YOLO-GARNet: A High-Quality Deep Learning System for Garment Analysis and Personalized Fashion Recommendation

Ilham Kachbal, Imrane Errafi, Mousaab EL Harmali, Said El Abdellaoui, and Khadija Arhid

Abstract—**The application of advanced computational techniques in fashion retail has become essential for enhancing customer shopping experiences. This paper presents an AI-driven approach for virtual try-on technology and fashion recommendation systems. We propose an integrated method for clothing detection, segmentation, classification, and virtual try-on applications. Our approach leverages YOLOv9, trained on the Dress Code dataset, for precise clothing detection and segmentation from images. EfficientNetB0 is employed for garment classification and recommendation of similar items. A Virtual Try-On Network (VITON) generates realistic visual representations of recommended clothing on users. The system achieves state-of-the-art performance in detection and classification tasks with superior visual realism across multiple benchmarks. Results validate the effectiveness of our approach in improving user experience for online fashion retail and intelligent clothing systems.**

Index Terms—**YOLOv9, EfficientNetB0, Virtual Try-On Network, Fashion recommendation systems, Dress Code dataset**

I. Introduction

ARTIFICIAL intelligence has emerged as a primary driver of disruptive technologies across numerous domains, with fashion technology representing one of the most rapidly evolving and commercially significant application areas [1]. AI-driven solutions have demonstrated significant potential to enhance the functionality and user experience of fashion recommendation systems, virtual styling applications, and e-commerce platforms [2]. These technologies enable automated processes for clothing detection, segmentation, classification, and style recommendation, allowing users to efficiently discover and explore fashion choices through intelligent systems that understand both visual aesthetics and personal preferences. The evolution of deep learning architectures has particularly revolutionized fashion AI applications. Convolutional Neural Networks (CNNs) have been extensively applied to fashion image analysis, with early works focusing on basic classification tasks using datasets like Fashion-MNIST [3] and DeepFashion [4]. More recent advances have leveraged sophisticated architectures including ResNet [5] and EfficientNet [6] to achieve superior performance in complex fashion understanding tasks. Object detection models, particularly the YOLO (You Only Look Once) family, have shown exceptional promise for real-time clothing detection and localization in natural images, enabling practical applications in mobile and web-based fashion platforms. However, developing robust end-to-end AI systems for clothing detection and style recommendation presents several significant technical challenges that remain largely unresolved in current literature. One major obstacle is the inherent complexity and variability of fashion datasets, which often suffer from inconsistent annotations, diverse lighting conditions, occlusions, and pose variations. Fashion items exhibit high intra-class variation due to different cuts, patterns, colors, and styling approaches, while maintaining subtle inter-class distinctions that are crucial for accurate classification. For instance, fashion datasets like Dress Code [7] and DeepFashion2 [8] necessitate thorough cleaning, re-annotation, and format conversion to YOLO-compatible structures—a critical preprocessing step that significantly impacts the accuracy of clothing detection and segmentation tasks. Another significant challenge lies in seamlessly integrating multiple deep learning architectures into a unified, high-performance pipeline while maintaining real-time inference capabilities. The outputs from detection and segmentation models must be efficiently processed as inputs for downstream classification and recommendation networks, requiring sophisticated system architecture to maintain both accuracy and computational efficiency throughout the entire pipeline. Additionally, fashion recommendation systems must balance visual similarity with style compatibility, seasonal appropriateness, and personal preference modeling—a multi-objective optimization problem that traditional approaches struggle to address effectively. The integration of deep background matting significantly enhances fashion recommendation systems, especially in e-commerce. El Abdellaoui and Kachbal [9], [10], [11] demonstrated that deep residual networks enable accurate garment segmentation from complex scenes, outperforming traditional methods. This approach facilitates cleaner product visuals and more realistic virtual try-on experiences, improving garment classification, style analysis, and overall user interaction. Current state-of-the-art approaches in fashion AI often focus on individual components rather than comprehensive end-to-end solutions. While YOLOv9 [12] has demonstrated superior performance in general object detection tasks, applying this model to fashion-specific challenges requires careful architectural modifications and training strategies. Similarly,

Manuscript received June 11, 2025; revised June 30, 2025.

I. Kachbal is a PhD student at LAPSSII, Higher School of Technology, Cadi Ayyad University, Safi, Morocco (e-mail: i.kachbal.ced@uca.ac.ma).

