray scaling [47][48].

	id	gender	masterCategory	subCategory	articleType	baseColour	season	year	usage	productDisplayName
0	15970	Men	Apparel	Topwear	Shirts	Navy Blue	Fall	2011.0	Casual	Turtle Check Men Navy Blue Shirt
1	39386	Men	Apparel	Bottomwear	Jeans	Blue	Summer	2012.0	Casual	Peter England Men Party Blue Jeans
2	59263	Women	Accessories	Watches	Watches	Silver	Winter	2016.0	Casual	Titan Women Silver Watch
3	21379	Men	Apparel	Bottomwear	Track Pants	Black	Fall	2011.0	Casual	Manchester United Men Solid Black Track Pants
4	53759	Men	Apparel	Topwear	Tshirts	Grey	Summer	2012.0	Casual	Puma Men Grey T-shirt
5	1855	Men	Apparel	Topwear	Tshirts	Grey	Summer	2011.0	Casual	Inkfruit Mens Chain Reaction T-shirt
6	30805	Men	Apparel	Topwear	Shirts	Green	Summer	2012.0	Ethnic	Fabindia Men Striped Green Shirt
7	26960	Women	Apparel	Topwear	Shirts	Purple	Summer	2012.0	Casual	Jealous 21 Women Purple Shirt
8	29114	Men	Accessories	Socks	Socks	Navy Blue	Summer	2012.0	Casual	Puma Men Pack of 3 Socks
9	30039	Men	Accessories	Watches	Watches	Black	Winter	2016.0	Casual	Skagen Men Black Watch

Figure 1: Sample Input dataset after loading into the model

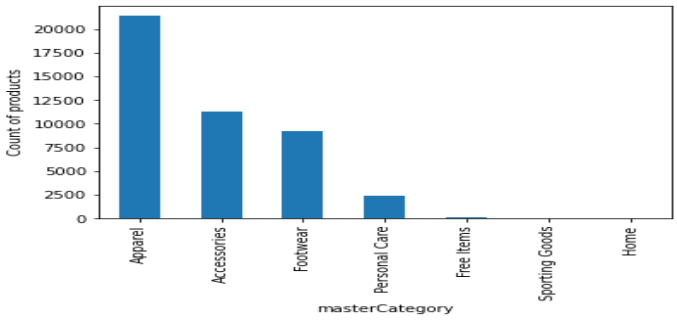


Figure 2: Count the Total no of images by masterCategory

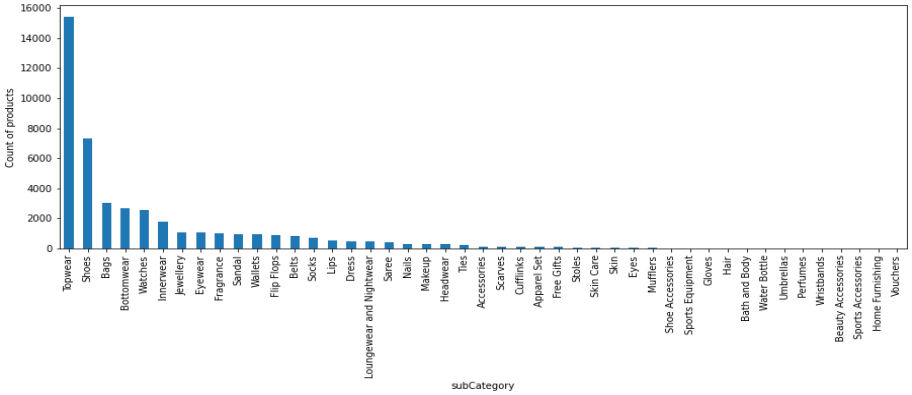


Figure 3: Count the Total no of images by SubCategory

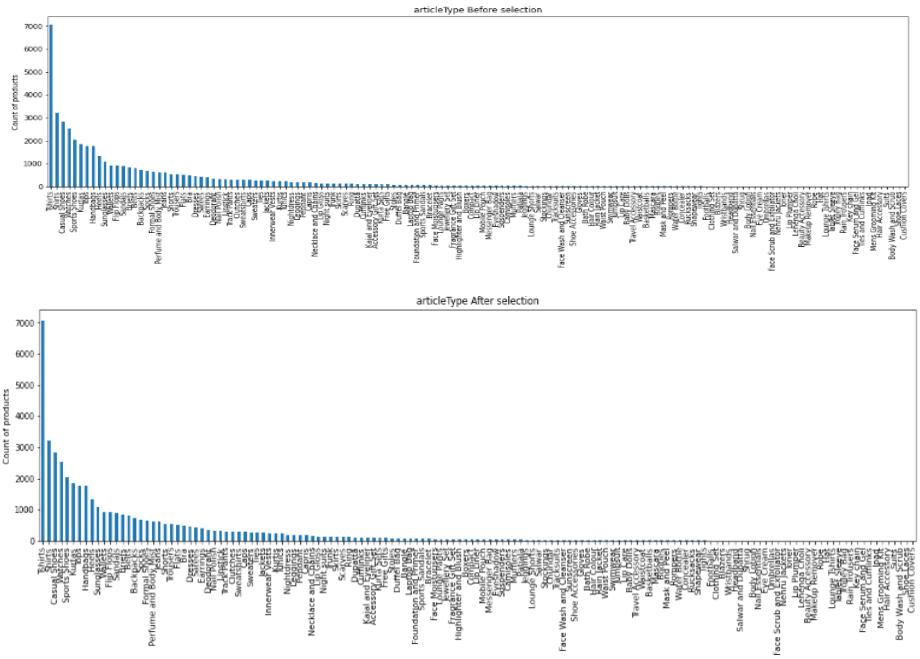


Figure 4: Count the Total no of images by ArticleTypeCategory

f) Auto Labelling

- To allow automatic labeling (assign unique values), the model must be enabled to understand what is depicted in the picture when information is being processed. It has to be trained to know which tag should be attached to each data unit based on article type.

