Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Clothes Matching App

This documentation submitted as required for the degree of bachelors in Computer and Information Sciences.

By

Mahmoud Shehata Saad Eldeen	[scientific computing department]
Eslam Omar Mohamed	[scientific computing department]
Mustafa Mohamed Ali Hassan	[scientific computing department]
Alaa Ahmed Elsayd Abdelrahman	[scientific computing department]
Omnya Samih Gomaa	[scientific computing department]

Supervisors

Dr. Hanan Ahmed
Doctor of Scientific Computing Department
Faculty of Computer and Information Sciences
Ain Shams University.

TA. Andrew Magdy
Teaching Assistant of Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

Cairo, June 2024

Acknowledgments

We would like to express our sincere gratitude to all those who have supported us throughout the completion of this project.

First and foremost, we would like to thank Allah for giving us the strength and perseverance to see this project through to the end.

We would like to thank our doctor, Dr. Hanan Ahmed, for her meticulous supervision, insightful comments, and helpful suggestions throughout the project. Her expertise and dedication to our success were instrumental in helping us achieve our goals.

We are also deeply grateful to our TA Andrew Magdy for her unwavering guidance, support, and encouragement throughout the project. Her insights and feedback were invaluable in helping us refine our ideas and improve the project's quality.

We would also like to thank our team members for their hard work, cooperation, and commitment to excellence. Their contributions were essential in bringing this project to fruition.

Finally, we would like to thank our families for their unwavering love, support, and encouragement throughout our academic journey. Their sacrifices and dedication have been the foundation of our success.

Thank you all for your contributions to this project and for making this.

Abstract

Recently, the challenges of selecting and coordinating outfits have become increasingly difficult for many people. While some organizations have developed fashion recommendation systems, the process of choosing appropriate and stylish clothing can still be time-consuming and frustrating. An uncoordinated wardrobe can lead to stress and inefficiency in daily routines. Some applications have emerged to assist in matching clothes and suggesting outfits, but analyzing fashion trends and individual preferences can be complex and the accuracy of these systems is not always high.

In this study, the proposed app is an integrated and advanced clothes matching application that combines the detection of clothing items, fashion trends, and user preferences through the analysis of wardrobe images. The app is based on modern technological techniques, specifically deep learning. The app processes images of clothing items and uses a set of digital filters to enhance and analyze the images. It then goes through the processes of detecting clothing types, matching items based on fashion trends, and considering user preferences. The classification stage begins to identify suitable outfits and recommend combinations that are both stylish and season-appropriate.

Offers personalized recommendations by learning from the user's style preferences and wardrobe choices. It can suggest new outfit ideas, highlight trending fashion items, and even help users make better purchasing decisions by recommending items that complement their existing wardrobe. The app aims to reduce the time and effort spent on selecting outfits while boosting the user's confidence in their fashion choices. It also helps users stay up-to-date with the latest fashion trends and ensure they always look their best.

Moreover, the app places a strong emphasis on general taste and fashion sense, ensuring that all recommended outfits not only match the user’s style but also adhere to broader aesthetic standards. This focus on overall fashion sense helps users feel more confident and stylish in their daily attire. The app is designed to cater to a wide range of fashion preferences, making it a versatile tool for anyone looking to enhance their wardrobe and fashion choices

Table of Contents

List of Figures

List of Table

List of Abbreviations

Abbreviation	**Stands for**
ANN	**A**rtificial **N**eural **N**etwork
AR	**Auto-Regressive**
CNN	**C**onvolutional **N**eural **N**etwork
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
KNN	**K N**earest **N**eighbor
LDA	**L**inear **D**iscriminant **A**nalysis
ML	**M**achine **L**earning
MLR	**M**ultinomial **L**ogistic **R**egression
NB	**N**aïve **B**ayes
NN	**N**earest **N**eighbor
SVM	**S**upport **V**ector **M**achines
TN	**T**rue **N**egative
TP	**T**rue **P**ositive

Chapter 1

Introduction

Chapter 1. Introduction

Imagine a world where everyone can effortlessly make stylish and informed clothing choices. This is the vision we are bringing to life by combining the power of advanced AI, real-time weather data, and curated outfit selections. Our goal is to provide an intelligent and seamless experience for users seeking fashion inspiration and guidance. With personalized recommendations tailored to the weather conditions and the latest fashion trends, we make it simple for everyone to look their best every day. Whether it’s a sunny day, a rainy afternoon, or a chilly evening, our platform ensures that you have the perfect outfit for any occasion. Embrace the future of fashion with us, where technology and style converge to elevate your daily wardrobe choices.

1.1 Problem Definition

Achieving compatibility between formal and informal occasions requires understanding the distinct dress styles and clothing cultures. Formal Attire includes suits, ties, dress shoes, and polished pieces for a refined, professional look. Examples are business meetings, weddings, and formal dinners. Informal Attire comprises casual wear like jeans, t-shirts, and sneakers, focusing on personal expression and comfort. Balancing these styles involves blending elements from both, such as pairing a blazer with jeans for a smart casual look. Understanding the context and dressing appropriately ensures you look polished and suitable for any occasion.

Fig 1.1: Example about formal and informal occasions

Examples include casual outings, weekend wear, and informal gatherings [1]. Key Differences in Dress Styles: Material and Fabric: Formal wear often uses high-quality, structured fabrics such as wool, silk, and satin, which hold their shape and look crisp. Informal wear utilizes softer, more flexible materials like cotton, denim, and jersey for comfort. Design and Fit: Formal attire is usually tailored to fit perfectly, giving a sharp, clean silhouette. Informal clothing tends to have a looser fit, prioritizing comfort over precision. Color and Patterns: Formal clothing typically features muted, classic colors (black, navy, gray) and subtle patterns (pinstripes). Informal attire allows for brighter colors, bold patterns, and more creative designs.

Reconciliation Strategies: Smart Casual: This style blends elements of both formal and informal wear. For instance, pairing a blazer (formal) with chinos and a polo shirt (informal) achieves a balanced look suitable for various semi-formal occasions. Versatile Pieces: Incorporating versatile clothing items that can transition between formal and informal settings is key. For example, a well-fitted blazer can be dressed up with dress pants and a tie or dressed down with jeans and a t-shirt. Layering: Using layers to adjust the formality of an outfit can be effective. A casual base layer (t-shirt) can be elevated with a formal outer layer (blazer or coat). Accessories: Accessories can play a significant role in adjusting the formality of an outfit. A tie or pocket square can make an outfit more formal, while casual accessories like a baseball cap or sneakers can dress it down

Cultural Sensitivity: Contextual Dressing: Understanding the cultural context and expectations of each occasion is crucial. For example, what is considered formal in one culture may be seen as informal in another, Awareness and sensitivity to these nuances ensure appropriate dress choices [2]. Practical Tips: Know the Occasion: Always consider the specific requirements of the occasion. If in doubt, leaning slightly towards the formal side is generally safer. Invest in Key Pieces: Investing in high-quality, versatile pieces that can work in both formal and informal settings maximizes wardrobe utility. Personal Style: While reconciling different dress codes, it’s important to maintain personal style and comfort. Confidence in what you wear is a significant factor in looking and feeling good.

Choosing formal attire for an informal occasion is often not the best idea. Wearing a full suit and tie to a family gathering or a casual meet-up with friends might make you look overdressed and feel out of place. People in these settings usually prefer comfortable and relaxed clothing, so it's better to opt for casual wear that fits the environment, such as jeans and a shirt or a blouse [1]

Conversely, attending a formal event, like a job interview or a business meeting, in casual attire is also inappropriate. Showing up in jeans and a t-shirt can give off an impression of being unprofessional or not taking the occasion seriously. In such scenarios, it's essential to wear formal clothing to make a good impression and show respect for the occasion [1]

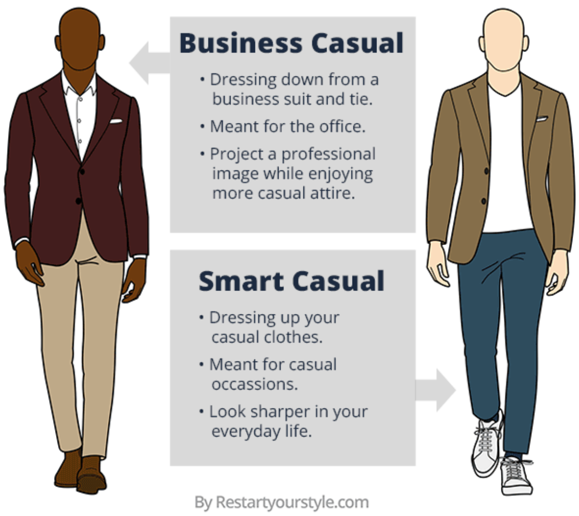

Fig 1.2: Example about formal and informal

1.2 Motivation

We address the significant time people often spend deliberating over their outfits, streamlining the process with our intelligent suggestions. Choosing what to wear can be a daunting task, especially when trying to match the outfit to the occasion and current fashion trends. Our platform simplifies this by providing smart, personalized recommendations, saving users valuable time and effort.

By staying abreast of the latest fashion trends, our platform eliminates the need for users to invest excessive time and effort in keeping up. Fashion trends are constantly evolving, and keeping track of what's in style can be overwhelming. Our platform continuously updates its recommendations to reflect the latest trends, ensuring users are always fashion-forward without the hassle of constant research.

We alleviate the common challenge of lacking confidence in selecting suitable outfits for different occasions, ensuring users feel confident and stylish. Many people struggle with choosing appropriate attire for various events, whether it's a formal meeting or a casual gathering. Our platform takes the guesswork out of this process, providing outfit suggestions that are not only suitable but also stylish, helping users to feel assured and look their best in any situation.

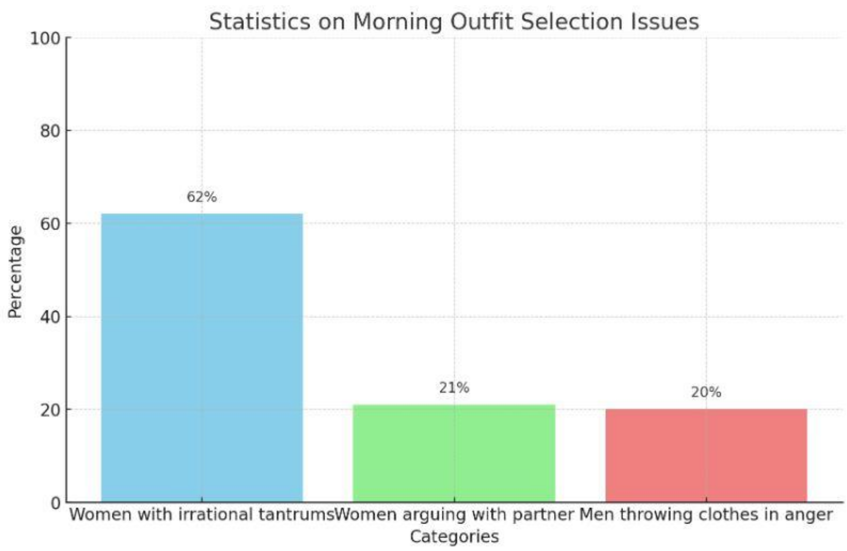

Fig 1.3: Statistics on the time spent choosing the right clothes

A study released by Marks & Spencer has revealed that the average woman will spend 17 minutes each morning trying to find an outfit to wear to work. This amounts to four days a year, and six months of our working lives searching for something to wear to the office. The poll of 2,000 men and women delivered some other interesting statistics. 62 per cent of women admitted to having 'irrational tantrums' during the daily hunt for an outfit, and 21 per cent confessed that the ordeal has caused them to have an argument with their partner. 20 per cent of men admitted to throwing clothes around the room in anger when they couldn't put together the perfect ensemble [2].