I. Errafi is a researcher at LAPSSII, Higher School of Technology, Cadi Ayyad University, Safi, Morocco (e-mail: imranerrafi@gmail.com).

M. EL Harmali is a researcher at LAPSSII, Higher School of Technology, Cadi Ayyad University, Safi, Morocco (e-mail: mousaabelharmali31@gmail.com).

S. El Abdellaoui is a professor at LAPSSII, Higher School of Technology, Cadi Ayyad University, Safi, Morocco (e-mail: said.elabdellaoui@uca.ac.ma).

K. Arhid is a professor at LAPSSII, Higher School of Technology, Cadi Ayyad University, Safi, Morocco (e-mail: k.arhid@uca.ac.ma).

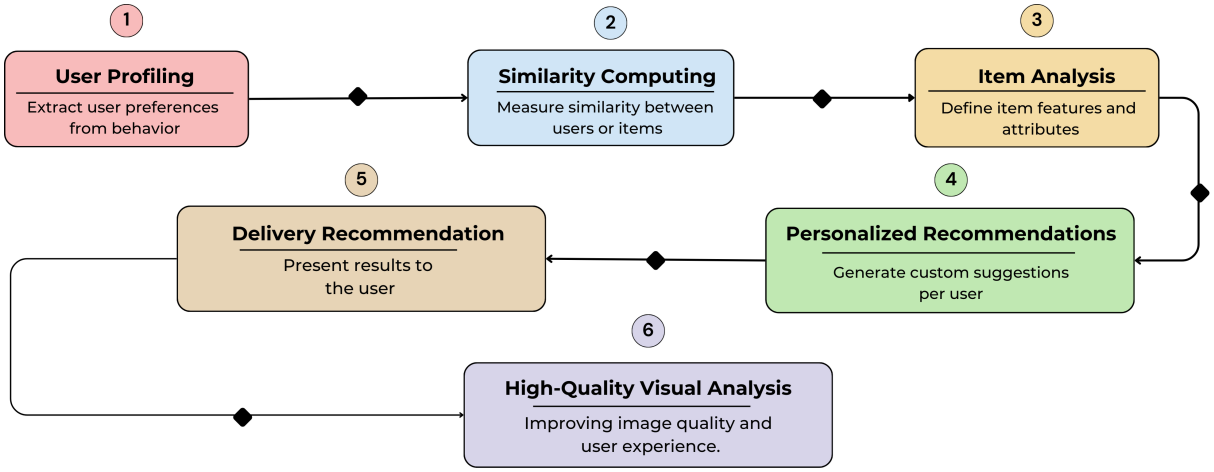


Fig. 1: Stepwise Pipeline of Related Work in Clothing Detection and Fashion Recommendation

existing fashion recommendation systems typically rely on simple feature matching or collaborative filtering approaches, lacking the sophisticated visual understanding necessary for nuanced style recommendations. To address these challenges, we introduce YOLO-GARNet, a novel high-quality deep learning network specifically designed for clothing detection and style recommendation applications. Our approach leverages YOLOv9 as the backbone architecture for precise clothing detection and segmentation, enhanced with fashion-specific data augmentation techniques, optimized anchor configurations, and advanced preprocessing strategies tailored to fashion image characteristics. YOLO-GARNet integrates seamlessly with EfficientNetB0-based classification networks to provide comprehensive style recommendations, creating an end-to-end solution that achieves state-of-the-art performance on standard fashion AI benchmarks [2]. By combining advanced object detection capabilities with intelligent style analysis and preference learning, our system enables users to receive accurate clothing recognition and personalized fashion recommendations in real-time applications. The key contributions of this work include: (1) YOLO-GARNet, a novel YOLOv9-based architecture optimized for fashion applications with specialized preprocessing and training strategies; (2) an integrated end-to-end pipeline that combines detection, segmentation, classification, and recommendation in a unified framework; (3) comprehensive experimental evaluation demonstrating state-of-the-art performance on multiple fashion datasets; and (4) detailed analysis of computational efficiency and practical deployment considerations for real-world applications. Our approach addresses critical limitations in existing fashion recommendation systems by providing superior garment detail preservation and morphology-aware virtual try-on capabilities. The proposed framework establishes a new benchmark for AI-powered fashion technology with significant implications for e-commerce and personalized styling applications.