g) Normalization

- It is a method often used in machine learning data preparation. The objective of normalization is to modify the numerical column values of the dataset to a similar scale without distorting the values or missing information.

h) Creating a List with Unique Value

- Created a list with unique values and added a column with their specified element position. Store the Labeled dataset for training the model.
- We create an augmented data frame with a file name and also mapping with product id.
- We also perform one of the important tasks called labeling, we assign the unique id to each article Type.
- Gray Scaling is the process to Convert the colored images into grayscale images.

```
# assign unique ID by article type
unique_types = augmentedDataframe['type'].unique().tolist()
total_class = len(unique_types)
print("Total no. of classes : ",total_class)
print(unique_types)
print(unique_types[0])
print(unique_types.index(unique_types[0]))
augmentedDataframe['number_types'] = augmentedDataframe['type'].app
ly(lambda x: unique_types.index(x) if x in unique_types else 0)
augmentedDataframe.head(10)
```

Normalization is a technique that is commonly used in data processing for machine learning. The goal of normalization is to transform the values of the dataset's numeric columns to comparable scales without distorting ranges of values or losing information. Figure 5 shows sample output images.

```
X = np.array(X).reshape(-1, 227, 227,1)
Y = np.array(Y)

#Normalizing the images
X = X/255.0
Y = Y.reshape(len(X),)
```

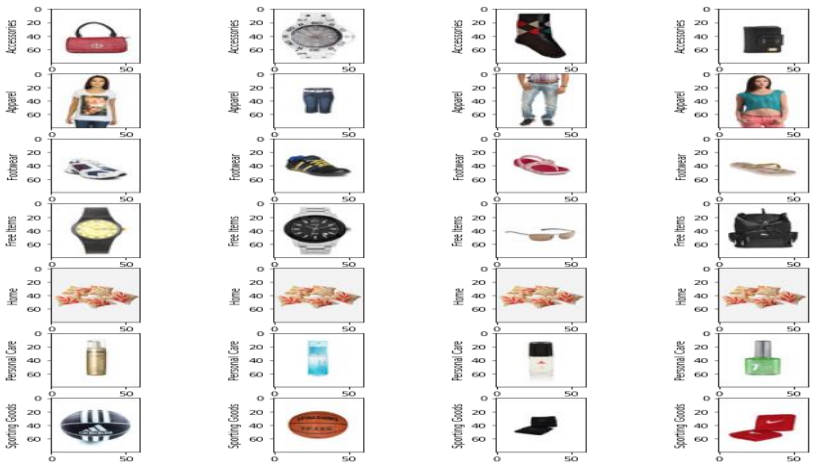


Figure 5: Sample output images

Step 5: Designing Proposed Deep Fashion Convolution Neural Network (DFCNN)

The proposed layered architecture of Deep Fashion Net is given in Figure 6. The design has 8 layers: 5 convolutional layers and 3 fully linked layers.

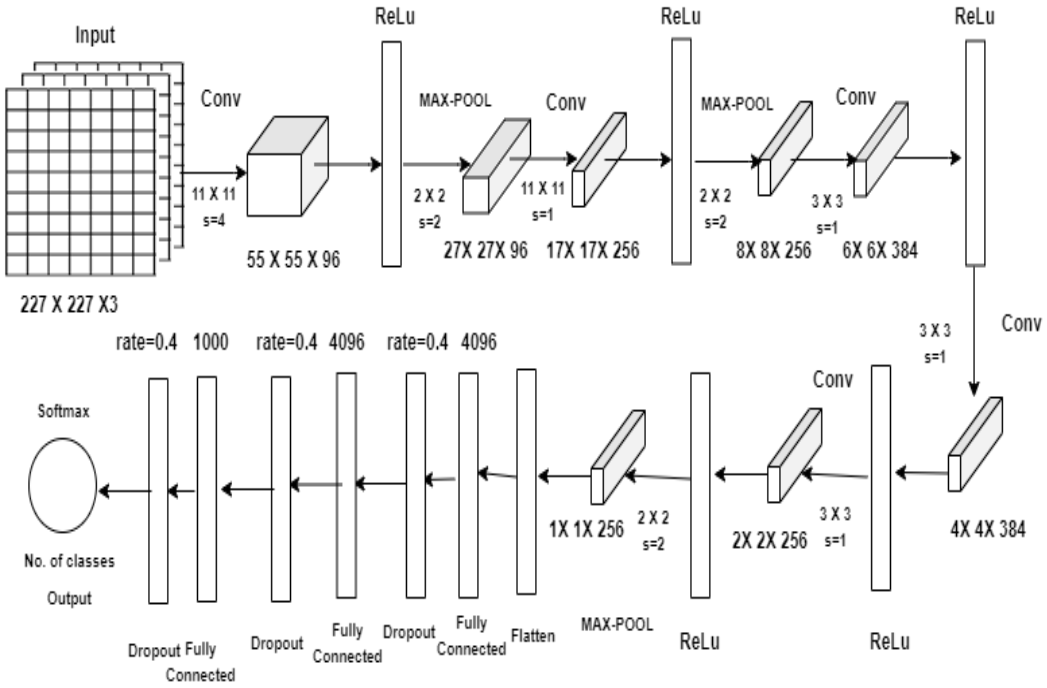


Figure 6: Layered architecture of proposed Deep Fashion Convolution Neural Network (DFCNN)