1.3 Objective

Our innovative concept aims to empower individuals to make informed and stylish clothing decisions effortlessly, integrating weather forecasts and color matching. Achieving compatibility between formal and informal occasions requires understanding the distinct dress styles and clothing cultures. Formal Attire includes suits, ties, dress shoes, and polished pieces for a refined, professional look. Examples are business meetings, weddings, and formal dinners. Informal Attire comprises casual wear like jeans, t-shirts, and sneakers, focusing on personal expression and comfort. Balancing these styles involves blending elements from both, such as pairing a blazer with jeans for a smart casual look. Understanding the context and dressing appropriately ensures you look polished and suitable for any occasion. Our solution bridges the gap between formal and informal occasions, accommodating diverse dress styles and clothing cultures seamlessly.

1.4 Organization Plan

Before the start of the project, a general time plan was made to work on it during the project period

Tasks	From	To	Time
Planning	15th Oct. 2023	30th Oct. 2023	6 weeks
Design	1th Nov. 2023	30th Nov. 2023	4 weeks
Development	1th Dec. 2023	30th Feb. 2024	8 weeks
Testing	1th Mar. 2024	30th Apr. 2024	8 weeks
Deployment	1th May. 2024	30th May. 2023	4 weeks

Tab 1.1: Project Time Plan

1.5 Documentation Organization

Chapter 2 (Literature Review) : In this chapter, the most important literature related to the subject of the clothes matching project

Chapter 3 (System Architecture and Algorithms) : In this chapter, we present the techniques that we used to implement all stages of the clothes matching

Chapter 4 (System Implementation and Results) : In this chapter, we present the experimental results for all phases of the App

Chapter 5 (Run the Application) : In this chapter, we present the process of running the application and displaying its features.

Chapter 6 (Conclusion and Future Work) : In this chapter, we present the conclusion of our findings and discuss potential future work for the application.

Chapter 2

Literature Review

Chapter 2. Literature Review

In this chapter, we will elucidate the system architecture designed for clothing matching utilizing Deep Learning Techniques. The system is structured into multiple stages, with each stage capable of incorporating a variety of techniques and algorithms. We will delve into each stage, highlighting the application of widely recognized and extensively used algorithms, drawing insights from previous projects and scholarly papers

The system architecture includes four stages: Pre-processing, Feature extraction, classification and finally application

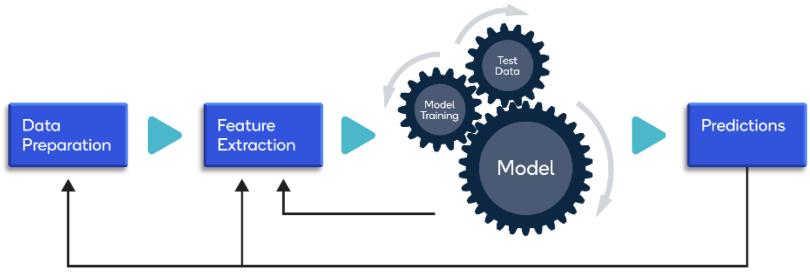

Fig 2.1: System Stage

2.1 Preprocessing

Techniques applied to raw data before it can be used for analysis or modeling. In the context of machine learning and deep learning, preprocessing plays a crucial role in preparing data to improve the performance and accuracy of models. Here are some common preprocessing steps:

2.1.1 Data Cleaning

This involves handling missing data, outliers, or noise that could adversely affect the model's performance. Techniques such as imputation (filling missing values) or removing outliers are used [3, 4]

• **Handling Missing Data**:

- Check for missing values in attributes such as item description, color, size, or price.
- Decide on appropriate strategies for filling missing values (e.g., using mean or mode values for numeric or categorical attributes).

• **Removing Duplicates**:

- Identify and remove duplicate entries of clothing items to ensure each item is unique and does not skew the recommendation results.

• **Standardizing Attributes**:

- Ensure consistency in attribute formats, such as standardizing color names (e.g., "navy blue”) and size formats (e.g., "S", "Small", "Size 8").

• **Handling Outliers**:

- Identify and investigate outliers in attributes such as price or measurements.
- Decide whether to remove outliers or apply transformations to mitigate their impact on the recommendation system.

2.1.2 Data Integration

Data integration involves combining and harmonizing data from multiple sources to create a unified and coherent dataset that can be used for matching and recommendation purposes. Here are some key aspects of data integration that are relevant for a clothes matching app [3, 4]

• **Data Source Integration**:

- **Vendor Data**: Integrate clothing data from various vendors or brands to provide a diverse selection of items for matching.
- **User Data**: Integrate user preferences and interaction data (e.g., likes, ratings, purchase history) to personalize recommendations.
- **Third-party APIs**: Integrate data from external sources such as fashion trend APIs, weather APIs (to suggest appropriate clothing based on weather conditions), or social media APIs (to incorporate social trends or influencers' recommendations).

• **Data Mapping and Alignment**:

- Ensure that data from different sources are mapped to a common schema or format. This includes standardizing attributes such as item

• **Data Quality Assurance**:

- Perform data validation and cleansing during the integration process to identify and address inconsistencies, duplicates, or errors in the data from different sources.

2.1.3 Data Transformation

Data transformation involves preparing and structuring the data in a way that optimizes it for analysis, modeling, and ultimately, providing accurate recommendations to users. Here are some key aspects of data transformation relevant to a clothes matching app

By performing these data transformation tasks effectively, a clothes matching app can enhance the accuracy, relevance, and personalization of clothing recommendations, ultimately improving user satisfaction and engagement with the platform [5]

• **Feature Extraction and Engineering**:

- **Textual Data**: Extract features from textual descriptions of clothing items, such as keywords, sentiment analysis, or style attributes (e.g., formal, casual).
- **Image Data**: Extract features from clothing images using techniques like convolutional neural networks (CNNs) to capture visual patterns, colors, and styles.
- **Metadata**: Extract and encode metadata such as item categories, sizes, colors, and prices into structured formats suitable for recommendation algorithms.

• **Normalization and Standardization**:

- Normalize numeric attributes like prices or sizes to a standard scale (e.g., between 0 and 1) to ensure uniformity and prevent certain features from dominating the model training process.
- Standardize categorical attributes by encoding them into numerical representations (e.g., one-hot encoding for item categories) to facilitate model training.

- **Aggregation and Combination**:

  - Aggregate data from multiple sources (e.g., vendor data, user preferences) to create comprehensive profiles of clothing items and users.
  - Combine different types of data (e.g., textual descriptions, image features, metadata) to enrich the dataset and provide a holistic view for recommendation system

2.1.4 Data Reduction

Data reduction techniques aim to streamline and optimize the dataset to improve the efficiency and effectiveness of the recommendation system. Here are some key strategies for data reduction specifically tailored for such applications [5]

1. **Dimensionality Reduction**:
   - **Principal Component Analysis (PCA)**: PCA is a widely used technique to reduce the dimensionality of data by transforming variables into a smaller set of principal components that retain as much variance as possible. In the context of a clothes matching app, PCA can be applied to features extracted from clothing images or textual descriptions to reduce computational complexity while preserving essential information.
2. **Feature Selection**:
   - **Filter Methods**: Use statistical techniques such as correlation analysis or Chi-square tests to identify and select the most relevant features that contribute significantly to the matching process. For instance, features related to color, style, and category that have the highest correlation with user preferences or purchase history could be prioritized.
   - **Wrapper Methods**: Employ algorithms like recursive feature elimination (RFE) or forward/backward selection, which iteratively evaluate subsets of features based on model performance metrics (e.g., accuracy, AUC) to identify the optimal set of features for the recommendation system.
   - **Embedded Methods**: Utilize machine learning algorithms with built-in feature selection mechanisms decision tree-based methods

3. **Sampling Techniques**:
   - **Stratified Sampling**: Ensure balanced representation of different categories or classes of clothing items in the dataset to prevent bias and improve model generalization.
   - **Random Sampling**: Reduce the size of the dataset by randomly selecting a subset of data instances while maintaining the overall distribution of features and labels.

2.1.5 Data augmentation

Data augmentation in a clothes matching app involves generating synthetic data or variations of existing data to enhance the diversity and richness of the dataset used for training and recommendation purposes. Here's how data augmentation can be applied specifically in this context [5]

• **Image Augmentation**:

- **Rotation**: Rotate clothing images by different angles (e.g., 90 degrees, 180 degrees) to simulate different viewpoints and orientations.
- **Flipping**: Horizontally flip clothing images to create mirror images, which helps the model generalize better to different orientations of clothing items.
- **Scaling and Resizing**: Increase or decrease the size of clothing images to simulate variations in zoom level or camera distance.
- **Color Jitter**: Apply slight variations in brightness, contrast, or hue to clothing images to simulate different lighting conditions or color shades.

• **Overlay and Composition**:

- **Background Variation**: Combine clothing images with different background scenes or settings to simulate diverse environmental contexts.
- **Accessory Addition**: Overlay or digitally add accessories such as bags, hats, or jewelry to clothing images to showcase complementary styling options.

• **Style Transfer**:

- Apply style transfer techniques to clothing images, allowing users to preview how an item might look in different artistic styles or fashion trends.

• **Data Synthesis**:

- Generate synthetic data by blending features from different clothing items to create hybrid designs or combinations that expand the range of available recommendations.

If you skip the data preprocessing step, it will affect your work later on when applying this dataset to a machine learning model. Most of the models can't handle missing values. Some of them are affected by outliers, high dimensionality and noisy data, and so by preprocessing the data, you'll make the dataset more complete and accurate. This phase is critical to make necessary adjustments in the data before feeding the dataset into your machine learning model.

2.2 Feature Extraction

The process of feature extraction plays a vital role in defining the discriminatory information carried by images, Extract highly efficient features in the Dataset. In order to achieve better accuracy and robustness, DL techniques needs not only a good algorithm but also a good input

2.2.1 Image Feature Extraction

Convolutional Neural Networks (CNNs): CNNs are particularly powerful for extracting hierarchical features from clothing images. Pre-trained CNN models like VGG, ResNet, or EfficientNet can be used to extract features from images effectively. Features extracted may include [5]

- Textures (e.g., patterns, fabric details)
- Shapes and contours (e.g., silhouette)
- Colors (e.g., dominant color palette)

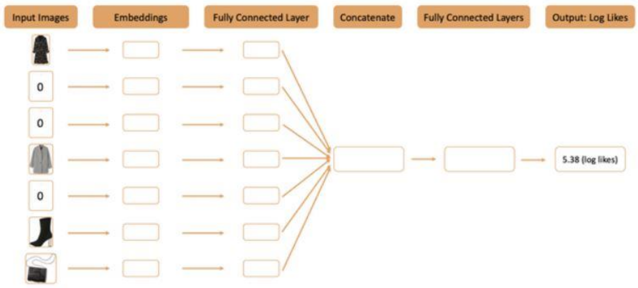

Fig 2.2: CNN Model

Transfer Learning: Utilize pre-trained CNN models trained on large-scale datasets (e.g., ImageNet) and fine-tune them on your specific clothing dataset to leverage learned representations effectively [6]

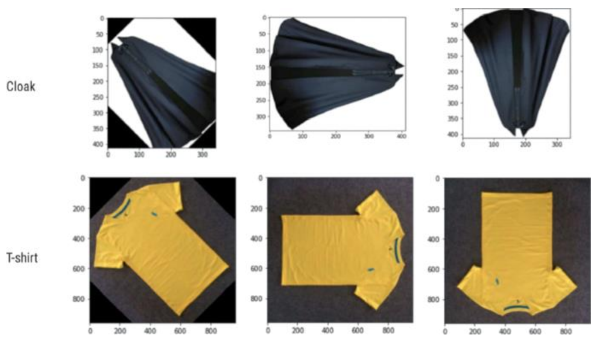

Fig 2.3: Transfer Learning

Histogram of Oriented Gradients (HOG): HOG can be used to extract shape and edge features from clothing images, particularly useful when fine-grained details like texture and patterns are important [6]