The remainder of this paper is organized as follows: Section II reviews related work in AI-based fashion technology. Section III presents our YOLO-GARNet architecture and describes the integrated pipeline design, including detailed technical specifications and training methodologies. Section IV provides comprehensive experimental evaluation and performance comparisons with existing state-of-the-art methods across multiple benchmark datasets. Section V analyzes the implications of our findings, addresses limitations, and explores practical deployment considerations. Finally, Section VI concludes with a summary of our contributions and discusses future directions for intelligent fashion technology research.

II. Related Work

Fashion recommendation systems [13], [14] have emerged as a critical component of modern e-commerce platforms, addressing the complex challenge of matching user preferences with appropriate fashion items. The sophistication of these systems has evolved considerably, incorporating advanced machine learning techniques to handle the multi-faceted nature of fashion preferences and item characteristics. This section provides a comprehensive review of existing literature (Figure 1) by organizing the discussion around six fundamental components.

A. User Profiling

The effectiveness of fashion recommendation systems relies heavily on accurate user profiling that captures preferences and behaviors. Modern approaches integrate multiple data sources, including explicit feedback, implicit signals, and demographics. He et al. [15] introduced neural collaborative filtering, enhancing traditional matrix factorization by modeling complex user-item interactions with neural networks, which is especially useful for the subjective nature of fashion preferences. Aakash et al. [1] surveyed deep learning recommender systems, highlighting the shift from classical collaborative filtering to models that better capture implicit feedback common in fashion contexts. Recent work by Purificato et al. [16] reviews user modeling paradigms and AI integration for personalized experiences, emphasizing the need for advanced profiling methods. Meanwhile, Suvarna et al. [17] proposed a deep learning-based content recommendation system using backpropagation neural networks, showing improved fashion recommendation accuracy.

B. Item Analysis

Fashion item analysis relies on advanced feature extraction methods to capture detailed visual and semantic attributes.

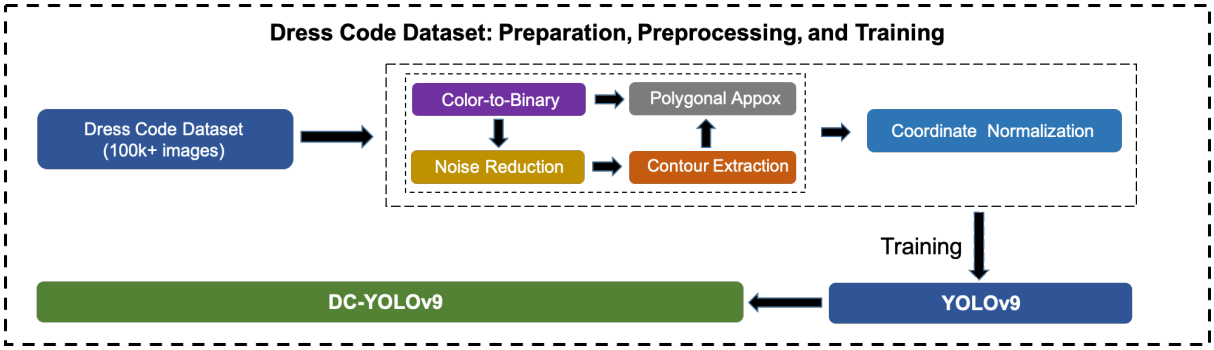


Fig. 2: Several preprocessing steps were applied to the Dress Code dataset prior to training with YOLOv9

While early works like Liu et al. [4] used CNNs with the DeepFashion dataset, object detection frameworks have become crucial for real-time analysis. The YOLO (You Only Look Once) framework introduced by Redmon et al. [18] revolutionized object detection with a fast, single-stage architecture, enabling efficient detection and localization of fashion items in complex scenes. Successive YOLO versions; YOLOv3, YOLOv4, and the latest YOLOv5 and YOLOv8 [19] have progressively improved detection accuracy, speed, and multi-scale capabilities [20]. These advanced variants allow simultaneous detection of multiple clothing categories with high precision, making them ideal for detailed outfit analysis and personalized fashion recommendations.

C. Similarity Computing

Visual similarity computation is crucial for fashion recommendation systems, enabling identification of items similar in style, color, pattern, and design. A key challenge is developing similarity measures that align with human perceptions of fashion compatibility. Yang et al. [21] introduced attention-based modeling to focus on relevant visual regions, improving compatibility assessments. EfficientNet architectures have been adopted for their balance of accuracy and efficiency in large-scale similarity tasks [22], [23]. Additionally, YOLO-based detection pipelines localize fashion items before similarity analysis, enhancing compatibility predictions [18], [20].