- **Working details of Deep Fashion Net Model**

The proposed DFCNN (Deep Fashion Convolutional Neural Network) is an 8-layer architecture that comprises five convolutional layers, and 3 dense layers. Input to this model is images of size 227X227X3. Figure 6 shows Layer-wise working details of Proposed DFCNN Model. The first layer of max-pooling, size 2*2, and step 2 are now available in the following. The resultant feature map in size 27X27X96 is then provided. After this, the second convolution operation is applied. We have 256 filters of this type in the filter size, 11 * 11. The step is 1 and ReLu is used again to activate the function. Now the output size we get is 17X17X256. Next, we have a second 2*2 size and step 2 max-pooling layer. Then we have 8X8X256 as the output map. Now we use the third convolution procedure with 384 filters of 3*3 size. The step is one. Again, ReLu is used for the activation function. The output feature map is in a 6X6X384 shape.The fourth convolution operation is then applied with 384 filters 3*3 in size. The step is one. Again, ReLu is used for the activation function. The output feature map is 4X4X384 in shape. We then get the last 3*3 convolution layer with 256 filters of this kind. The step is 1 and the activation function is ReLu. The output feature map is 2X2X256. If the architecture is observed so far, there are more and more filters as we go deeper. Hence it is extracting additional features as we get further into the architecture. Furthermore, the filter size is lowering and the initial filter is larger and the filter size decreases as we proceed and the feature map shape is decreasing. Now, we apply the 3rd max-pooling layer of size 2*2 & stride 2. The output of the feature map shape is 1X1X256. Flatten is the function to transform the pooled map into a single column which is sent to a fully connected layer. Dense will add the fully connected neural network layer. We have a 1st fully connected layer with a ReLu activation function. The output dimension is 4096. The first drop-out layer comes next with a drop-out rate of 0.4 fixed. 2nd fully connected layer with 4096 neurons & ReLu activation function. Next follows the second drop-out layer with a 0.4 drop-out rate. 3rd fully connected layer with 1000 neurons & ReLu activation function. The third drop-out layer comes in the next

one with 0.4 drop-out rates. Finally, there is a layer or output layer fully connected with no. of class in the dataset. This layer uses Softmax for the activation function.

Step 6: Split the dataset

- We split the dataset into 80:20, 70:30, and 50:50 training or testing datasets.

Step 7: Model Training

The DFCNN model is trained and results are obtained for multiclass classification under SubCategory, MasterCategory, ArticleTypeCategory, also, a combination of categories under Gender+Master Category.

Step 8: Performance Evaluation

The performance of learning algorithms on test data is typically used to determine the quality of the algorithms [38]. The first is the Receiver Operating Characteristic Curve (ROC), also known as the Area Under the Curve (AUC), which is a widely used performance measure in supervised classification and is based on the relationship between sensitivity and specificity [39]. The second is the Receiver Operating Characteristic Curve (ROC), also known as the Area Under the Curve (AUC), which is a widely used performance measure in unsupervised classification. It was decided to use a generalization of the AUC for multiple classes in this study. Hand and Till [49][50][40] defined a function that performs multi-class AUC and is composed of the mean of several AUC values. This function is defined as follows: This can be calculated using Equation (1), where TP denotes the true positives, which are the number of instances that are positive and are correctly identified, and FN denotes the false negatives, which are the number of positive cases that are incorrectly classified as negative.

$$Sensitivity\ (Recall) = \frac{TP}{TP + FN} \quad (1)$$

If the conditional probability of true negatives is given a secondary class, specificity corresponds to the probability of the negative label being true, which means that it approximates the probability of the negative label being false; where TN is the number of true positives or negative cases that are negative and correctly classified as negative, and FP is defined as the number of false positives, defined as the number of negative instances that are incorrectly classified as positive cases.

$$Specificity = \frac{TN}{TN + FP} \quad (2)$$

In general, sensitivity and specificity are used to evaluate the effectiveness of an algorithm on a single class of data, with positive and negative values being used to indicate effectiveness.

The accuracy of a classification system is the most commonly used metric to evaluate classification performance. When it came to the evaluation stage, the accuracy was calculated after every 20 iterations of the procedure. Calculated as a percentage of correctly classified samples, this metric can be expressed in the following way:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

When it comes to precision, it's defined as the number of true positives divided by the sum of true positives and false positives. This measure is concerned with correctness, i.e., it assesses the algorithm's ability to predict the future. Precision refers to how "precise" the model is in predicting positive outcomes and how many of those predictions are correct.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

It is determined by the harmonic mean precision and recall, which is called the F-score. It is primarily concerned with the analysis of positive classes. This metric has a high value, which indicates that the model performs better when compared to the negative class.

$$F\text{-}score = \frac{2*Precision * Recall}{Precision + Recall} \quad (5)$$

5. Experimental Results & Analysis

We plot the accuracy evolution graph, Training and validation accuracy graph, confusion matrix, and model accuracy graph. We also display 3-4 results on internal images and 5-6 results on external images based on different classes (shirts, T-shirts, watches, sarees, jeans, jewelry, etc.). A confusion matrix is a table i.e. commonly applied to define the performance of a classification model (or "classifier") onset of known test data. The confusion matrix itself is simple to understand, but the verbiage connected with it may be confusing.

A. Overall Performance Analysis

The performance analysis of Master Category, ArticleType, SubCategory, and MasterGender Category is given in Table II, Table III, Table IV, and Table V.