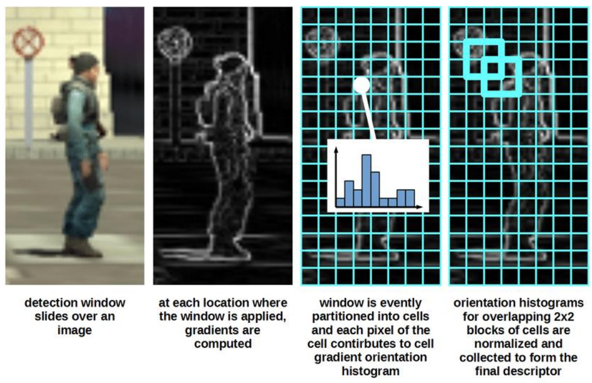

Fig 2.4: Histogram of Oriented Gradients

Attribute-GAN model: Our model can be regarded as an extension of cGAN, which learns a mapping from label *x* and random noise vector *z*, to *y: G: z → y*. Our proposed approach, Attribute-GAN, learns a mapping from a pair of outfits, conditioned on attributes of clothing. Both the generator network *G* and the discriminator network *D* perform feed-forward inference conditioned on clothing attributes [7]

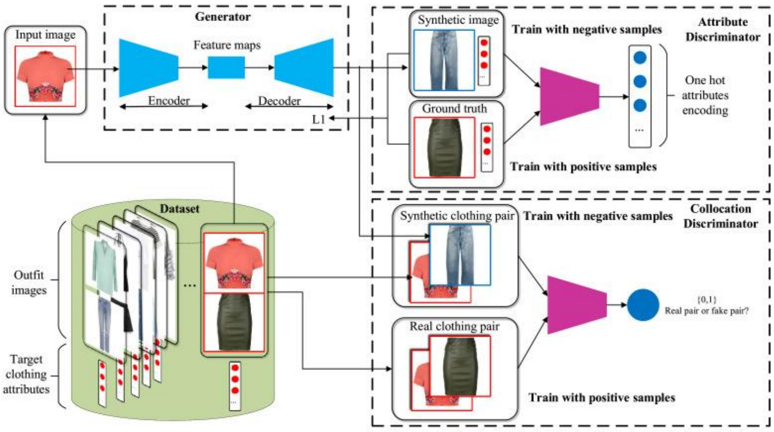

Fig 2.5**:** Attribute-GAN model

Siamese Networks: Siamese networks are specifically designed for tasks where similarity between pairs of inputs needs to be evaluated (e.g., matching similar clothing items). These networks learn embeddings (vector representations) of clothing images that emphasize their similarities and differences [8]

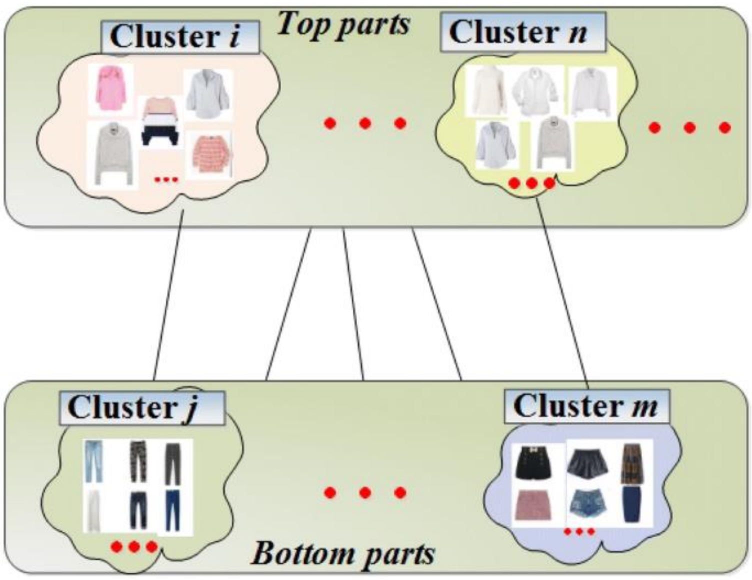

Fig 2.6: Siamese Networks

2.2.2 Metadata and Attribute Feature Extraction

• **One-Hot Encoding**: Convert categorical attributes such as clothing category (e.g., "shirt", "dress") or size (e.g., "small", "medium", "large") into binary vectors representing each category or size class.

• **Numeric Features**: Extract numerical attributes such as price, discount percentage, or popularity scores directly as features, which can complement image and text-based features in the recommendation process [5]

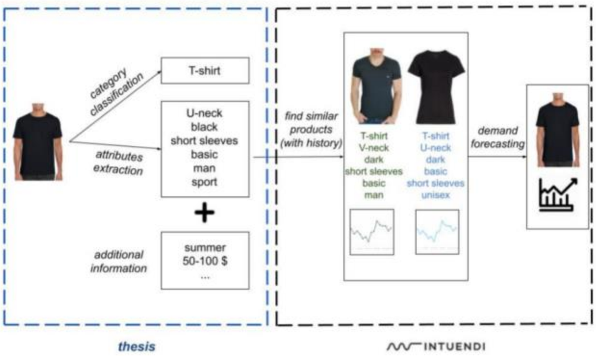

Fig 2.7: Metadata

2.2.3 Implementation Considerations

• **Dataset Size and Diversity**: Large and diverse datasets of clothing images are crucial for training deep learning models effectively. Curate or collect datasets that encompass various clothing styles, colors, textures, and sizes to improve the generalization capability of your models.

• **Evaluation Metrics**: Use appropriate evaluation metrics (e.g., cosine similarity, Euclidean distance) to assess the quality of extracted features in terms of their ability to accurately represent and distinguish between different clothing items.

Fig 2.8: Dataset Size and Diversity

2.3 Classification

The process of the information extracted from the image will be then fed into classifier to map different patterns and match them appropriately. Classifiers should be deployed to distinguish different categories of the features extracted

Crucial task where items of clothing are categorized into different classes or labels. This can include categories like "shirts," "pants," "dresses," and subcategories such as "casual," "formal," or "sportswear." Classification helps in organizing the clothing items effectively and improving the recommendation system. Here’s how you can approach classification

2.3.1 Traditional Machine Learning

Machine learning algorithms are built to “learn” to do things by understanding labelled data, then use it to produce further outputs with more sets of data. However, they need to be retrained through human intervention when the actual output is not the desired one.

Techniques can be effectively applied to various tasks within a clothes matching app, including classification, recommendation, and feature extraction. Here’s an overview of how traditional machine learning

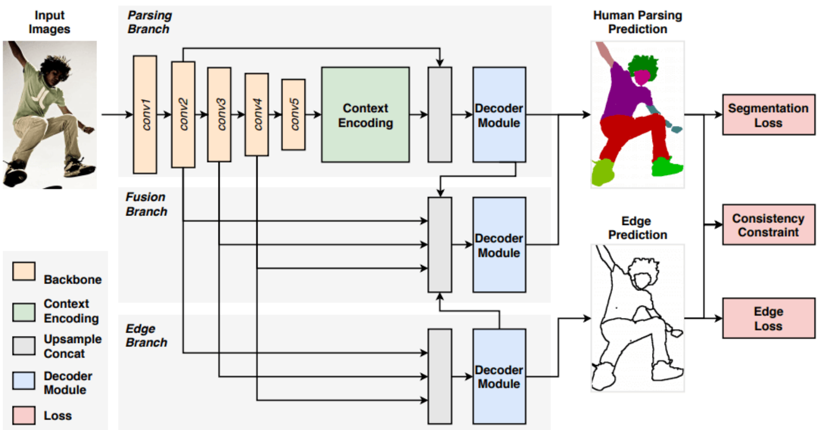

Fig 2.9: Machine Learning Architecture

Support Vector Machines (SVM):

A support vector machine (SVM) is a multi-class classifier that has been successfully applied in many disciplines. The SVM algorithm gained its success from its excellent empirical performance in applications with relatively large numbers of features [9]

- **Application**: SVMs can be used to classify clothing images based on extracted features. They work well for high-dimensional data and can handle non-linear classification tasks using kernel functions.
- **Example**: Classifying clothing items into categories like "casual" or "formal."

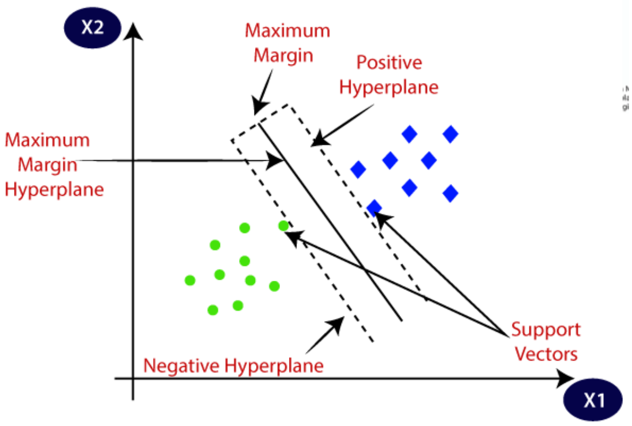

Fig 2.10: Support Vector Machines Architecture

K-Nearest Neighbors (k-NN):

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand but has a major drawback of becoming significantly slows as the size of that data in use grows [10]

- **Application**: k-NN is a simple, instance-based learning algorithm that can be used for classification by comparing the feature vectors of new clothing items to those in the training set.
- **Example**: Identifying the category of a new clothing item by finding the most similar items in the existing dataset.

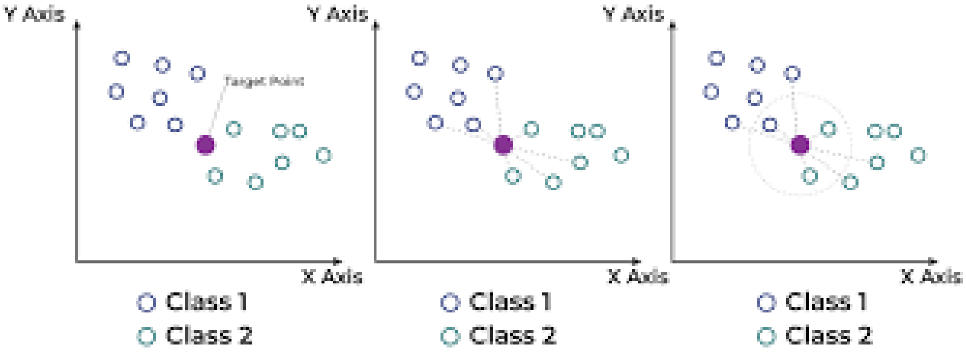

Fig 2.11: K-Nearest Neighbors Architecture

Decision Trees:

A decision tree is a popular machine learning algorithm used for classification and regression tasks. It is a tree-like structure where each internal node represents a decision based on a feature, each branch represents the outcome of the decision, and each leaf node represents a class label (in classification) or a continuous value (in regression) [11]

- **Application**: Decision trees can be used to make decisions based on feature values, which is useful for classifying clothing items based on various attributes like color, style, and fabric.
- **Example**: Classifying items into "summer wear" or "winter wear" based on features like material and color.

