2024 3rd International Conference for Innovation in Technology (INOCON)
Karnataka, India. Mar 1-3, 2024

FashionAI: Image-Based Clothing Detection and Shopping Recommendation

Disha Jain
Dept. of Computer Science
PES University
Bengaluru, India
disha200129@gmail.com

Elizabeth Maria Thazhathu
Dept. of Computer Science
PES University
Bengaluru, India
elizabethmt1711@gmail.com

Isha Adiraju
Dept. of Computer Science
PES University
Bengaluru, India
isha.adiraju@gmail.com

Juhi Bhattacharya
Dept. of Computer Science
PES University
Bengaluru, India
bejuhi13@gmail.com

Prof Dinesh Singh
Dept. of Computer Science
PES University
Bengaluru, India
dineshs@pes.edu

Abstract—**In today's world, the retail fashion industry is going through radical transformations in terms of technology. The magnitude of options available on the internet makes it difficult to find suitable clothing items on online retail stores in an effective manner. To overcome this, we aimed to create a project that detects specific items of clothing from an image and makes real-time recommendations. We have developed a system which uses computer vision techniques to analyze images and recognize different types of clothing such as shirts, pants, dresses, etc. The system will then use machine learning algorithms to make product recommendations based on the similarity between the image uploaded by the user and products available on various e-commerce websites.**

Keywords— Deep Learning, Clothing item detection, Object detection, Image Classification, Feature extraction, Transfer Learning, Product Recommendation, Convolutional Neural Networks (CNN), YOLOv5, Inceptionv3, Deepfashion2. Webscraping, MongoDB

I. INTRODUCTION

With advancement in technology and in the apparel industry, there are numerous options available for every search result. Amidst the abundant online information, it has become challenging to articulate our preferences via textual searches in terms of product details. Moreover, it becomes tedious to scroll through large amounts of search results offered by thousands of online websites. Visual searches have thus emerged as a potential remedy to these issues at hand. We aimed to create an interactive and user-friendly application that not only saves time and effort but also enhances the quality of results. Our project takes in user input in the form of uploaded images and displays relevant clothing items available online from various websites [1]. The application uses computer vision techniques to analyze the images given by the user and performs various image classification techniques. We used models such as YOLOv5 for bounding box detection, and CNN for feature extraction. The results obtained from the models are used to extract product information from chosen sample websites and is then saved in a MongoDB database. Finally as the output, we present the appropriate product links from online retail stores to the user.

A few other features include — A strictly visual-based search platform, which allows for more accurate results based on similarity compared to text input, identification of every individual clothing article in the given image so that the user can search for any of the desired items from the image, a location-based feature where users can view shipping time of the clothing article to their location, and filters such as cost, sizing, etc to provide results tailored to customers' needs.

II. RELATED WORKS

The following section provides a comprehensive review of the research papers that have been consulted in order to explore and analyze the diverse range of approaches that have been undertaken with regards to our problem statement.

- Chang and Zhang's paper, "Deep Learning for Clothing Style Recognition Using YOLOv5" [2], applies the YOLOv5 algorithm to clothing style detection. In order to increase speed, the study focuses on one-stage object detection and investigates CNN models for clothing categorization, such as YOLOv3 and Tiny YOLO. To overcome classification mistakes, the authors introduce a learning framework for automatic clothing genre classification and offer a collaborative fashion style recognition model. The study also contrasts the effectiveness of one-stage (YOLO series) and two-stage (R-CNN) object detection methods.
- Islam et al.'s study, "A CNN-Based Approach for Garments Texture Design Classification" [3], presents two deep CNN models—AlexNet and VGGS—for the categorization of garment textures and compares them to currently used hand engineering techniques. Two datasets

are used in the study for training and testing. Using distinct pooling layers, AlexNet's eight layers—five convolutional and three fully connected—address overfitting. The VGGS model includes a dropout layer for multiway softmax classification, a bigger pooling size, and small stride in some convolutional layers. With a 77.8% accuracy rate across two different datasets, the paper shows how well CNNs perform in classifying the texture of clothing.