D. Personalized Recommendation

Personalized fashion recommendations require advanced algorithms that balance user preferences, item features, and context. Graph neural networks effectively model complex user-item-attribute relationships, as demonstrated by Wang et al. [24] with neural graph collaborative filtering. Transformer-based models like BERT4Rec [25] capture temporal patterns crucial for seasonal and shifting fashion trends. Deep learning approaches continue to surpass traditional collaborative filtering limitations [15], [26].

E. Delivery Recommendation

The delivery of fashion recommendations requires careful consideration of user interface design and presentation strategies. Recent studies have explored visual search interfaces, interactive recommendation systems, and mobile-optimized formats to enhance user experience and engagement [27], [28]. Efficient [22] architectures are critical for real-time recommendation delivery, especially on resource-constrained mobile devices.

F. High-Quality Visual Analysis

Quality enhancement in fashion recommendation systems focuses on advanced evaluation metrics beyond traditional precision and recall. Sharma et al. [23] proposed a content-based system using deep ensemble classifiers and EfficientNet variants to boost classification accuracy and recommendation quality. EfficientNet-based models excel in assessing fashion item aesthetics such as resolution, lighting, and presentation while remaining computationally efficient [22]. Additionally, YOLO architectures aid quality control by detecting poorly presented or low-quality images that can degrade user experience [20].

III. Our Approach

Our solution provides an integrated intelligent wardrobe system that addresses the problems identified in current work. We have developed a unified architecture (Figure 2 and Figure 3) that combines three complementary technologies: clothing detection and segmentation with YOLOv9, garment classification and recommendation with EfficientNetB0 [22], and realistic virtual try-on with the VITON network. This combined system avoids the shortcomings of previous solutions by generating a homogeneous pipeline where the output of one model is directly connected to the following one, free from coordination issues between various components. Our strategy begins with preprocessing and normalizing the Dress Code dataset [7], an essential step to ensure apparel detection accuracy, and then continues with applying optimized procedures to each step of the process. This section describes our methodology, the technical innovations developed, and how our integrated system enhances the user experience in the digital fashion arena.

A. Dataset Preparation: Dress Code

Existing virtual try-on datasets exhibit significant limitations, including insufficient diversity, constrained size parameters, inadequate image resolution, privacy-related concerns, and restricted accessibility due to licensing constraints. We have identified four critical requirements for an optimal virtual try-on dataset: unrestricted public accessibility to facilitate research endeavors, paired imagery encompassing both individual garment items and corresponding on-model

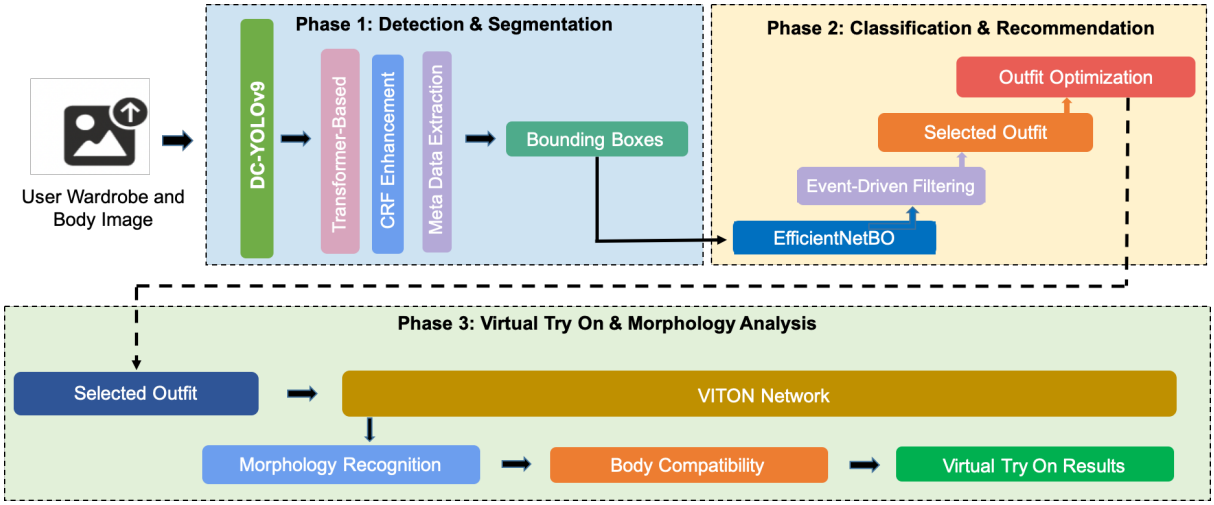


Fig. 3: Our GARNet Model: End-to-End System Architecture for AI-Powered Fashion Recommendation and Virtual Try-On