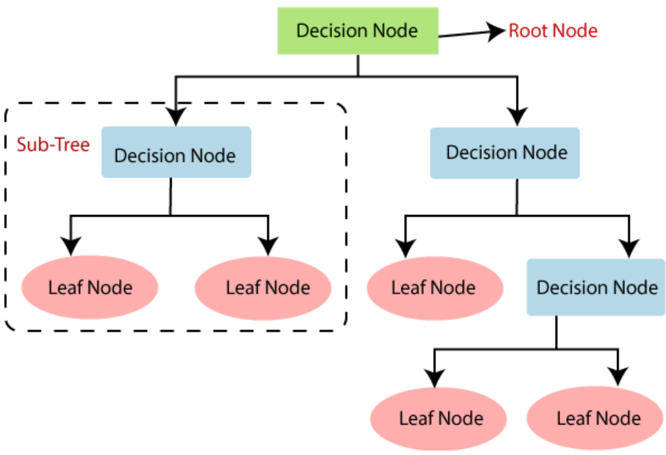

Fig 2.12: Decision Trees Architecture

Random Forests:

Random forest is a Supervised Machine Learning Algorithm that builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification [12]

- **Application**: An ensemble method that combines multiple decision trees to improve classification accuracy and handle overfitting.
- **Example**: Categorizing clothing items into multiple categories with high accuracy by leveraging the robustness of multiple decision trees

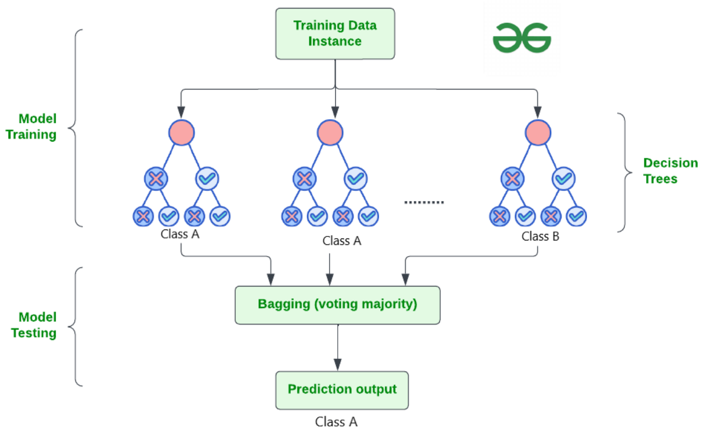

Fig 2.13 Random forest Architecture

2.3.2 Deep Learning Models

Deep learning techniques play a crucial role in the development of a sophisticated and effective clothes matching app. These techniques can be applied to various aspects such as image classification, feature extraction, similarity measurement, and recommendation systems. Here are some deep learning

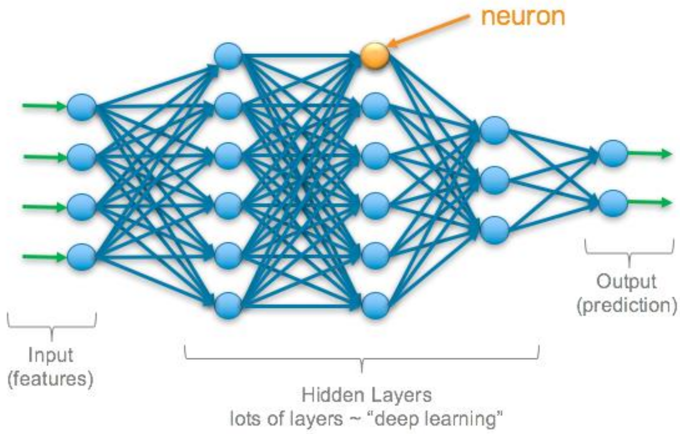

Fig 2.14 Deep Learning Architecture

CNN (Convolutional Neural Network)

A Convolutional Neural Network (CNN) is a type of deep learning model particularly well-suited for analyzing visual data, such as images and videos. CNNs are widely used for various tasks in computer vision, including image classification, object detection, and image segmentation [13]

Key Components of CNN:

1. **Convolutional Layers**:
   - **Convolution Operation**: This involves applying a filter (or kernel) over the input image to produce a feature map. The filter slides over the input image, performing element-wise multiplication and summing the results to detect patterns such as edges, textures, and shapes.
   - **Activation Function**: After the convolution operation, an activation function (typically ReLU) is applied to introduce non-linearity into the model, allowing it to learn more complex patterns.

2. **Pooling Layers**:
   - **Max Pooling**: This operation reduces the spatial dimensions (height and width) of the feature map by taking the maximum value from a defined window, typically 2x2 or 3x3. This helps in reducing the number of parameters and computational load while retaining important features.
   - **Average Pooling**: Similar to max pooling, but instead of taking the maximum value, it takes the average value within the window.
3. **Fully Connected Layers (Dense Layers)**:
   - These layers come at the end of the network and are used to combine the features extracted by the convolutional and pooling layers to make a final prediction. They connect every neuron in one layer to every neuron in the next layer.
4. **Flatten Layer**:
   - Before the fully connected layers, the multi-dimensional output from the convolutional and pooling layers is flattened into a one-dimensional vector.
5. **Output Layer**:
   - This layer provides the final prediction. For classification tasks, it often uses a softmax activation function to output probabilities for each class.

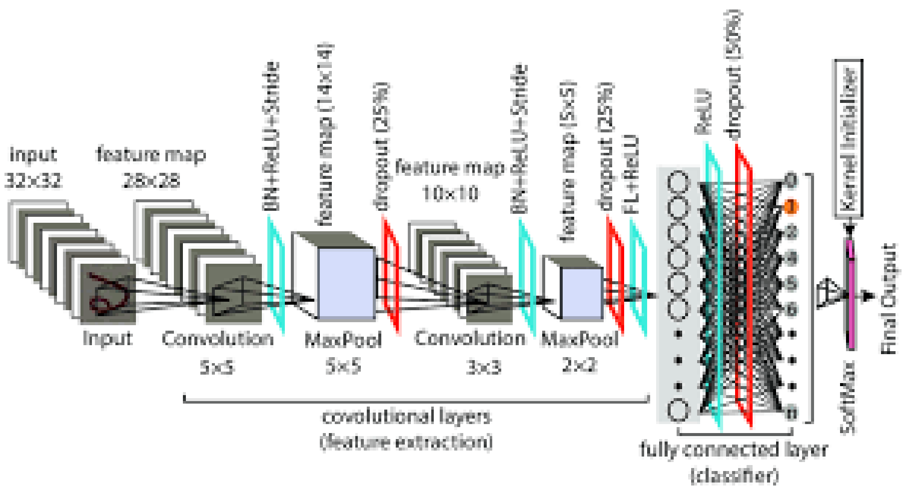

Fig 2.15 CNN Architecture

VGGNet

VGGNet, developed by the Visual Graphics Group at Oxford University, is a convolutional neural network (CNN) model that became famous for its performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014. It is known for its simplicity and depth, utilizing very small (3x3) convolution filters in all layers. The most popular configurations of VGGNet are VGG16 and VGG19, which have 16 and 19 weight layers, respectively.

The architecture of VGGNet is characterized by its use of small convolution filters (3x3) with a stride of 1 and padding of 1 to preserve the spatial resolution of the input. This allows the network to capture more fine-grained details in the images. Additionally, VGGNet uses max-pooling layers (2x2) with a stride of 2 to progressively reduce the spatial dimensions of the feature maps, thereby reducing the computational load and the number of parameters.

In VGGNet, the convolutional layers are followed by three fully connected layers: the first two have 4096 channels each, and the third performs the final classification. The activation function used throughout the network is the Rectified Linear Unit (ReLU), which introduces non-linearity and helps in training deep networks. The final layer uses the softmax activation function to output probabilities for each class.

A typical VGGNet configuration (e.g., VGG16) includes multiple convolutional layers followed by pooling layers, with an increasing number of filters as we move deeper into the network (e.g., 64 filters in the first layers, then 128, 256, and so on). This design choice allows the network to learn more complex and abstract features at each layer [14]

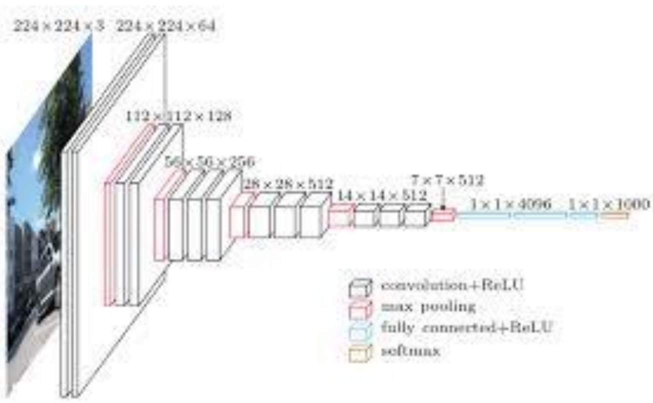

Fig 2.16 VGGNet Architecture

Siamese Networks

Siamese Networks are a type of neural network architecture that is designed to learn the similarity between pairs of inputs by computing a meaningful distance metric between them. Unlike traditional neural networks that classify inputs directly, Siamese Networks take two inputs and compare them to determine how similar or different they are. This architecture is particularly useful for tasks like image recognition, signature verification, and one-shot learning where the goal is to recognize or verify similarities between instances.

A Siamese Network consists of two identical subnetworks, each taking one of the two inputs. These subnetworks share the same architecture and weights, ensuring that the features they extract are directly comparable. After passing the inputs through their respective subnetworks, the outputs are combined using a distance function, such as Euclidean distance or cosine similarity, to measure the similarity between the two inputs.

Siamese Networks are highly effective in scenarios where there are many categories with few examples each. For example, in a clothing matching app, a Siamese Network can be used to compare different clothing items to find those that are visually similar or complementary. By training the network with pairs of clothing items labeled as matching or non-matching, it can learn to identify subtle similarities in patterns, colors, and styles that might not be apparent through traditional classification approaches [15]

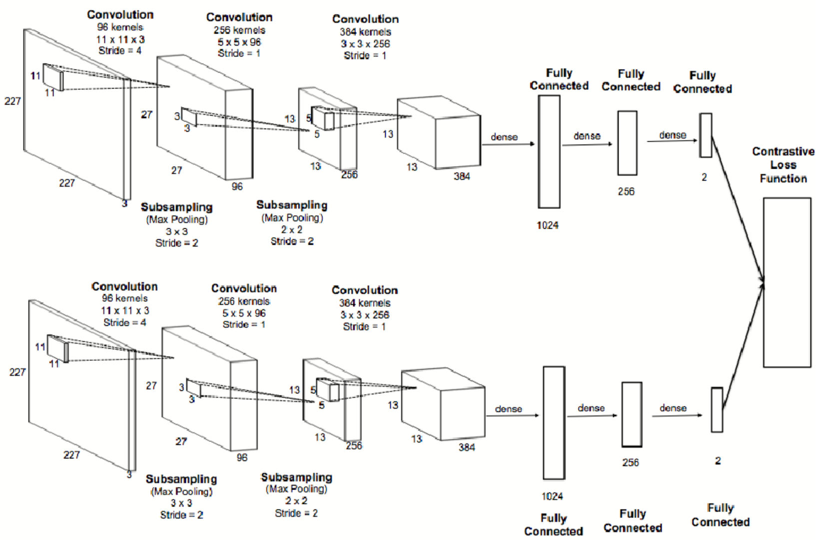

Fig 2.17 Siamese network Architectur

YOLO (You Only Look Once)

State-of-the-art real-time object detection system. Unlike traditional object detection algorithms that rely on region proposal mechanisms and subsequent classification, YOLO frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. This approach allows YOLO to predict bounding boxes and class probabilities directly from full images in one evaluation. Therefore, YOLO is extremely fast, making it suitable for real-time applications such as autonomous driving, surveillance, and robotics.

The key idea behind YOLO is its unified model that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. By dividing the input image into an S x S grid, each grid cell is responsible for predicting a fixed number of bounding boxes, along with confidence scores indicating the presence of an object and class probabilities. This design enables YOLO to achieve high accuracy and speed by optimizing object detection and classification jointly.

YOLO has undergone several iterations (YOLOv1, YOLOv2, YOLOv3, and YOLOv4), each improving upon the previous version in terms of accuracy, speed, and robustness. These improvements include techniques such as batch normalization, anchor boxes, feature pyramid networks, and enhanced training strategies to handle a wide variety of objects in diverse environments effectively [16]

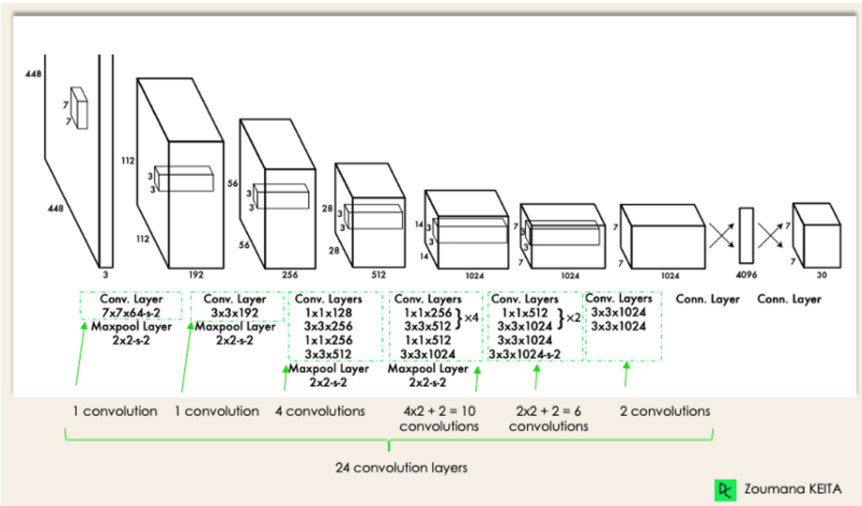

Fig 2.18 Yolo Architecture

Common Components of YOLO:

- **Anchor Boxes**: Used to predict bounding box shapes and sizes.
- **Darknet Architecture**: A custom neural network architecture designed for YOLO, known for its simplicity and efficiency.
- **Non-Maximum Suppression (NMS)**: Post-processing technique used to filter redundant overlapping bounding boxes.
- **Training Strategies**: YOLO models are typically trained on large-scale datasets like COCO (Common Objects in Context) to generalize well across various object categories and environmental conditions.
- **Real-time Capability**: YOLO models are optimized for real-time object detection tasks, making them suitable for applications in autonomous vehicles, surveillance systems, and robotics.