Table II. Master Category based Classification

Category	Total classes	Splitting dataset	Training Accuracy	Validation Accuracy	No. of epochs
masterCategory	7	80:20	0.99	0.98	50
masterCategory	7	70:30	0.99	0.97	50
masterCategory	7	50:50	0.99	0.96	50

Table III. SubCategory based Classification

Category	Total classes	Splitting dataset	Training Acc	Validation Acc	No. of epochs
subCategory	45	80:20	0.99	0.90	50
subCategory	45	70:30	0.99	0.93	50
subCategory	45	50:50	0.99	0.91	50

Table IV. Gender+ Master Category based Classification

Category	Total classes	Splitting dataset	Training Acc	Validation Acc	No. of epochs
Gender+masterCategory	23	80:20	0.99	0.88	50
Gender+masterCategory	23	70:30	0.98	0.85	50
Gender+masterCategory	23	50:50	0.98	0.85	50

Table V. Article-type Category based Classification

Category	Total classes	Splitting dataset	Training Acc	Validation Acc	No. of epochs
articleType	143	80:20	0.98	0.81	50
articleType	143	70:30	0.97	0.79	50
articleType	143	50:50	0.98	0.74	50

B. Category wise Detailed Experimental Results

Master Category

The performance analysis of the Master Category is given in Table II.

- **Displaying total products**

```
# total no. of product based on masterCategory
df['masterCategory'].value_counts()
```

Apparel 21397
Accessories 11274
Footwear 9219
Personal Care 2403
Free Items 105
Sporting Goods 25
Home 1