- The method described in the paper "Image-Based Fashion Product Recommendation with Deep Learning" [4] by Tuinhof et al. is a two-step process that involves employing a CNN as a feature extractor once it has been trained for image classification tasks. These features are ranked using a modified k-nearest neighbours (k-NN) method. For texture and category prediction, separate CNNs are trained, and their softmax output layers are eliminated. The feature vector of an input image is retrieved for k-NN similarity search, and the remaining CNNs are concatenated. Because of their promising outcomes, batch-normalized Inception architectures and AlexNet are used.

III. EXPLORATORY DATA ANALYSIS AND DATA PREPROCESSING

A. Data Sources and Collection

1) Fashion MNIST:
   - Fashion MNIST consists of 70,000 black and white images of fashion products that are classified into 10 different categories such as trouser, dress, sandal, etc. Its a simplified representation of clothing items in which each grayscale image is 28 pixels in height and 28 pixels in widths. This dataset is considered a benchmark to validate ones algorithms.
   - However, real-world clothing items have diverse patterns, colors, and textures. As this dataset includes only 10 clothing categories, it may not cover the full spectrum of fashion styles and types.

2) Deepfashion2:
   - DeepFashion2 is a detailed and fine-grained comprehensive fashion dataset [5]. It contains 801K diverse images of 13 popular clothing categories from both commercial shopping stores and consumers. It provides a more comprehensive representation and understanding of different styles, which is crucial for accurate clothing recommendations.
   - The largest fashion dataset previous to Deepfashion2 was called Deepfashion [6]. It had its own drawbacks such as single clothing item per image, sparse landmark and pose definition. Deepfashion2 has 801k clothing images where each clothing item in an image is labeled with scale, occlusion, zooming, viewpoint, bounding box, dense landmarks, and per-pixel mask.

3) Dress Pattern Dataset :
   The collection of apparel items includes a variety of patterns, such as floral, plain, striped, etc. There are ten different classes in all, and each class has roughly 500 photos. This custom dataset was created for convolutional neural network (CNN) applications. It has been carefully crafted to meet CNN-specific needs, negating the need for any preprocessing.

For this project, we selected the DeepFashion2 dataset as the primary source of fashion images to train the image detection model. DeepFashion2 is a large-scale dataset that contains diverse images of clothing items, making it suitable for training and evaluation. We also selected the Dress Pattern Dataset to train the CNN model for feature extraction and classification.

B. Data Preprocessing and Conversion

The Deepfashion2 dataset is split into a training set (391K images), a validation set (34k images), and a test set (67k images). Each image in separate image set has a unique six-digit number and a corresponding annotation file in json format is provided in annotation set with the same unique six-digit number.
This format is compatible with the COCO (Common Objects in Context) format [7]. To make it compatible with the YOLOv5 model, the annotations were converted to a text file in the Darknet format which consists of the class of the item and the coordinates of its bounding box. The dataset was then split into two folders; Images and Labels, each consisting of the training and validation sets.

IV. MODELS AND METHODOLOGY

A. Transfer Learning

Transfer learning is a machine learning strategy that involves re-purposing knowledge gained from solving one problem to enhance performance on a related task. This technique is advantageous due to its data efficiency as well as time and resource saving ability [8].
In our project, to enhance the model's performance and adapt it to our specific use case, we implemented transfer learning using the DeepFashion2 dataset. On using a pre-trained model, it allowed for efficient adaptation to a specific target task through fine-tuning. Transfer learning not only reduced the need for an extensive amount of labeled data but also significantly expedited the training process. The model, enriched with knowledge from the pre-training phase, demonstrated a higher proficiency in recognizing clothing items in user-uploaded images

B. Bounding box detection

The following models were considered for bounding box detection:

1) *VGG-16:* VGG-16 is a Deep convolutional neural network which comprises of 16 layers with usage of small 3x3 convolutional filters [9][10]. It employs max pooling for down-sampling of the features being extracted. VGG-16 is renowned for its deep architecture and strong feature extraction capabilities. However, in our specific application, we encountered challenges with VGG-16, primarily related to its computational intensity and relatively lower efficiency in detecting bounding boxes for clothing items in diverse images.