Each version of YOLO has built upon its predecessor, incorporating advancements in deep learning and computer vision research to achieve better performance in terms of accuracy, speed, and robustness for real-world applications.

2.4 Color Matching Algorithm

Color matching algorithms are essential in fashion recommendation systems, helping users select clothing items that complement each other. These algorithms use various techniques including color theory, computer vision, and machine learning [17] Color Theory This involves principles such as complementary colors (opposite on the color wheel, e.g., blue and orange) for high contrast, analogous colors (next to each other, e.g., blue, green, cyan) for harmony, and triadic colors (evenly spaced, e.g., red, blue, yellow) for vibrant combinations Color Spaces RGB (Red, Green, Blue) is common for digital displays, while HSV/HSL (Hue, Saturation, Value/Brightness) aligns more with human color perception. Lab color space is used for precise matching as it approximates human vision Color Harmony Models Algorithms suggest pleasing combinations based on established harmony rules Euclidean Distance in Color Space Measures the distance between colors to determine their match.

Fig 2.19 Color matching algorithm

2.5 Object Detection

Object detection and clothes matching applications use advanced image recognition and machine learning to help users select complementary clothing items. These applications typically use several key techniques [18]

Object Detection Techniques like YOLO (You Only Look Once) and Faster R-CNN (Region Convolutional Neural Networks) detect and identify clothing items in images quickly and accurately Clothing Feature Extraction Convolutional Neural Networks (CNNs), such as ResNet, extract features from clothing images, capturing details like color, texture, and patterns Color Matching Color theory principles, such as complementary and analogous colors, help in suggesting visually appealing combinations. Different color spaces (e.g., HSV, Lab) are used for better color matching.

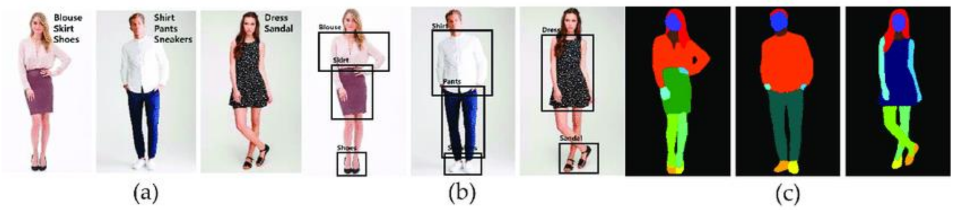

Fig 2.20 Object detection

Objective	Methodology	Title	Year	
Attribute-GAN enhances clothing coordination via semantic attributes, demonstrating superior effectiveness compared to alternatives.	Attribute-GAN in cGAN framework	Toward AI fashion design	2019	**1**
Proposing a vision-based tech for blind-friendly clothes matching, robust to variations. Evaluated on a challenging database, results conveyed through audio outputs.	Develop computer vision method for clothes coordination using wearable camera.	Clothes matching for blind	2010	**2**
StyleIt aims seamless fashion experience, integrating Apple Pay, adapting trends.	computer vision	A Fashion App for Personalized Outfit	2014	**3**
learn latent style features from a large dataset of user-created "style sets," where a style set is a collection of garments and accessories that make up a single outfit.	Style2Vec Word2Vec	Fashion Items from Style Sets	2017	**4**

Table 2.1 Comparison between Clothes matching app

Chapter 3

System Architecture And Algorithms

Chapter 3. System Architecture and Algorithms

The proposed deep learning approach contains two main phases: preprocessing, and deep learning as shown in Figure 3.1

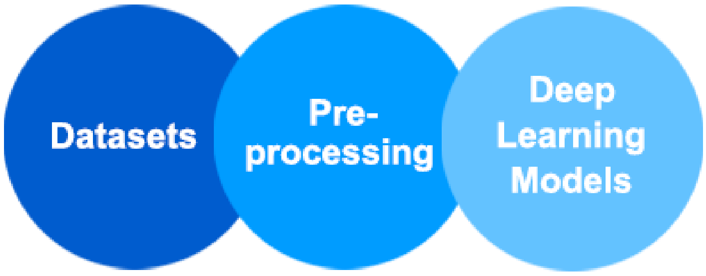

Fig 3.1 Deep learning proposed approaches architecture

3.1 Preprocessing

Preprocessing in the context of a clothes matching application involves a series of steps designed to prepare clothing images for further analysis and matching. It is a crucial phase in the development of any computer vision application, as it ensures that the input data is clean, standardized, and ready for the next stages of the processing pipeline.

The primary purpose of preprocessing is to enhance the quality of the images and extract meaningful features that can be used by machine learning models to accurately compare and match clothing items. By doing this, we can improve the performance and accuracy of the clothes matching application.

- background removal

  In the context of the provided Python code, a filter refers to the various processes and operations applied to the image data to achieve background removal. Firstly, the GrabCut algorithm uses statistical filters based on Gaussian Mixture Models (GMMs) to differentiate between foreground and background pixels. These filters are applied iteratively to refine the segmentation. Secondly, the binary mask creation step uses a logical filter to classify pixels as either foreground or background by setting probable and definite background pixels to 0 and probable and definite foreground pixels to 1. Thirdly, the binary mask is used to filter the original image, isolating the foreground. This step applies the binary mask to the image, effectively filtering out the background. Fourthly, a filtering operation is used to change all non-black pixels in the background to white, ensuring that any remaining artifacts or shadows in the background are neutralized by filtering them out and replacing them with white pixels.

In image processing, a filter typically refers to an operation or function applied to the image data to achieve a specific effect or transformation. Filters can be used for various purposes such as enhancing image quality, detecting edges, blurring, sharpening, and segmenting regions of interest. Convolutional filters are used in convolutional neural networks (CNNs) and image processing to detect features such as edges, textures, and patterns by sliding a kernel (small matrix) over the image. Logical/thresholding filters are used to create binary images by applying a condition to the pixel values, such as setting pixels above a certain threshold to one value and the rest to another [19]

- Image LANCZOS

Advanced resampling filter used for resizing images with superior quality preservation. It employs a weighted average of neighboring pixels based on a sinc function, which helps maintain sharpness and minimize artifacts such as aliasing. This filter is particularly effective for both reducing (down sampling) and increasing (up sampling) image sizes, ensuring high fidelity and clarity compared to simpler interpolation methods like bilinear or nearest-neighbor. Its versatility makes it suitable for various types of images, including photographs and detailed graphics, where preserving fine details is crucial [20]

- Image EXPAND

prepares the image as a batch by adding an extra dimension to the array. This converts the image from a 3D array (height x width x channels) to a 4D array (1 x height x width x channels), which is typical for batch processing in machine learning models. This transformation allows the image to be processed effectively in batch mode, which is beneficial for training and inference in machine learning tasks.

3.2 Deep Learning Models

Deep learning is a subset of machine learning that involves neural networks with many layers, called deep neural networks. These networks are designed to automatically and adaptively learn hierarchical features from data, such as images, text, and audio. Deep learning excels at handling large and complex datasets, making it ideal for tasks like image recognition, natural language processing, and autonomous driving.

In the context of a clothes matching application, deep learning plays a crucial role. It can analyze and understand the visual features of clothing items, such as color, pattern, texture, and style. Convolutional Neural Networks (CNNs) are particularly effective for image-related tasks, as they can automatically extract and learn these features from clothing images.

The benefits of deep learning in a clothes matching application include improved accuracy in identifying and categorizing clothing items, the ability to suggest outfits based on user preferences and trends, and enhanced user experience through personalized recommendations. By leveraging deep learning, the application can provide more relevant and stylish outfit combinations, helping users to coordinate their clothes effortlessly

3.2.1Convolutional Neural Network

Convolutional Neural Networks (CNNs) are pivotal in deep learning for their proficiency in processing and analyzing visual data such as images and videos. They excel by capturing intricate spatial relationships and hierarchical patterns within the data efficiently. CNNs incorporate several critical components seamlessly integrated within their architecture: Convolutional Layers: These layers operate by applying learnable filters (kernels) to the input data. Each filter performs convolution across the input, extracting distinctive features that represent various aspects of the image, such as edges, textures, or shapes.

Pooling Layers: Following convolution, pooling layers play a crucial role in reducing the spatial dimensions of the feature maps produced by the convolutional layers. Techniques like max pooling help maintain essential features while reducing computational complexity and overfitting.

Activation Functions: CNNs typically utilize activation functions such as ReLU (Rectified Linear Unit), which introduce non-linearity into the network. This allows the model to learn and represent complex relationships in the data, enhancing its ability to generalize and make accurate predictions. Fully Connected Layers: Towards the end of the CNN architecture, fully connected layers assimilate and process the features extracted by the convolutional and pooling layers. These layers enable the network to perform tasks such as classification or regression by combining the learned representations from earlier layers.

CNNs have transformed various computer vision applications, including image classification, object detection, and semantic segmentation. Their capability to automatically learn hierarchical features from raw data has propelled advancements in fields like medical imaging, autonomous driving, and robotics, demonstrating their versatility and impact across diverse industries.

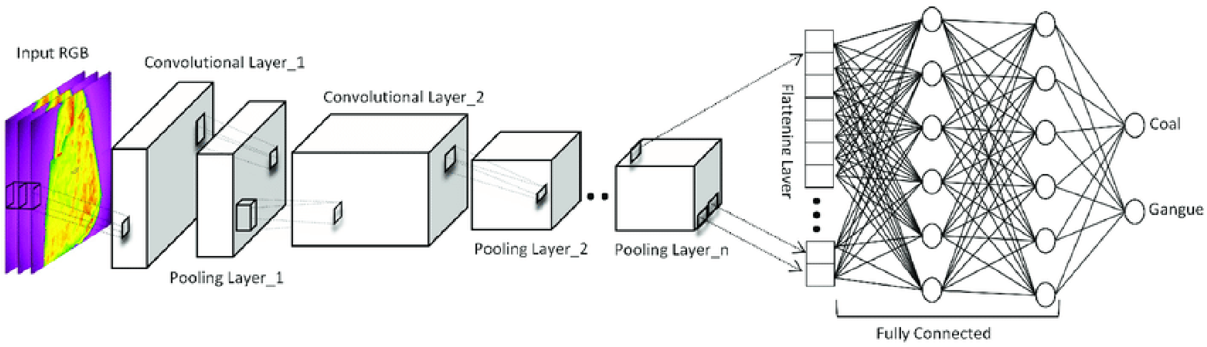

Figure 3.2 Convolution Neural Network Architecture

In this study, a Two-Dimensional Convolutional Neural Network (2D CNN) is considered for the task at hand. Figure 3.2 shows the architecture of the proposed 2D CNN. The convolutional layer creates filters that pass over the two-dimensional input image (RGB) to produce the output tensor The proposed CNN model consists of multiple basic blocks, each comprising two convolutional layers followed by one max pooling layer. The filter sizes in these blocks may vary (e.g., 32, 64, 128, and 256). A dropout layer is added with a rate of 50% to prevent overfitting during the training phase Finally, a flattening layer is used to convert the 2D feature maps into a 1D feature vector, which serves as input to the fully connected layers. The fully connected layer produces the output classes (e.g., matching clothing categories). The SoftMax activation function is utilized to classify the input image according to the highest probability The best results are achieved by training the model for a sufficient number of epochs (e.g., 200 epochs) using the ReLU activation function and the Adam optimizer with a learning rate of 0.0001 In this clothing matching application, the network identifies and matches clothing items by learning and extracting relevant features from the input images through the convolutional and pooling layers, eventually making accurate predictions through the fully connected layers 4o

CNNs, or Convolutional Neural Networks, have several advantages that make them particularly effective for image-related tasks. Firstly, they automatically detect important features without human supervision. Secondly, CNNs are highly effective at capturing spatial hierarchies in images due to their convolutional layers. Thirdly, they reduce the number of parameters compared to fully connected networks, which makes them more efficient and less prone to overfitting. Fourthly, CNNs can handle large amounts of data and are scalable. Lastly, they are robust to variations in the input, such as translation, rotation, and scaling, making them suitable for various applications, including image recognition, object detection, and segmentation [13]

3.2.2 YOLO

YOLO, which stands for "You Only Look Once," is a state-of-the-art deep learning algorithm used for object detection. Unlike traditional object detection methods that involve a two-stage process (first generating region proposals and then classifying each region), YOLO approaches the problem as a single regression problem. It directly predicts bounding boxes and class probabilities from full images in one evaluation, making it extremely fast and efficient.