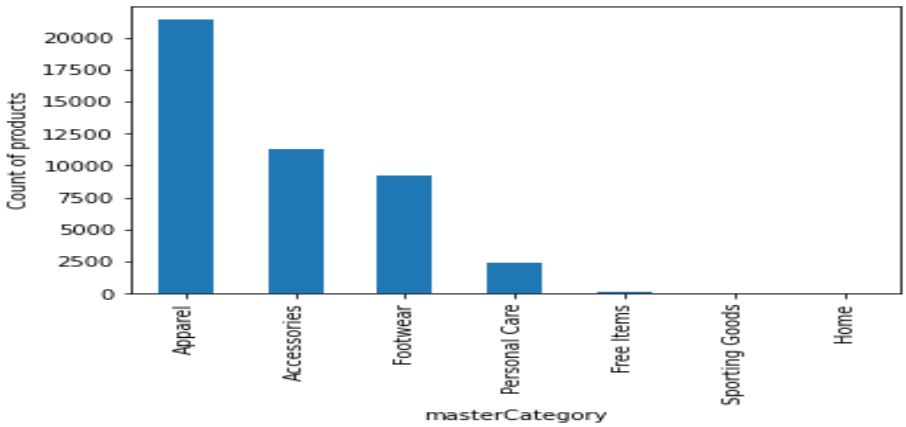


Figure 7: The Graph of Count total no of images by Master Category

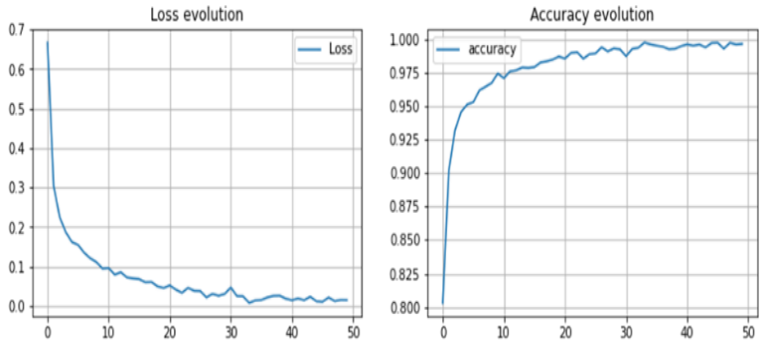


Figure 8: Master Category Loss-Accuracy Evolution Graph

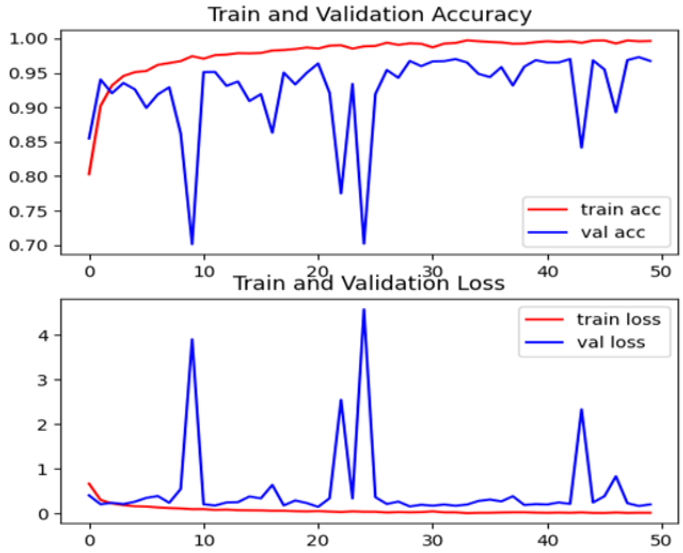


Figure 9: Training and Testing Validation Accuracy and Loss

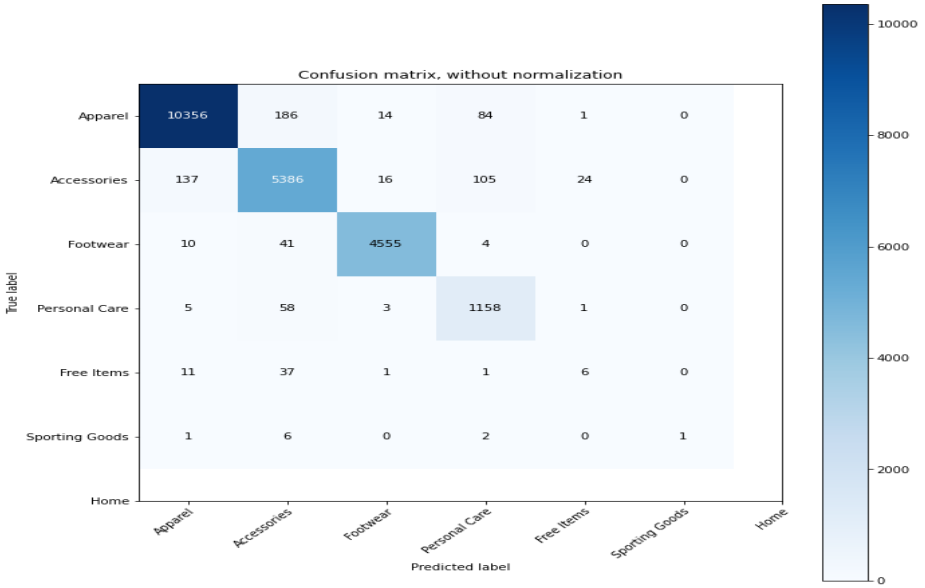


Figure 10: Master Category Classification Confusion Matrix

- **Per class Performance Evaluation**

For each class in MasterCategory, the performance is evaluated as given below.

ID	precision	recall	f1-score	support
0	0.98	0.97	0.98	10641
1	0.94	0.95	0.95	5668
2	0.99	0.99	0.99	4610
3	0.86	0.95	0.90	1225
4	0.19	0.11	0.14	56
5	1.00	0.10	0.18	10

Accuracy-0.98

- **Prediction Results**

The results of the predictions are given in Figure 11. PC indicates predicted class and TC indicates True Class of product.

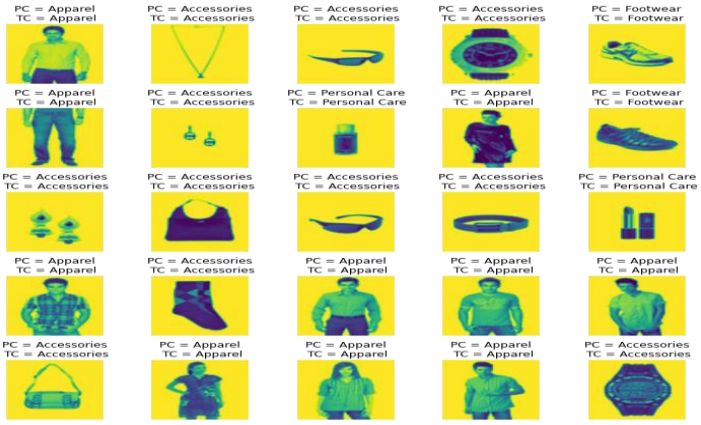


Figure 11: Prediction Results

Similarly, results are obtained by the training model in 50:50, 70:30 and 80:20. Figure 12, Figure 13 and Figure 14 shows Accuracy-Loss Evolution graphs of each categories under experimental work.

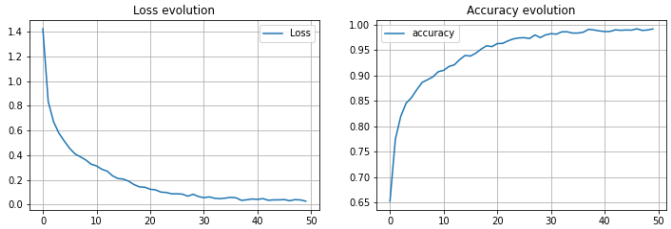


Figure 12: Sub Category Accuracy-Loss Evolution Graphs

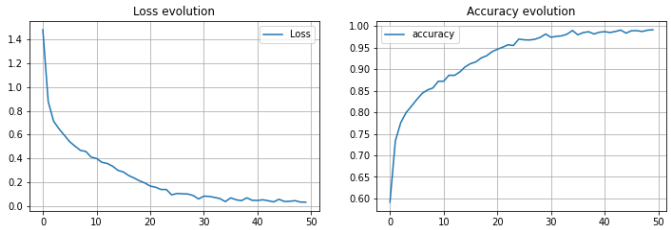


Figure 13: Gender+MasterCategory Accuracy-Loss Evolution Graph

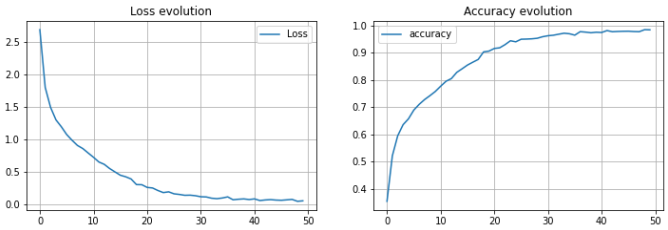


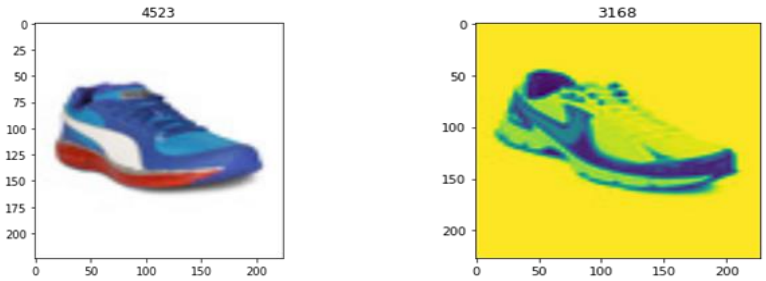
Figure 14: ArticleType Accuracy-Loss Evolution Graph

C. Search Results using DFCNN

The cosine similarity approach is used to evaluate the performance of our visual search models. A measure of cosine similarity between two non-zero vector arrays in a vector space is computed by taking the cosine of the angle between these two vectors and computing their similarity. The result is neatly bounded in the range [0,1], with 1 indicating that two vectors with the same orientation are similar, and 0 indicating that two vectors with opposite orientations are dissimilar. The visual search models are also evaluated qualitatively, with the following steps: 1) random selection of a target image to get recommendations from the current dataset; 2) comparison of the recommended image labels (gender, masterCategory, sub-category, and articleType) against the target image; and 3) using external images as targets and retrieving the recommended image labels from the current dataset. During the visual search process, the top 5, top 10, and top 20 matching images are recommended based on the CNN, autoencoder, and DFCNN embedding models, respectively. The results of the image search using both internal and external images are depicted in Figures 16, 17, and 18. The target image is the image on the left-hand side of each box, and the five images on the right-hand side of each box are the results of the recommendation using the autoencoder model.