2) *YOLOv5:* YOLOv5, standing for ”You Only Look Once, version 5” is a cutting-edge object detection model in the YOLO family created by Ultralytics [2][7][11]. It is renowned for its speed and accuracy. YOLOv5 uses a single neural network to analyze the entire image at once. This makes it significantly faster than older, two-stage models like VGG-16. YOLOv5's architecture has the ability to predict bounding boxes and class probabilities in a single pass. It is also better for real-time object detection and handles dense objects better. Hence, the YOLOv5 proved to be most suitable model for our intended use.

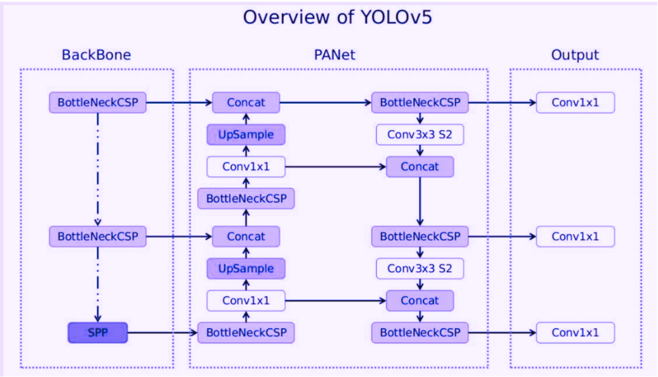


Fig. 1. YoloV5 Architechture

C. Image classification based on features

The following models were considered for feature extraction:

1) *Keras' MNIST CNN:*
Keras' MNIST Sequential CNN model is a foundational example for understanding Convolutional Neural Network and building image classification models. Keras' user-friendly API makes it straightforward to build and train the model even for beginners. However, while efficient for MNIST dataset, the model is not the best choice for complex image classification problems with larger datasets and real-time applications.

2) *ResNet:*
ResNet architecture introduces the key innovation of 'residual learning', which pertains to the challenges faced during training of extremely deep neural networks [12]. The architecture is hence highly parameterized to adapt with much larger data sets and can be more memory-intensive especially for deeper variants. This becomes an issue when there are less computational resources available at hand, as well as limited data to train. Over-fitting is also faced as not enough data is available to fine-tune the large number of parameters.

3) *MobileNet:*
MobileNet is a popular choice for applications with constraints on processing power and battery life [13]. However, MobileNet's simpler feature representation limits its effectiveness on fine-grained tasks. Compared to deeper models like ResNet, MobileNet generally reaches its accuracy limit earlier, making it less suitable for tasks requiring the absolute highest accuracy. With fewer parameters, MobileNet can be more susceptible to overfitting, especially on smaller datasets.

4) *Alexnet:*
AlexNet is a pioneering convolutional neural network used primarily for image recognition and classification tasks. It was designed to process images with a specific input size of 227x227 pixels [14]. If the input images have different dimensions, resizing them to fit the model's input size might result in information loss or distortion, potentially impacting the model's performance. Newer CNNs often have more flexible input sizes, offering greater adaptability. It also requires significant processing power and memory for training making it unsuitable for real-time applications.

5) *InceptionV3:*
Inception v3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for GoogLeNet [8]. The model itself is made up of symmetric and asymmetric building blocks - including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. It also achieves multi-scale feature extraction by using filters of various sizes in its convolutional layers. Hence, though not the latest, the model remains popular for its balanced attributes of accuracy, efficiency, and ease of use.

D. Webscraping

In order to perform Web Scraping, Python provides various libraries to retrieve data from websites in a simple manner [14]. We have performed web scraping on two varieties of websites, static websites such as H&M, and dynamic websites such as Amazon and Myntra.

- *Static websites:*
Static websites refer to the online clothing websites that are relatively small-scaled. In this case, the web pages are stored in the servers using fixed HTML code, and hence is standard for every viewer. An example of a static website which we chose for the project is H&M.
Python's BeautifulSoup4 library provides an easy and seamless way to access the required static HTML tags. Using the lxml parser, product details are retrieved from the desired HTML labels such as product name, product link, price and delivery. Since the same labels are repeated for every product on the website, the same code is used to iterate through all the desired products. The data collected is finally stored in the form of a list of dictionaries, each dictionary dedicated for a separate product. The list of products are stored in the database for displaying and filtering purposes.