The YOLO algorithm divides the input image into a grid, with each grid cell responsible for predicting a certain number of bounding boxes and their associated class probabilities. Each bounding box is defined by four coordinates (x, y, width, and height) and a confidence score that indicates the likelihood of the box containing an object. The class probabilities represent the likelihood of the detected object belonging to a particular class.

The core of YOLO's architecture is a convolutional neural network (CNN) that processes the entire image in a single pass. This end-to-end learning approach allows YOLO to consider the global context of the image, improving its ability to detect objects accurately and reducing the number of false positives and false negatives. The CNN extracts high-level features from the image, such as edges, textures, and shapes, which are then used to predict the bounding boxes and class probabilities.

One of the major advantages of YOLO is its speed, which enables real-time object detection. This is particularly useful in applications like autonomous driving, video surveillance, and robotics, where fast and accurate object detection is crucial. YOLO's efficiency comes from its streamlined architecture, which eliminates the need for region proposal networks and other complex components found in traditional object detection systems.

YOLO has gone through several iterations, with YOLOv8 being the latest version. YOLOv8 introduces several improvements over its predecessors, including enhanced accuracy and speed, better handling of small objects, and more robust training procedures. It uses a more refined CNN architecture with additional layers and advanced techniques like anchor-free detection and mosaic data augmentation. These advancements make YOLOv8 even more powerful

In summary, YOLO is a powerful deep learning algorithm that redefines object detection by treating it as a single regression problem. Its end-to-end approach, real-time detection capability, and successive improvements across versions, particularly in YOLOv8, make it an indispensable tool for various applications requiring fast and accurate object detection.

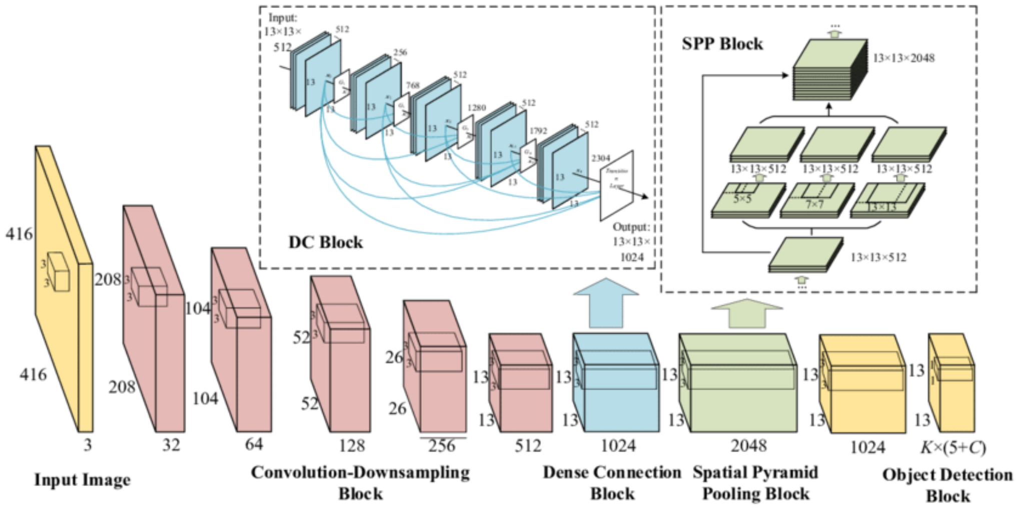

Figure 3.3 Convolution Neural Network Architecture

3.2.3 VGGNet

VGG Net, developed by the Visual Geometry Group at Oxford, is a type of Convolutional Neural Network (CNN) known for its deep architecture and simplicity. It uses very small (3x3) convolution filters throughout the network, which allows for capturing fine details in images. The network's architecture focuses on simplicity and depth by stacking multiple convolutional layers with small receptive fields, followed by max-pooling layers to progressively reduce the spatial dimensions and increase the depth of the feature maps. This hierarchical structure enables the network to learn complex patterns and features in the input images. The VGG Net typically ends with a few fully connected layers and a softmax layer for the final classification, making it suitable for various image recognition tasks.

VGG16 is a specific implementation of the VGG Net architecture with 16 weight layers, consisting of 13 convolutional layers and 3 fully connected layers. Here's a detailed breakdown of VGG16:

1. **Convolutional Layers**: The network starts with two convolutional layers, each having 64 filters of size 3x3. These layers are followed by a max-pooling layer. This pattern is repeated with an increasing number of filters:
    - Two convolutional layers with 128 filters, followed by a max-pooling layer.
    - Three convolutional layers with 256 filters, followed by a max-pooling layer.
    - Three convolutional layers with 512 filters, followed by a max-pooling layer.
    - Three more convolutional layers with 512 filters, followed by a final max-pooling layer.
2. **Max-Pooling Layers**: These layers reduce the spatial dimensions (height and width) of the feature maps by taking the maximum value in each window, thereby retaining the most important features while reducing computational complexity.
3. **Fully Connected Layers**: After the convolutional and max-pooling layers, the network includes three fully connected layers. The first two fully connected layers have 4096 neurons each, and the third fully connected

layer has 1000 neurons, corresponding to the number of classes in the ImageNet dataset.

4. **Softmax Layer**: The final layer is a softmax activation layer that outputs probabilities for each of the 1000 classes, helping to classify the input image into one of the predefined categories.

VGG16 is renowned for its depth and simplicity, using only 3x3 convolutional filters and maintaining a consistent architecture throughout the network. This consistency makes it easier to implement and extend. Despite its simplicity, VGG16 has achieved state-of-the-art performance on image classification benchmarks like ImageNet, making it a popular choice for various computer vision applications. Its pre-trained weights are also widely used for transfer learning in many other tasks, further demonstrating its versatility and effectiveness.

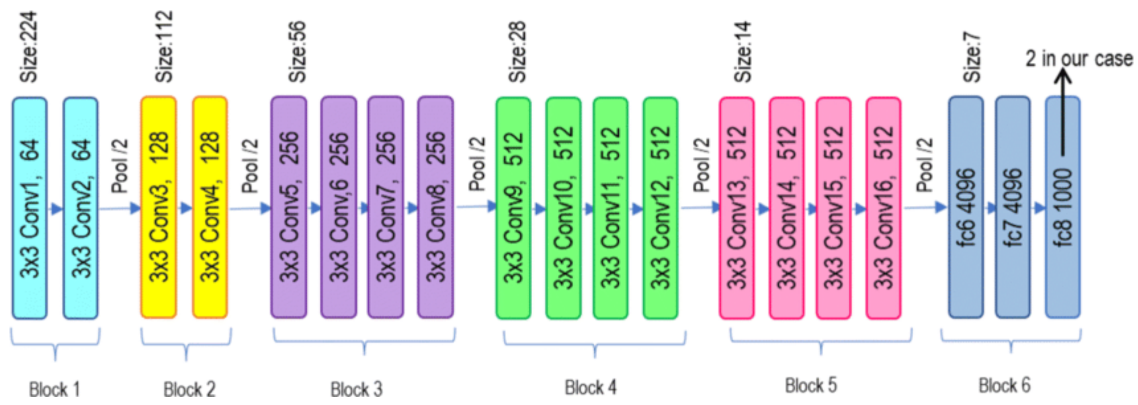

Figure 3.4 Very Deep Convolutional Networks Architecture

Chapter 4

System Implementation and Results

Chapter 4. System Implementation and Results

Experiments have been conducted for both approaches. The aim of this work is not only to achieve high overall effectiveness but also to enhance sensitivity, specificity, and precision in fashion coordination. Therefore, evaluations have been based on four key criteria derived from the style matrix: overall coordination and average coordination.

Each criterion is derived from the style matrix and calculated as follows:

- **Overall Coordination**: Measures the correct prediction of coordinated outfits.

  Overall Coordination = (TP+TN) **/** (TP+TN+FP+FN)

- TP (True Positive): Correctly predicted coordinated outfits.
- TN (True Negative): Correctly predicted non-coordinated outfits.
- FP (False Positive): Incorrectly predicted coordinated outfits.
- FN (False Negative): Incorrectly predicted non-coordinated outfits.

- **Average Coordination**: Calculates the average accuracy of each outfit style.

These metrics help ensure effective outfit coordination and styling in fashion programs.

4.1 Datasets

Object detection clothes

- Dataset size : 2682 image
- 10 classes
- Train : 2145 train
- Test : 537

Category	Instances	Precision	Recall	mAP50	mAP50-95
All	2035	0.734	0.752	0.770	0.506
Sunglass	82	0.862	0.229	0.377	0.120
Hat	77	0.740	0.766	0.771	0.458
Jacket	181	0.810	0.740	0.832	0.621
Shirt	366	0.765	0.792	0.814	0.577
Pants	114	0.685	0.965	0.957	0.734
Shorts	107	0.787	0.794	0.810	0.516
Skirt	186	0.675	0.849	0.820	0.625
Dress	128	0.545	0.883	0.784	0.559
Bag	274	0.703	0.679	0.727	0.387
Shoe	520	0.763	0.817	0.811	0.460

Figure 4.1 YOLO dataset categories

Metric	Value
Precision (B)	0.7336
Recall (B)	0.7520
mAP50 (B)	0.7696
mAP50-95 (B)	0.5057
Fitness	0.5321
Speed (preprocess)	0.4028 ms
Speed (inference)	12.8119 ms
Speed (loss)	0.0011 ms
Speed (postprocess)	1.1611 ms
Average Precision (AP)	
- Sunglass	0.1217
- Hat	0.4566
- Jacket	0.6205
- Shirt	0.5778
- Pants	0.7338

Figure 4.2 YOLO dataset categories

Outfits

- Dataset size : 1000 image
- 2 classes
- Formal and informal

4.2 Experimental Result of Deep Learning Approaches

The Three deep learning models: Convolutional neural networks (CNN), Yolo, VGG Network the experiments have been carried out on all deep learning models

4.2.1 YOLO

YOLO is a state-of-the-art model for object detection used in various applications, including fashion. In a clothing matching application, YOLO can accurately identify each garment in an outfit, such as shirts, pants, and accessories. By detecting and classifying these items in real-time, YOLO enables the application to provide users with complete outfit suggestions and personalized fashion advice, enhancing the overall shopping and styling experience.