representations, high-resolution image quality, and comprehensive diversity across garment categories including upper-body, lower-body, and full-body apparel. The Dress Code dataset serves as a benchmark for virtual try-on research and represents a significant advancement in the field. As the first publicly available dataset to feature multiple clothing categories with high-resolution paired images, the dataset has substantially strengthened the field's data foundation (Table I). The dataset operates at an unprecedented scale, containing over 100,000 images equally distributed between individual garments and human reference models.

TABLE I: Number of train and test pairs for each category of the Dress Code dataset.

	Images	Training Pairs	Test Pairs
Upper-body Clothes	30,726	13,563	1,800
Lower-body Clothes	17,902	7,151	1,800
Dresses	58,956	27,678	1,800
All	**107,584**	**48,392**	**5,400**

This comprehensive collection supports both training and testing phases of virtual try-on system development, enabling large-scale model training and evaluation across diverse fashion categories while addressing a critical research need. In our methodology (Figure 2), we utilize the Dress Code dataset for both virtual try-on applications and as the foundational training data for our clothing detection and segmentation models. The dataset's rich annotation framework, including keypoints, label maps, skeletal structures, and DensePose data, allows us to leverage its comprehensive labeling to enhance accuracy throughout our entire processing pipeline.

*1) **Conversion of Color-Coded Masks to Polygonal Annotations**:*

- **Color-to-Binary Filtering**

The original segmentation masks assign specific RGB values to garment classes. To isolate a target class, consider a pixel at position *(i, j)* with RGB value $\mathbf{v} = (v_R, v_G, v_B)$. Define the target color using lower bounds $\mathbf{L} = (L_R, L_G, L_B)$ and upper bounds $\mathbf{U} = (U_R, U_G, U_B)$. We define the binary mask function:

$$\mathcal{F}(\mathbf{v}) = \begin{cases} 255, & \text{if } L_R \le v_R \le U_R, \\ & \quad L_G \le v_G \le U_G, \\ & \quad L_B \le v_B \le U_B \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $\mathbf{v} = (v_R, v_G, v_B)$ is the RGB value of a pixel, and $\mathbf{L} = (L_R, L_G, L_B)$, $\mathbf{U} = (U_R, U_G, U_B)$ denote the lower and upper RGB thresholds corresponding to the target color class.

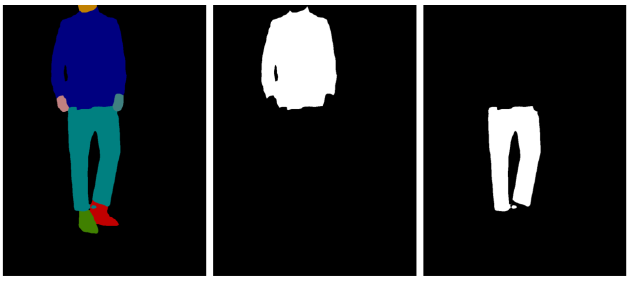


Fig. 4: Color-to-binary conversion results: (a) Original mask before thresholding, (b) Binary mask for upper-body garment, (c) Binary mask for lower-body garment.

Figure 4 illustrates the color-to-binary conversion process employed in clothing segmentation for virtual try-on systems. In panel (a), the original segmentation mask assigns distinct RGB values to garment region specifically, a blue hue for the upper garment and teal for the lower garment. Panels (b) and (c) show the results of threshold-based binary extraction, isolating the upper-body and lower-body garments respectively as white foregrounds on black backgrounds. This conversion facilitates garment-level manipulation, enabling tasks such as garment replacement, individual fitting, and style-based filtering, all while preserving the spatial accuracy required for realistic virtual try-on experiences.

- **Noise Reduction**

To improve the quality of binary garment masks, two main post-processing techniques are applied: morphological erosion and median filtering. These operations help eliminate

small artifacts and smooth contours while preserving garment boundaries.