- *Dynamic websites:*
Dynamic websites change as users interact with them. An example used in this case is Amazon - the website targets users to provide personalised recommendations and hence the overall structure is not static.
Selenium WebDriver, imported from python, is a collection of API's that allow automated parsing of a website. It can manipulate a browser programmatically, and is required when working with dynamic websites where JavaScript is used to load content asynchronously. The python script uses Selenium library for browsing through the webpage for elements that are relevant to the search query. The data collected is then appended into a list of dictionaries, similar to how static websites are stored. However, BeautifulSoup4 library is incapable of obtaining real-time product details from e-commerce websites and hence is not used in this case.

E. Data Storage

MongoDB was used for data storage purpose. MongoDB is an open-source document-oriented database program [16][17]. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with flexible schemas. The metadata collected from the websites from web scraping are converted to a JSON format which is then combined and easily imported to MongoDB. A NoSQL database system like MongoDB is best suited for unstructured data with low latency. It is also easier to query to find specific entries. This makes it a good fit to filter fashion products based on parameters such as price, location of shipping, etc. It can also easily be expanded for use on a commercial level.

F. Front-end

Flask is used to implement a seamless front end for the project. Flask is a simple Python framework used to build web-applications. It provides an easy methodology to create APIs that follow RESTful principles, which allows clean and predictable interfaces for applications. Flask also comes with a built-in development server, making it convenient for testing and debugging during the development phase.
The user interacts with the system through the front end. It accepts an input from the user in the form of an image consisting of the desired clothing item. The front end then displays the links of products obtained from online websites in the form of output. The user can click on these links which re direct them to the website to make purchases from.

V. Results and Analysis

The final project consists of a simple front end which accepts an image input from the user. The image gets saved and passed on to the models which perform their functionalities in the back-end. These processes are not visible to the user. The final output consists of the products extracted from online stores which are saved in the created MongoDB database. Filtering of data using attributes such as price and availability are enabled from here.

A. YoloV5

The YOLOv5 model was trained on the deepfashion2 dataset and attained an accuracy of 96%. During the training process, the model parameters are optimized based on the provided dataset. After setting up the model, the script iterates through the files in the dataset directory, filtering for image files with '.jpg' or '.png' extensions. For each qualifying image, the script constructs the full file path and performs inference using the trained YOLOv5 model. Inference results, which typically include information about detected objects and their bounding boxes, are then printed for each image. Finally, the model successfully creates bounding boxes for each item detected in the input image. It then generates appropriate labels by categorizing it into its respective clothing type. Each image is also cropped by the model which are then sent to the CNN model.