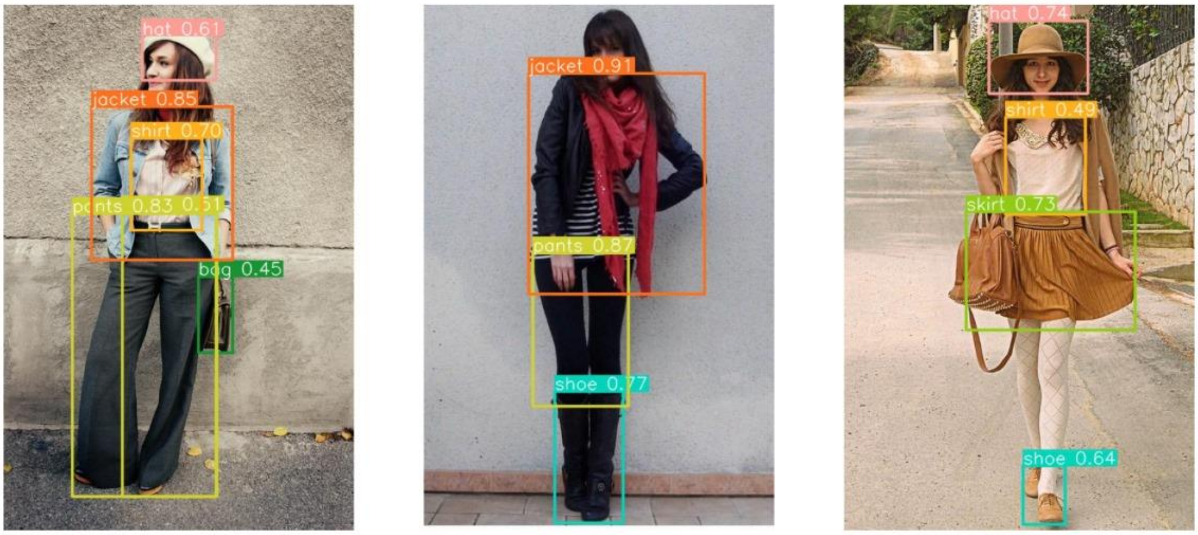

Figure 4.3 Outfit object detection

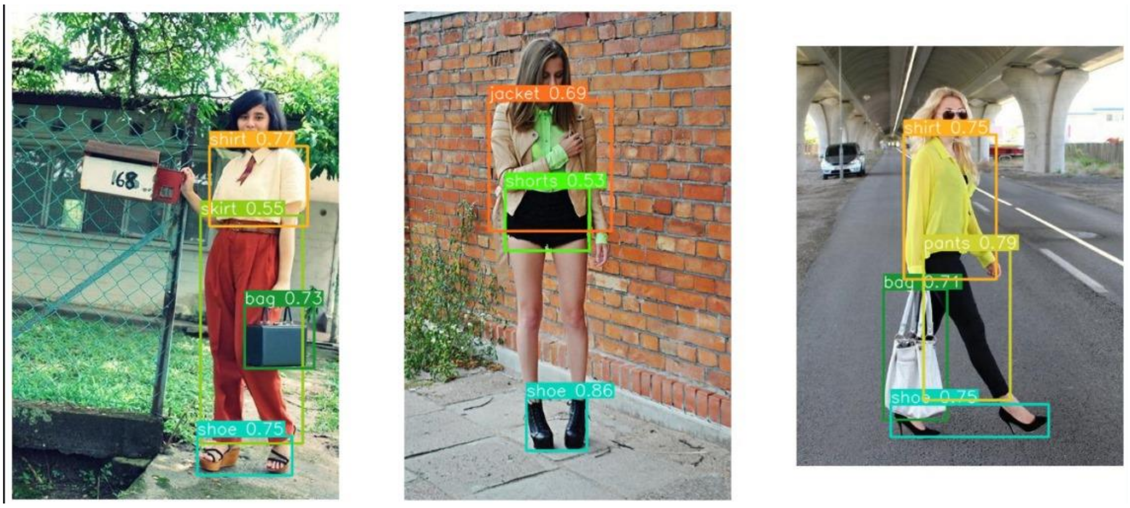

Figure 4.4 Outfit object detection

Epoch	Learning Rate	Momentum	Batch Size	Preprocessing	Box Loss	Class Loss	DFL Loss	Test Accuracy (mAP50-95)
1	0.000714	0.9	32	Applied	1.306	1.817	1.433	0.395
2	0.000714	0.9	32	Applied	1.252	1.227	1.385	0.420
3	0.000714	0.9	32	Applied	1.218	1.122	1.372	0.469
4	0.000714	0.9	32	Applied	1.180	1.020	1.348	0.500
5	0.000714	0.9	32	Applied	1.158	0.9911	1.336	0.506

Table 4.1 object detection

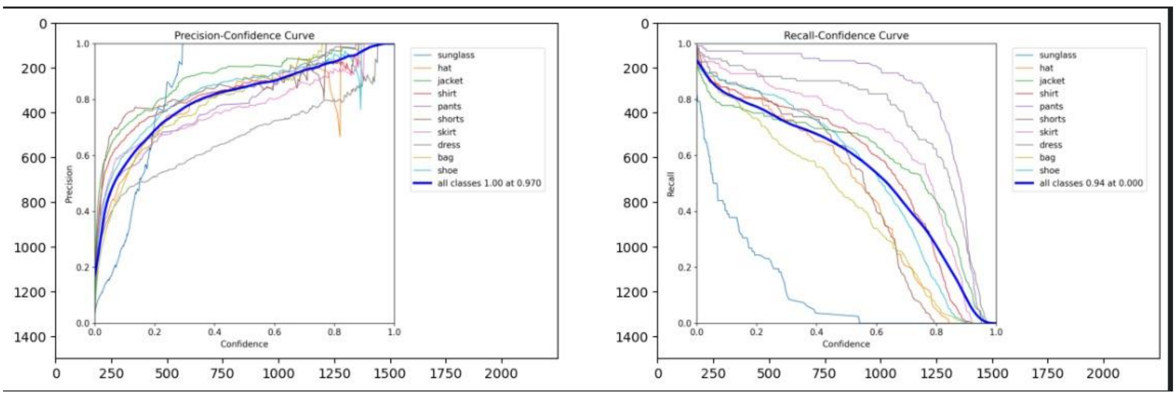

Fig 4.5 summarized Result for object detection

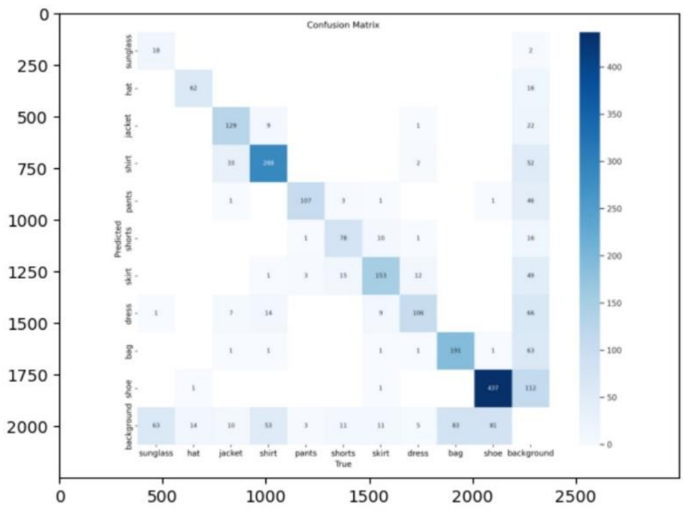

Fig 4.5 confusion matrix

4.2.2 VGG

Outfit coordination application that processes images and extracts features of clothing items

Feature Extraction: Use the VGG model to extract features from the clothing images. These features will be used to train a classifier that can identify different clothing items.

VGG16 Model Input and Output

Input	Description
Image of size `(224, 224, 3)`	The VGG16 model takes an input image of size 224x224 pixels with 3 color channels (RGB).

Output	Description
Features from layer "fc2"	The output is a feature vector extracted from the layer named "fc2" of VGG16. This layer is a fully connected layer near the end of the network, typically with 4096 dimensions.

Fig 4.6 VGG16 Model Input and Output Summary

4.2.3 Color matching

In the provided code, color matching is achieved by extracting dominant colors from images using K-Means clustering. Each clothing category (like dress, jacket, pants) has its images processed to find the most prominent colors. These dominant colors are then compared across categories, enabling tasks like color-based recommendation systems or image segmentation based on color similarity. The process involves using K-Means to cluster image pixels, extracting dominant colors as RGB values, and utilizing these colors for various color-related applications in fashion and image processing contexts

K-Means Clustering Summary

Input	Description
Data points `` `(n_samples, n_features)` ``	Input data points where each row represents a data point and each column represents a feature.

Parameters	Description
Number of clusters `` `k` ``	Number of clusters (groups) to form.

Output	Description
Cluster centers	Coordinates (centroids) of the `` `k` `` clusters, each represented as a feature vector.

Fig 4.7 K-Means Clustering Summary

We are developing a clothes matching app that brings a unique twist to outfit coordination. Our app suggests outfits based on the styles of famous individuals and fashion influencers, ensuring that our users stay in tune with the latest fashion trends. This approach sets us apart from other similar applications, as we provide trend-forward recommendations inspired by popular fashion icons, keeping our users stylish and up-to-date with the current fashion scene.

Additionally, our app highlights the top three outfits, carefully curated to match your preferences and the latest trends. To add an element of surprise and exclusivity, each outfit undergoes a drop-down mechanism and is revealed to the user after a certain period, enhancing the excitement and anticipation of discovering new fashion ideas.

Chapter 5

Run the Application

Chapter 5. Run the Application

A graphical user interface (GUI) is a visual interface that enables users to interact with electronic devices or software applications. It utilizes graphical elements such as icons, buttons, and windows to represent and manipulate data. GUIs enhance user experience by providing an intuitive and user-friendly environment for navigation and interaction.

5.1 Sign Window

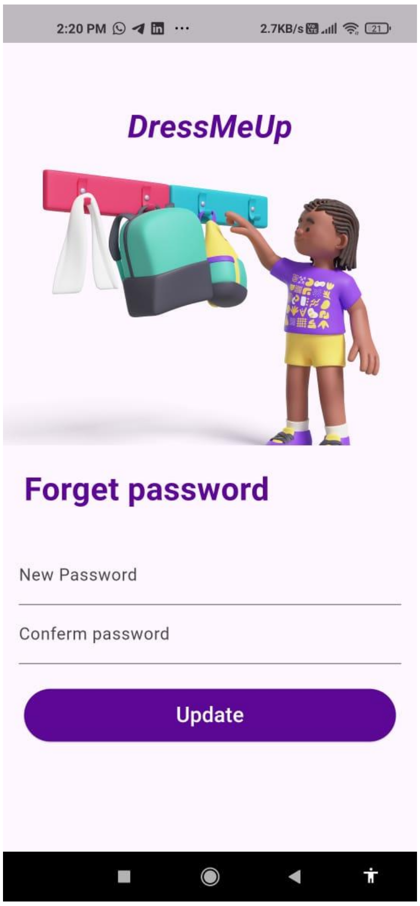

Fig 5.1 Sign Mode

The login page is the gateway to the application, allowing users to securely enter their credentials to access personalized features and content. By providing their username and password

5.2 Main Window

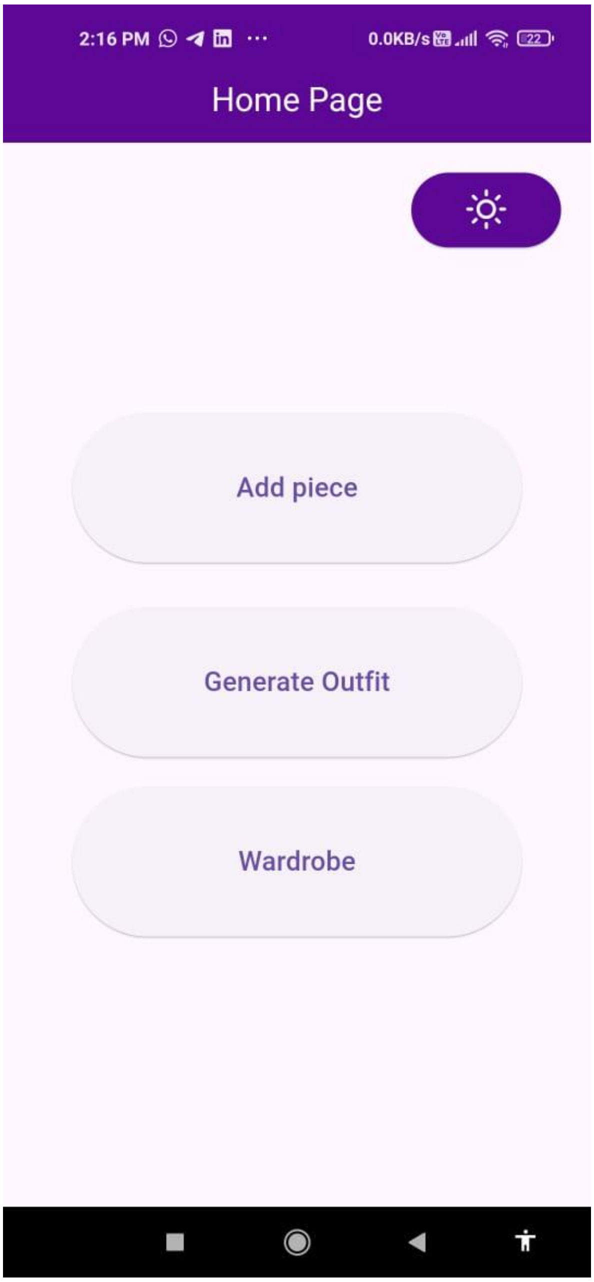

Fig 5.2 Main Window App

This is the main page in the application and there are the following choices. The first choice is that he takes pictures of his own clothes. The second choice is that it is the appropriate clothing for him. The third choice is wardrobe.