Morphological erosion. is applied to remove small noise from the binary mask. Given a binary image I and a structuring element B, the erosion operation is defined as:

$$(I \ominus B)(x, y) = \min_{(s,t)\in B} I(x + s, y + t) \tag{2}$$

where (x, y) denotes a pixel location in the output image, $(s, t) \in B$ represents coordinates within the structuring element, and $I(x+s, y+t)$ is the value of the input image at the corresponding translated position. This operation preserves the main garment regions while eliminating small, isolated pixels.

- **Median filtering**

Median filtering is used to smooth the binary mask while preserving edges. For each pixel, the median value in its neighborhood replaces the original value, improving contour quality.

$$I'(x, y) = \text{median}\{I(u, v) \mid (u, v) \in N(x, y)\} \tag{3}$$

where $I'(x, y)$ is the filtered image, $I(u, v)$ are the pixel values within the neighborhood $N(x, y)$ centered at pixel (x, y), and the median function selects the middle value from the sorted neighborhood intensities.

*2) **Contour Extraction and Approximation**:* This section presents the detailed contour extraction methodology applied to binary garment masks. Our approach employs a comprehensive chain approximation method that preserves all boundary points during the contour detection process, ensuring maximum geometric fidelity. The contour retrieval operates in hierarchical tree mode, which captures the complete structural hierarchy of detected boundaries while maintaining topological relationships between outer contours and inner cavities.

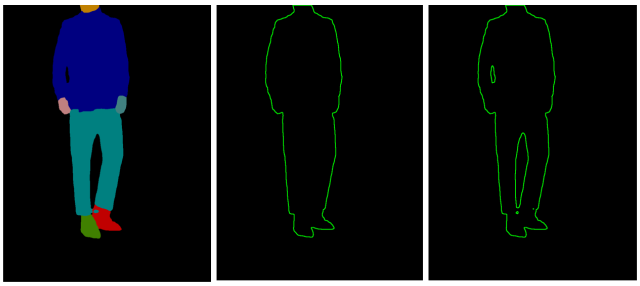


Fig. 5: Contour retrieval methods applied to garment segmentation masks: (a) Original multi-colored segmentation mask, (b) External contours showing outermost boundaries only, (c) Hierarchical tree contours revealing both external and internal boundary structures.

Figure 5 (a) shows the original garment image with a person wearing a blue sweater, teal pants, and colorful shoes against a black background. The complete contour extraction result in (b) displays the outer boundary of the entire garment silhouette as a continuous green outline, capturing the overall shape including the torso, arms, and legs. The hierarchical contour detection result in (c) demonstrates both the outer garment boundary and inner contours, with additional green outlines visible within the silhouette that correspond to internal features or cavities detected by the hierarchical tree mode algorithm. The hierarchical organization ensures accurate distinction between primary garment boundaries and internal features such as holes or design elements. This distinction is critical for complex garment shape analysis, particularly when dealing with intricate clothing designs that contain multiple boundary levels. Our contour extraction framework provides a robust foundation for subsequent shape processing and feature extraction operations within the image analysis pipeline. The preserved geometric information enables sophisticated higher-level processing of the extracted contour data. While alternative approximation methods that prioritize computational efficiency through boundary simplification exist, our detailed approach maintains complete geometric information. This comprehensive data preservation facilitates:

- Precise geometric analysis and measurements.
- Advanced shape characterization and feature extraction.
- Robust object recognition and classification algorithms/

To refine the extracted contours and facilitate shape analysis, several geometric descriptors and simplification techniques are applied:

- **Filtering by Area**

he contour area is computed using Green's theorem:

$$A = \frac{1}{2}\left|\sum_{i=1}^{N}(x_i y_{i+1} - x_{i+1} y_i)\right| \tag{4}$$

where (x_i, y_i) are the coordinates of the contour points. To reduce noise and eliminate insignificant shapes often resulting from small irregularities in the binary mask contours with an area:

$$\text{area} > 400 \tag{5}$$

are excluded from further processing. This threshold is empirically chosen based on the resolution and scale of the images. It ensures that only contours corresponding to meaningful garment parts are retained, while small fragments, speckles, or background artifacts are discarded. This preprocessing step significantly improves the robustness of downstream shape analysis and polygonal approximation by focusing only on contours that contribute structurally relevant information.