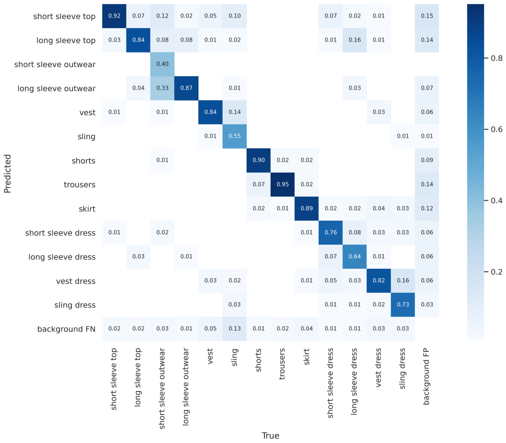


Fig. 2. YoloV5 Confusion Matrix

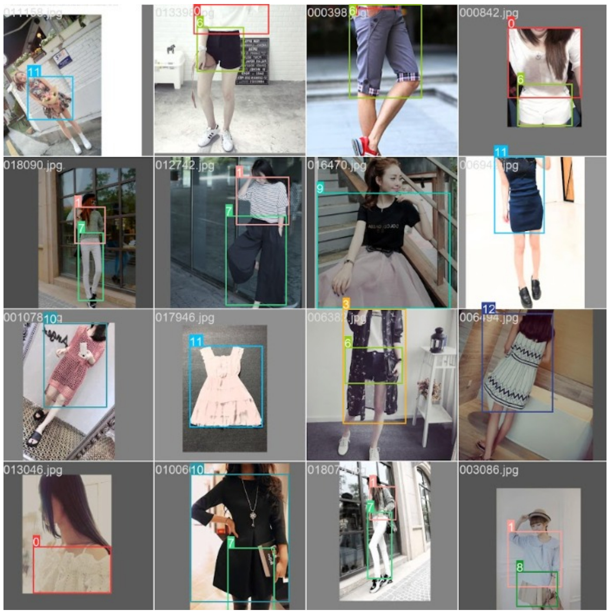
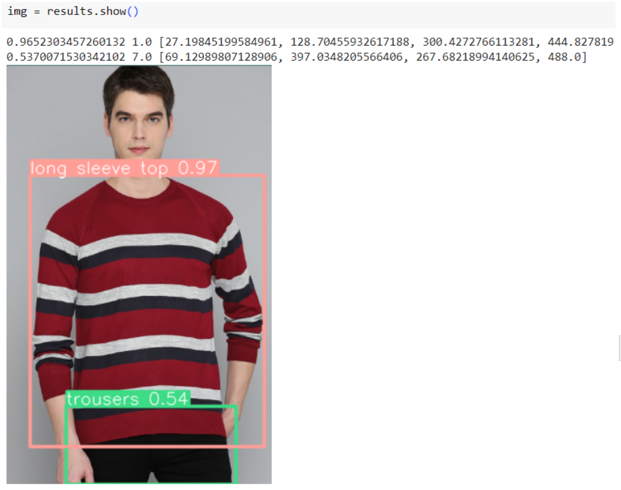


Fig. 3. YoloV5 created Bounding Boxes



```
img = results.show()
```

```
0.9652303457260132 1.0 [27.19845199584961, 128.70455932617188, 300.4272766113281, 444.827819
0.5370071530342102 7.0 [69.12989807128906, 397.0348205566406, 267.68218994140625, 488.0]
```

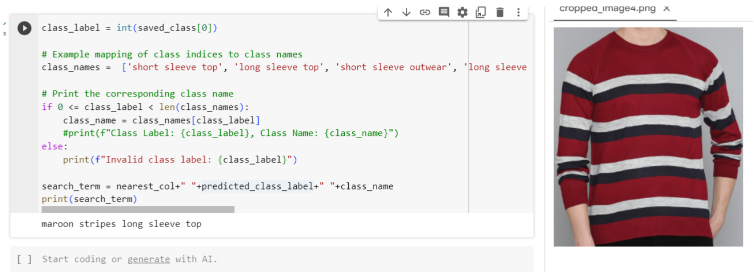
Fig. 4. YoloV5 output on test image

B. CNN

The CNN model is built upon the Inception v3 architecture. The image output from the YOLO model is cropped to isolate the detected clothing item, and this cropped region serves as the input to the CNN model for pattern classification.

The CNN architecture was designed to learn and differentiate between patterns such as stripes, polka dots, plain and floral patterns on the detected clothing items. The training process involved optimizing the network parameters using a labelled dataset of clothing patterns. The training process involved 20 epochs, with the model iteratively refining its parameters to minimize the specified loss function.

The ColourThief Python library was integrated into the system to extract the dominant color of the detected clothing items. This library analyses the color palette of an image and identifies the most prominent color, providing valuable information about the overall color scheme of the clothing.



```
class_label = int(saved_class[0])

# Example mapping of class indices to class names
class_names =  ['short sleeve top', 'long sleeve top', 'short sleeve outwear', 'long sleeve

# Print the corresponding class name
if 0 <= class_label < len(class_names):
    class_name = class_names[class_label]
    #print(f"Class Label: {class_label}, Class Name: {class_name}")
else:
    print(f"Invalid class label: {class_label}")

search_term = nearest_col+" "+predicted_class_label+" "+class_name
print(search_term)
```