5.3 Add Piece

5.3.1 Camera mode

Fig 5.3 Camera Mode

Can store your clothes using the phone's camera

5.3.1 Upload Photo

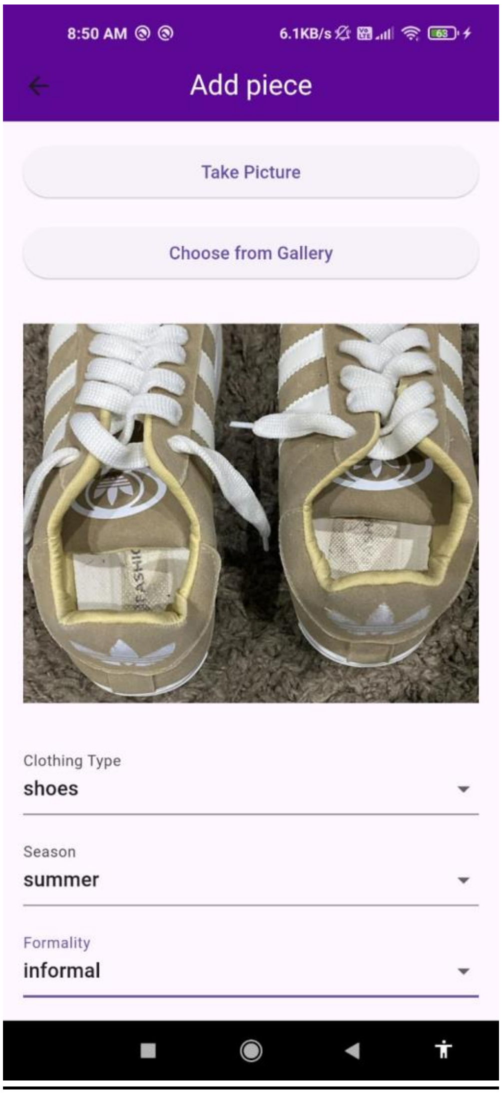

Fig 5.4 Upload Mode

Can choose a photo from your gallery and then upload it to the application

5.4 Generate Outfit

Fig 5.5 Generate Outfit

After selecting whether you prefer formal or informal clothing, we create the appropriate outfits tailored to your choice. We provide you with three meticulously curated outfits, each representing the best options available. These top three outfits have been chosen specifically for you, ensuring that you always look your best

5.5 Wardrobe

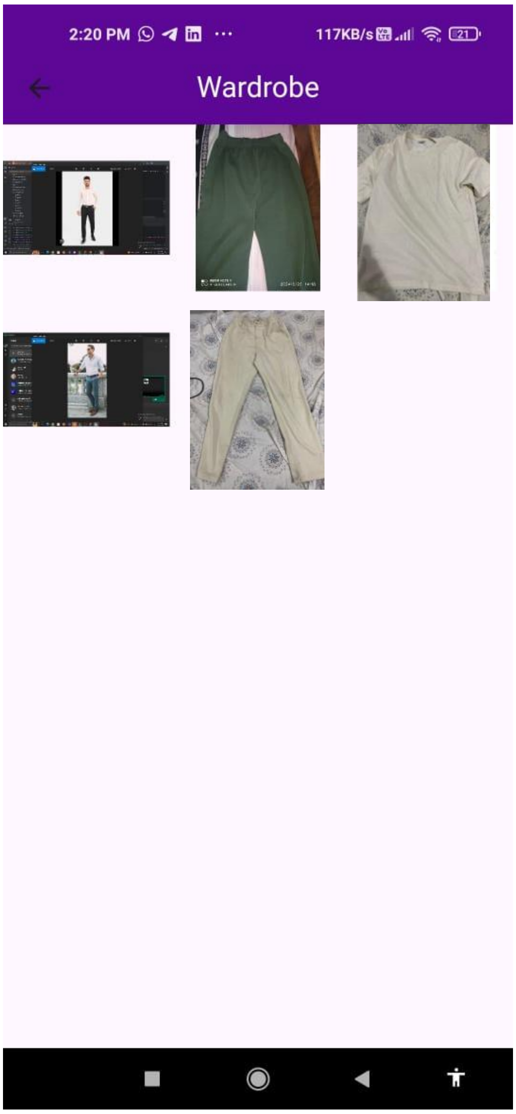

Fig 5.5 wardrobe Mode

Chapter 6

Conclusion and Future Work

Conclusion and Future Work

In this chapter, we will summarize the problem and present its impact on the community. We will also show how we solved this problem, the models used and the GUI to display to the user. We will also show in the future work what extra features can be added together.

6.1 Conclusions

In today's fast-paced world, selecting the right outfit can be a time-consuming and often frustrating task for many individuals. Our clothes matching app addresses this issue head-on by offering a seamless and efficient solution. By leveraging the latest fashion trends and styles from famous individuals and influencers, our app ensures that users receive recommendations that are not only stylish but also in line with the current fashion scene.

Our innovative approach significantly reduces the time and effort required to choose an outfit. Users can now enjoy a curated selection of the best three outfits, meticulously chosen to match their preferences and current trends. This feature not only saves valuable time but also guarantees that users always look their best with minimal effort.

Moreover, our app introduces an element of excitement and exclusivity with its drop-down mechanism. Outfits are revealed to the user after a specific period, adding an element of anticipation and making the experience more engaging.

Understanding the importance of dressing appropriately for the weather, our app also incorporates real-time weather data to suggest outfits that are suitable for the current climate. This functionality ensures that users are not only stylish but also comfortable, regardless of the weather conditions.

In conclusion, our clothes matching app revolutionizes the way individuals choose their outfits. By providing quick, stylish, and weather-appropriate recommendations, we save our users a considerable amount of time and effort. Our unique features, inspired by the latest fashion trends and tailored to individual preferences, set us apart from other similar applications. Users can now enjoy a hassle-free, stylish, and efficient way to stay fashionable and comfortable in any weather.

6.1 Future Work

Looking ahead, we are committed to continuously enhancing our clothes matching app to provide even greater value and convenience to our users. Our Future plans include several exciting features and improvements aimed at making the app more versatile, inclusive, and user-friendly.

One of our primary goals is to integrate advanced personalization features that leverage artificial intelligence and machine learning. These features will analyze user preferences, feedback, and behavioral patterns to provide even more accurate and tailored outfit recommendations. By continuously learning from user interactions, our app will evolve to meet the unique style needs of each individual, ensuring a truly personalized fashion experience.

In addition to improving personalization, we plan to expand our weather integration capabilities. Our aim is to provide even more precise and localized weather-based outfit suggestions, helping users stay comfortable and stylish regardless of their geographical location or changing weather conditions.

Inclusivity is a core value for us, and we are dedicated to making our app accessible to everyone, including individuals with visual impairments. In the future, we will develop features specifically designed for blind and visually impaired users. This includes voice-guided navigation, audio descriptions of outfits, and seamless integration with screen readers. By incorporating these features, we aim to empower visually impaired individuals to effortlessly select fashionable and weather-appropriate outfits, enhancing their overall quality of life.

Furthermore, we plan to introduce a social component to the app, allowing users to share their outfits, receive feedback, and draw inspiration from a community of like-minded fashion enthusiasts. This feature will create a vibrant and supportive fashion community within the app, fostering creativity and collaboration among users.

In summary, our future work will focus on enhancing personalization, expanding weather integration, and improving accessibility for visually impaired users. We are excited about the potential these advancements hold and remain committed to making our app the ultimate solution for effortless, stylish, and inclusive outfit selection.

References

References

[1] Charu C. Aggarwal, Senthilkumar Krishnan, and Subramanian Raman. (2021). "Fashion Recommendation Systems: Models and Methods: A Review." *ResearchGate*.

[2] Allure Editors. (2017). "This Is How Much Time Women Spend Picking Out Clothes." *Allure*.

[3] Yadav, P. (2023). A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods. India: Expert Systems With Applications.

[4] Kotouza, M. T., Tsarouchis, S.-F., Kyprianidis, A.-C., Chrysopoulos, A. C., & Mitkas, P. A. (2020). Towards Fashion Recommendation: An AI System for Clothing Data Retrieval and Analysis. In *Artificial Intelligence Applications and Innovations* (AIAI 2020), pp. 433–444. IFIP Advances in Information and Communication Technology (IFIPAICT, volume 584)

[5] Huang, E. (2021, November 9). Building an AI-Powered Outfit Recommendation System With Dataiku.

[6] Jain, S. (n.d.). Fashion-Recommendation-System. GitHub repository.

[7] Liu, L., Zhang, H., Ji, Y., & Wu, Q. M. J. (2019). Toward AI fashion design: An Attribute-GAN model for clothing match. *Neurocomputing*, 345, 39-48.

[8] ssabrut. (2019). Fashion Similarity Model. GitHub Repository.

[9] Evgeniou, T., & Pontil, M. (2001). Support Vector Machines: Theory and Applications. In *Machine Learning and Its Applications, Advanced Lectures* (pp. 249-257). DOI: 10.1007/3-540-44673-7_12.

[10] Wilson, D.R., Martinez, T.R.: Reduction Techniques for Exemplar-Based Learning Algorithms. Machine learning 38(3), 257–286 (2000).

[11] Rokach, L., & Maimon, O. (2005). Decision Trees. In *The Data Mining and Knowledge Discovery Handbook* (pp. 165-192). DOI: 10.1007/0-387-25465-X_9.

[12] Ali, J., Khan, R., & Ahmad, N. (2012). Random Forests and Decision Trees. *Journal Name*, 9(5), Article Number. DOI: [DOI Number]. License: CC BY-NC-ND 4.0.

[13] Chauhan, R. (2018). Convolutional Neural Network (CNN) for Image Detection and Recognition. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 1-5). DOI: 10.1109/ICSCCC.2018.8703316.

[14] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv.*

[15] Li, Y., Chen, C. L. P., & Zhang, T. (2022). A Survey on Siamese Network: Methodologies, Applications and Opportunities. *IEEE Transactions on Artificial Intelligence.*

[16] Lavanya, G., Pande, S. D., & Pimpri Chinchwad University. (2023). Enhancing Real-time Object Detection with YOLO Algorithm. *EAI Endorsed Transactions on Internet of Things.*

[17] Nguyen, T. H.-Y., & Hoang, T. T. (2018). A Low-Cost, Light-Weight, High-Efficiency Fabry-Perot Antenna. In *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*

[18] Hara, K., Jagadeesh, V., & Piramuthu, R. (2016). Fashion apparel detection: The role of deep convolutional neural network and pose-dependent priors. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV).*

[19] Daniel Gatis. (2023). rembg. GitHub Repository

[20] "A Novel Visual Cryptographic Method for Color Images" by Amarjot Singh, Devinder Kumar, and Omkar S N published in May 2013Kumar, and Omkar S N published in May 2013