1) Internal Image as Input

- The internal image is given as input to obtain search results.

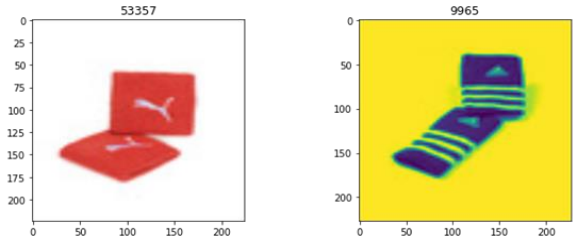



```
numpy_image,result=make_prediction(img)
print(result)
Output-Sports Shoes
```

Figure 15: Auto-labeling of product

- Internal Image as Input with less number of products

The proposed model is designed to obtain results for products with the minimum number of product images.




```
numpy_image,result=make_prediction(img)
print(result)
Output-Writstbands
len(typeList)
Output- 6 (Number of products)
```

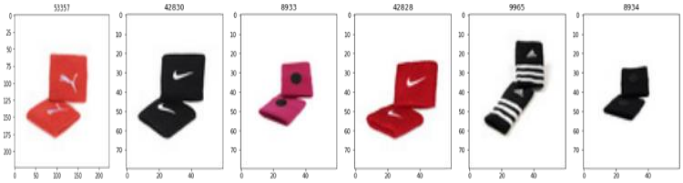


Figure 16: Search Result for the product with less number of products in the category using DFCNN

- Internal Image Input-Top-20 Results

Figure 17: Top-20 search results with DFCNN

2) External Image as Input

The external image is given as input that is not present in the dataset and search results are obtained.

Figure 18: Search results for External image or photograph given as input

5. Conclusion & Future Work

The E-commerce industry necessitates the use of optimized visual search to facilitate the retrieval or searching of products. The application of machine learning models in enabling visual search, which solves the problem of searching for products when the correct keywords are not known by the user, is particularly interesting to researchers. Because it allows users to search for similar products simply by taking a photo of the product in question, it improves the customer experience for everyone who uses it. Machine learning models can also be extremely useful in improving the experience of sellers when they are listing their products on a platform. Sellers can upload photos of their products, and automated image-to-text machine learning algorithms can generate appropriate tags to label the products on the marketplace for them. This can help to reduce inaccuracies in product labeling, which can hurt demand because the products are not displayed correctly in search results when they are not rendered correctly. The proposed Deep Fashion Convolution Neural Network (DFCNN) model improved the performance and accuracy of search results by using deep fashion convolution neural networks. In previous models, the limitation was found to be that if the number of products was less than a certain threshold, those products were ignored. With DFCNN model, this limitation is no longer present. In addition, the computation time is reduced. Indexing method implemented makes searching faster. When internal and external images are provided as input, along with appropriate product labeling, the autoencoders developed in conjunction with the DFCNN model outperform search results in terms of performance.

Machine learning models may also prove useful in the future when it comes to improving the experience of sellers when they are listing their products on the platform. Sellers can upload photos of their products, and automated image-to-text machine learning algorithms can generate appropriate tags to label the products on the marketplace for them. This can help to reduce inaccuracies in product labeling, which can have a negative impact on demand because the products are not displayed correctly in search results when they are not rendered correctly. To accomplish this, CNN models must be combined with natural language processing techniques such as Word2vec to predict text information from image data and its features. Image classification may also be used in the identification of counterfeit products, which is another possible application. An in-depth examination of the characteristics of a brand's logo, such as its design, colors, placement, and so on, can aid in the identification of bogus products. As computing power and machine learning algorithms continue to advance at a rapid pace, we may even be able to use generative adversarial networks (GANs) to come up with new designs for

fashion accessories, thus reducing our reliance on human creativity. Despite the fact that training a GAN model is extremely difficult, GAN models in the fashion industry have the potential to provide significant business value very quickly.

Acknowledgments

Ms. Smita Bhoir

M.E. Computer Engineering, PhD (Pursuing). She is pursuing her PhD from K.J. Somaiya College of Engineering, Mumbai , India. She is having 12 years of R & D experience. Published 15 research papers, delivered 45 technical talks, organized 40workshops and training programs. Her area of specialization is Data Science, Deep Web Mining, Wireless Networks.

Dr. Sunita Patil

PhD. Computer Engineeering .She is working as Vice Principal, Dean Academics, Professor, Department of Computer Engineering, K.J. Somaiya IEIT, Sion, Mumbai, India. She is having more than 20 years of R & D experience. She has published 30 research papers, organized more than 50 workshops, delivered more than 50 technical talks. Her area of specialization is Data Mining.

REFERENCES

[1] H. Hong and J. H. Kang, "The impact of moral philosophy and moral intensity on purchase behavior toward sustainable textile and apparel products," *Fash. Text.*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40691-019-0170-8.