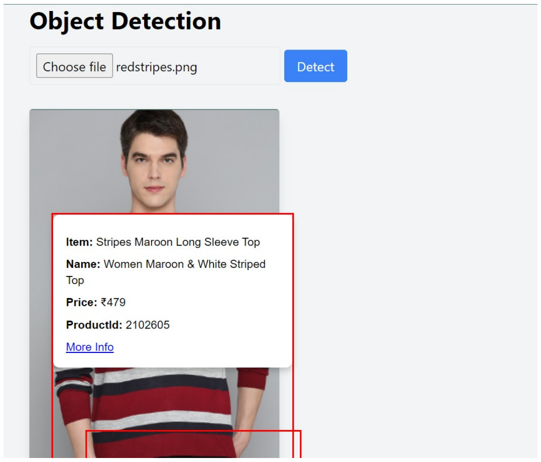
```
maroon stripes long sleeve top
```

```
[ ]  Start coding or generate with AI.
```

Fig. 5. CNN predicted label on test image

C. Webscraping Results

The labels generated from the CNN model act as input for the web scraping code. These labels act as input for the web scrapping code. The final output consists of relevant searches obtained from online websites.



Object Detection

Choose file redstripes.png Detect

Item: Stripes Maroon Long Sleeve Top

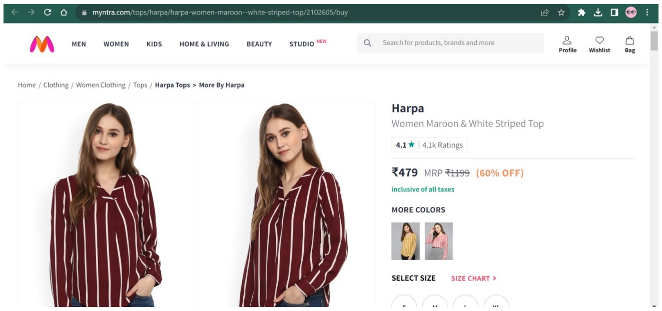
Name: Women Maroon & White Striped Top

Price: ₹479

ProductId: 2102605

More Info

Fig. 6. Front-end result on test image



myntra.com/tops/harpa/harpa-women-maroon--white-striped-top/2102605/buy

Home / Clothing / Women Clothing / Tops / Harpa Tops > More By Harpa

Harpa

Women Maroon & White Striped Top

4.1 ★ | 4.1k Ratings

₹479 MRP ~~₹1199~~ (60% OFF)

inclusive of all taxes

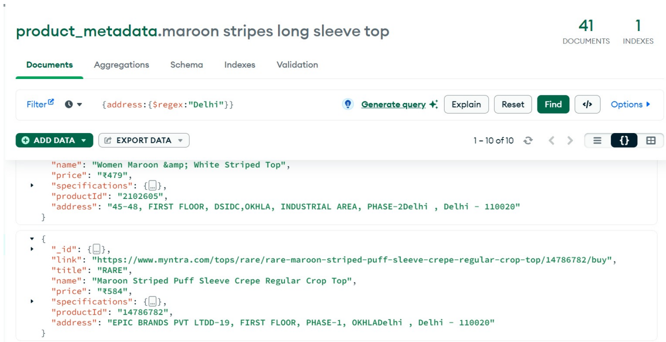
MORE COLORS

SELECT SIZE SIZE CHART >

Fig. 7. Product obtained on clicking the link

D. Database

After obtaining product information from online websites, the data is inserted into the database. A python script takes the list with dictionaries containing all the data and converts into a .JSON format. It then establishes a connection with MongoDB server.
Through querying, we are able to perform filtering according to various attributes such as price, colour, location, etc. On each query, a separate collection is created which stores all the products related to the filter applied.



```
{address:{$regex:"Delhi"}}
```

```
      "name": "Women Maroon &amp; White Striped Top",
      "price": "₹479",
    ▸ "specifications": {…},
      "productId": "2102605",
      "address": "45-48, FIRST FLOOR, DSIDC,OKHLA, INDUSTRIAL AREA, PHASE-2Delhi , Delhi - 110020"
    }

  ▾ {
    ▸ "_id": {…},
      "link": "https://www.myntra.com/tops/rare/rare-maroon-striped-puff-sleeve-crepe-regular-crop-top/14786782/buy",
      "title": "RARE",
      "name": "Maroon Striped Puff Sleeve Crepe Regular Crop Top",
      "price": "₹584",
    ▸ "specifications": {…},
      "productId": "14786782",
      "address": "EPIC BRANDS PVT LTDD-19, FIRST FLOOR, PHASE-1, OKHLADelhi , Delhi - 110020"
    }
```