[2] S. Y. Kim and J. Ha-Brookshire, "Evolution of the Korean Marketplace From 1896 to 1938: A Historical Investigation of Western Clothing Stores' Retail and Competition Strategies," *Cloth. Text. Res. J.*, vol. 37, no. 3, pp. 155–170, Jul. 2019, doi: 10.1177/0887302X19835967.

[3]S. Joo and J. Ha, "Fashion Industry System and Fashion Leaders in the Digital Era," *J. Korean Soc. Cloth. Text.*, vol. 40, no. 3, pp. 506–515, 2016, doi: 10.5850/JKSCT.2016.40.3.506.

[4] J. Jo, S. Lee, C. Lee, D. Lee, and H. Lim, "Development of fashion product retrieval and recommendations model based on deep learning," *Electron.*, vol. 9, no. 3, pp. 1–12, 2020, doi: 10.3390/electronics9030508.

[5] P. Kaur and R. K. Singh, "An efficient analysis of machine learning algorithms in CBIR," *Proc. Int. Conf. Comput. Autom. Knowl. Manag. ICCAKM 2020*, pp. 18–23, 2020, doi: 10.1109/ICCAKM46823.2020.9051528.

[6] L. Fengzi, S. Kant, S. Araki, S. Bangera, and S. S. Shukla, "Neural Networks for Fashion Image Classification and Visual Search," *SSRN Electron. J.*, 2020, doi: 10.2139/ssrn.3602664.

[7] J. Sang, C. Xu, and D. Lu, "Learn to personalized image search from the photo-sharing websites," *IEEE Trans. Multimed.*, vol. 14, no. 4 PART1, pp. 963–974, 2012, doi: 10.1109/TMM.2011.2181344.

[8] N. Kumar, S. Awasthi, and D. Tyagi, "Web Crawler Challenges and Their Solutions," vol. 7, no. 12, pp. 95–99, 2016.

[9] X. Zhang, Z. Li, and W. Chao, "Improving image tags by exploiting web search results," *Multimed. Tools Appl.*, vol. 62, no. 3, pp. 601–631, 2013, doi: 10.1007/s11042-011-0863-5.

[10] P. S., "An Image Crawler for Content-Based Image Retrieval System," *Int. J. Res. Eng. Technol.*, vol. 02, no. 11, pp. 33–37, 2013, doi: 10.15623/ijret.2013.0211006.

[11] M. Vagač *et al.*, "Crawling images with web browser support," *2015 IEEE 13th Int. Sci. Conf. Informatics, INFORMATICS 2015 - Proc.*, pp. 286–289, 2016, doi: 10.1109/Informatics.2015.7377848.

[12] X. Liu, J. Li, J. Wang, and Z. Liu, "MMFashion: An Open-Source Toolbox for Visual Fashion Analysis," pp. 1–4, 2020, [Online]. Available: http://arxiv.org/abs/2005.08847.

[13] V. Tyagi, "Content-Based Image Retrieval Techniques: A Review," *Content-Based Image Retr.*, pp. 29–48, 2017, doi: 10.1007/978-981-10-6759-4_2.

[14] N. C. Mithun, R. Panda, and A. K. Roy-Chowdhury, “Construction of Diverse Image Datasets from Web Collections with Limited Labeling,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 1147–1161, 2020, doi: 10.1109/TCSVT.2019.2898899.
[15] W. Zhou, H. Li, and Q. Tian, “Recent Advance in Content-based Image Retrieval: A Literature Survey,” pp. 1–22, 2017, [Online]. Available: http://arxiv.org/abs/1706.06064.
[16] A. E. A. M. Alomairi and G. Sulong, “An overview of content-based image retrieval techniques,” *J. Theor. Appl. Inf. Technol.*, vol. 84, no. 2, pp. 215–223, 2016.
[17]A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.” Accessed: Jun. 03, 2021. [Online]. Available: http://code.google.com/p/cuda-convnet/.
[18] W. Lu, L. Li, J. Li, T. Li, H. Zhang, and J. Guo, “A multimedia information fusion framework for web image categorization,” *Multimed. Tools Appl.*, vol. 70, no. 3, pp. 1453–1486, 2014, doi: 10.1007/s11042-012-1165-2.
[19] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, “Supervised Hashing for Image Retrieval via Image Representation Learning.” Accessed: Jun. 03, 2021. [Online]. Available: www.aaai.org.
[20] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, “Deep Learning of Binary Hash Codes for Fast Image Retrieval.” Accessed: Jun. 03, 2021. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W03/html/Lin_Deep_Learning_of_2015_CVPR_paper.html.
[21] C. Szegedy *et al.*, “Going Deeper with Convolutions.” Accessed: Jun. 04, 2021. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html.
[22]J.-C. Chen and C.-F. Liu, “Visual-based Deep Learning for Clothing from Large Database,” *dl.acm.org*, vol. 07-09-October, Oct. 2015, doi: 10.1145/2818869.2818902.
[23] N. Khosla and V. Venkataraman, “Building Image-Based Shoe Search Using Convolutional Neural Networks.” Accessed: Jun. 04, 2021. [Online]. Available:http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/nealk_final_report.pdf.
[24] Bhoir, Smita V., and Sunita Patil. "A Review on Recent Advances in Content-Based Image Retrieval used in Image Search Engine." *Library Philosophy and Practice* (2021): 1-45.
[25] X. Xie, W. Huang, H. H. Wang, and Z. Liu, “Image retrieval with adaptive SVM and random decision tree,” *Proc. Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud), I-SMAC 2018*, no. 1, pp. 784–787, 2019, doi: 10.1109/I-SMAC.2018.8653699.
[26] N. Kondylidis, M. Tzelepi, and A. Texas, “Exploiting tf-idf in deep Convolutional Neural Networks for Content-Based Image Retrieval,” *Multimed. Tools Appl.*, vol. 77, no. 23, pp. 30729–30748, 2018, doi: 10.1007/s11042-018-6212-1.
[27] X. Shi *et al.*, “Pairwise based deep ranking hashing for histopathology image classification and retrieval,” *Elsevier*, Accessed: Jun. 04, 2021. [Online]. Available:https://www.sciencedirect.com/science/article/pii/S0031320318301055?casa_token=Fp3WRjLtl9MAAAAA:-gdc73HVwoDSM-kofFFyRcifUb5v3kgc86SppFdoeZzfOTWWKya5lbgqyaLEJrs_mqrY1ZrApk0QLw.
[28] Z. Li, J. Tang, and T. Mei, “Deep Collaborative Embedding for Social Image Understanding,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2070–2083, 2018, doi: 10.1109/TPAMI.2018.2852750.
[29] R. D. Meltser, S. Banerji, and A. Sinha, “What’s that Style? A CNN-based Approach for Classification and Retrieval of Building Images,” *2017 9th Int. Conf. Adv. Pattern Recognition, ICAPR 2017*, pp. 9–14, 2021, doi: 10.1109/ICAPR.2017.8593206.
[30] A. Shah, R. Naseem, Sadia, S. Iqbal, and M. A. Shah, “Improving CBIR accuracy using convolutional neural network for feature extraction,” *Proc. - 2017 13th Int. Conf. Emerg. Technol. ICET2017*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/ICET.2017.8281730.

[31] S. Jain and J. Dhar, “Image-based search engine using deep learning,” *2017 10th Int. Conf. Contemp. Comput. IC3 2017*, vol. 2018-Janua, no. August, pp. 1–7, 2018, doi: 10.1109/IC3.2017.8284301.
[32] M. R. Kapadia and C. N. Paunwala, “Improved CBIR System Using Multilayer CNN,” *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, no. Icirca, pp. 840–845, 2018, doi: 10.1109/ICIRCA.2018.8597199.
[33] T. Anjali, N. Rakesh, and K. M. P. Akshay, “A Novel Based Decision Tree for Content-Based Image Retrieval: An Optimal Classification Approach,” *Proc. 2018 IEEE Int. Conf. Commun. Signal Process. ICCSP 2018*, pp. 698–704, 2018, doi: 10.1109/ICCSP.2018.8524326.
[34] F. Sabahi, M. O. Ahmad, and M. N. S. Swamy, “Perceptual Image Hashing Using Random Forest for Content-based Image Retrieval,” *2018 16th IEEE Int. New Circuits Syst. Conf. NEWCAS 2018*, pp. 348–351, 2018, doi: 10.1109/NEWCAS.2018.8585506.
[35] R. Fu, B. Li, Y. Gao, and W. Ping, “Content-based image retrieval based on CNN and SVM,” *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 638–642, 2017, doi: 10.1109/CompComm.2016.7924779.
[36] N. Bhosle and M. Kokare, “Random forest-based long-term learning for content-based image retrieval,” *2016 Int. Conf. Signal Inf. Process. IConSIP 2016*, 2017, doi: 10.1109/ICONSIP.2016.7857468.
[37] A. Latif *et al.*, “Content-based image retrieval and feature extraction: A comprehensive review,” *Math. Probl. Eng.*, vol. 2019, 2019, doi: 10.1155/2019/9658350.
[38] M. M. Rahman, S. K. Antani, and G. R. Thoma, “A learning-based similarity fusion and filtering approach for biomedical image retrieval using SVM classification and relevance feedback,” *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 4, pp. 640–646, 2011, doi: 10.1109/TITB.2011.2151258.
[39] D. Zhang, M. M. Islam, and G. Lu, “A review on automatic image annotation techniques,” *Pattern Recognit.*, vol. 45, no. 1, pp. 346–362, 2012, doi: 10.1016/j.patcog.2011.05.013.
[40] Z. Zeng, S. Cai, and S. Liu, “A novel image representation and learning method using SVM for region-based image retrieval,” *Proc. 2010 5th IEEE Conf. Ind. Electron. Appl. ICIEA 2010*, pp. 1622–1626, 2010, doi: 10.1109/ICIEA.2010.5514756.
[41] W. S. Wang and J. J. Wu, “Training asymmetry SVM in image retrieval using unlabeled data,” *Proc. 2009 2nd Int. Congr. Image Signal Process. CISP’09*, 2009, doi: 10.1109/CISP.2009.5303323.
[42] J. Donahue *et al.*, “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.” Accessed: Jun. 04, 2021. [Online]. Available: https://github.com/.
[43] D. Han, Q. Liu, and W. Fan, “A new image classification method using CNN transfer learning and web data augmentation,” *Expert Syst. Appl.*, vol. 95, pp. 43–56, 2018, doi: 10.1016/j.eswa.2017.11.028.
[44] W. Zhou, H. Li, Y. Lu, and Q. Tian, “Large scale image search with geometric coding,” in *ACM international Conference on multimedia*, 2011, pp. 1349–1352.
[45] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Lost in quantization: Improving particular object retrieval in large scale image databases,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[46] O. Chum, J. Philbin, and A. Zisserman, “Near duplicate image detection: min-hash and tf-idf weighting,” in *British Machine Vision Conference*, vol. 3, 2008, p. 4
[47] Z. Wu, Q. Ke, M. Isard, and J. Sun, “Bundling features for large scale partial-duplicate web image search,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 25–32.
[48] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, “Spatial coding for large scale partial-duplicate web image search,” in *ACM International Conference on Multimedia*, 2010, pp. 511–520.
[49] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, “Total recall II: Query expansion revisited,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 889–896.
[50] Y. Zhang, Z. Jia, and T. Chen, “Image retrieval with geometry preserving visual phrases,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 809–816.