Fig. 8. Product obtained on clicking the link

VI. CONCLUSION AND FUTURE WORK

In conclusion, this project successfully integrated the YOLOv5 model for clothing object detection and identification, a custom CNN for clothing pattern identification and ColourThief for dominant colour detection. The incorporation of web scraping to recommend similar clothing items from ecommerce websites like Myntra enhances the practical applications of the project, particularly in providing personalized recommendations to users.

With respect to the future work prospects in our project, there are many avenues we can explore and include to add more functionalities.

- Refinement of CNN model: Fine-tuning the CNN model for enhanced pattern classification and scalability is an essential next step.
- Optimizing Web Scraping model: Enhancing the efficiency of the web scraping module for faster and more responsive data retrieval from e-commerce websites.
- User Feedback Mechanisms: Implementing user feedback loops to collect information on the effectiveness of recommendations and areas for improvement.
- Expansion to Diverse Categories: Broadening the detection capabilities beyond clothing to include accessories, shoes, bags, and other fashion items.
- Location-Based Recommendations: Implementing a feature to recommend products based on the user's location, leveraging APIs and geospatial data.

REFERENCES

[1] Jaechoon Jo, Seolhwa Lee, Chanhee Lee, Dongyub Lee, Heuiseok Lim (2020). Development of Fashion Product Retrieval and Recommendations Model Based on Deep Learning. *Multidisciplinary Digital Publishing Institute*

[2] Chang, Y.-H.; Zhang, Y.-Y (2022). Deep Learning for Clothing Style Recognition Using YOLOv5. *Micromachines* 2022, 13, 1678.

[3] S. M.Sofiqul Islam, Emon Kumar Dey, Md. Nurul Ahad Tawhid, B.M.Mainul Hossain (2017). A CNN Based Approach for Garments Texture Design Classification. *Taiwan Association of Engineering and Technology Innovation: E-Journals.*

[4] Hessel Tuinhof, Clemens Pirker, Markus Haltmeier(2018). Image Based Fashion Product Recommendation with Deep Learning.

[5] Ge, Yuying, et al. "DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images." arXiv (Cornell University), Cornell University, 2019.

[6] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *In CVPR, 2016.*

[7] Manikandan, Malavika, et al. "Object Detection Using YOLO V5." *European Chemical Bulletin*, 2023

[8] Fengzi, Li, et al. "Neural Networks for Fashion Image Classification and Visual Search." *Social Science Research Network, Social Science Electronic Publishing, Apr. 2020*

[9] Tayade, Sejpal, et al. "Deep Learning Based Product Recommendation System and its Applications." *International Research Journal of Engineering and Technology (IRJET)*,2021.

[10] Mathews, Anu, et al. "Analysis of content based image retrieval using deep feature extraction and similarity matching." *International Journal of Advanced Computer Science and Applications*, 2022.

[11] Ultralytics *https://github.com/ultralytics/yolov5*

[12] Sridevi, M., et al. "Personalized Fashion Recommender System With Image Based Neural Networks." *IOP Conference Series: Materials Science and Engineering, vol. 981, IOP Publishing*, Dec. 2020, p. 022073.

[13] Verma, Dhruv, et al. "Addressing the cold-start problem in outfit recommendation using visual preference modelling." *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, 2020.

[14] C. Stan, I. Mocanu. "An Intelligent Personalized Fashion Recommendation System," *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019.

[15] Singrodia, Vidhi, et al. "A review on web scrapping and its applications." *2019 International Conference on Computer Communication and Informatics (ICCCI)*, 2019.

[16] Krishnan, Hema, et al. "MongoDB – a comparison with NoSQL databases." *International Journal of Scientific and Engineering Research*, 2016.

[17] Anjali Chauhan. "A Review on Various Aspects of MongoDb Databases" *International Journal of Engineering Research & Technology (IJERT)*, 2019.