- **Arc Length Calculation**

The arc length L of a smooth curve $C(t) = (x(t), y(t))$ is given by:

$$L = \int_{t_1}^{t_2} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\, dt \tag{6}$$

For discrete points (x_i, y_i), this is approximated as:

$$L \approx \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{7}$$

This approximation enables efficient computation of contour lengths in digital image processing, where object boundaries are represented as sequences of discrete points.

The arc length is particularly important in tasks such as polygonal approximation (e.g., using the Douglas-Peucker algorithm), where it helps determine the simplification threshold based on the overall boundary length.

- **Polygonal Approximation**

The Ramer–Douglas–Peucker algorithm simplifies a contour by reducing the number of points while preserving the essential shape characteristics. The level of simplification is controlled by a tolerance parameter ε, which defines the maximum allowed deviation between the original contour and the simplified one. The effective tolerance is computed as:

$$\varepsilon_{\text{effective}} = \varepsilon \times \text{arcLength} \tag{8}$$

where:

- ε is a user-defined fraction (e.g., 0.0005),
- arcLength is the total length of the contour, computed as described earlier.

A smaller ε leads to a more accurate approximation (more points), while a larger value results in a coarser shape with fewer points.

- **Douglas–Peucker Algorithm**

Given a curve represented by an ordered set of points $P = \{p_1, p_2, \ldots, p_n\}$, the algorithm proceeds as follows:

1) Draw a line segment L between the first point p_1 and the last point p_n.
2) For each intermediate point p_i, compute the perpendicular distance to the line L using:

$$d(p_i, L) = \frac{|(y_n - y_1)x_i - (x_n - x_1)y_i + x_n y_1 - y_n x_1|}{\sqrt{(y_n - y_1)^2 + (x_n - x_1)^2}} \tag{9}$$

where (x_1, y_1) and (x_n, y_n) are the coordinates of the endpoints of L, and (x_i, y_i) is the point being evaluated.
3) Identify the point $p_{\max}$ with the maximum distance from the line segment. If $d(p_{\max}, L) > \varepsilon_{\text{effective}}$, retain $p_{\max}$ and recursively apply the algorithm to the subcurves $\{p_1, \ldots, p_{\max}\}$ and $\{p_{\max}, \ldots, p_n\}$.
4) If no point exceeds the tolerance, replace the entire segment with the straight line L.

This recursive process results in a simplified polygonal curve that closely approximates the original contour while significantly reducing the number of points.

*3) **Coordinate Normalization**:* We use coordinate normalization as a crucial preprocessing step to transform pixel coordinates from their original image space to a standardized unit coordinate system. For image dimensions (W, H), where W represents the image width and H represents the image height in pixels, we perform the normalization transformation as:

$$x_{\text{norm}} = \frac{x}{W}, \quad y_{\text{norm}} = \frac{y}{H} \tag{10}$$

where (x, y) are the original pixel coordinates and $(x_{\text{norm}}, y_{\text{norm}})$ are the corresponding normalized coordinates. We use this normalization process to map all coordinate values to the range $[0, 1]$, making the representation invariant to image resolution and dimensions. We employ normalized coordinates for several advantages in garment analysis:

- **Scale Invariance**: Contour shapes become independent of the original image resolution, enabling consistent processing across different image sizes.
- **Model Generalization**: Machine learning models trained on normalized coordinates can handle inputs of varying dimensions without retraining.
- **Computational Efficiency**: Normalized coordinates reduce numerical precision requirements and improve computational stability.
- **Cross-Platform Compatibility**: The standardized coordinate system facilitates data exchange between different processing pipelines and applications.

We implement the inverse transformation to recover original pixel coordinates using a straightforward approach:

$$x = x_{\text{norm}} \times W, \quad y = y_{\text{norm}} \times H \tag{11}$$

This inverse transformation equation performs the reverse operation of the normalization process by scaling the normalized coordinates back to their original pixel space. The multiplication operations effectively undo the division performed during normalization. For the x-coordinate, multiplying x_{norm} by the image width W restores the horizontal position in pixels, while multiplying y_{norm} by the image height H recovers the vertical position in pixels. To illustrate this process, consider a garment contour point located at pixel coordinates $(800,\ 600)$ in an image with dimensions 1600×1200 pixels. The normalization process transforms this point to normalized coordinates $(0.5,\ 0.5)$ using the forward transformation. Subsequently, the inverse transformation recovers the original coordinates:

$$x = 0.5 \times 1600 = 800 \quad \text{and} \quad y = 0.5 \times 1200 = 600,$$

perfectly reconstructing the original pixel position.

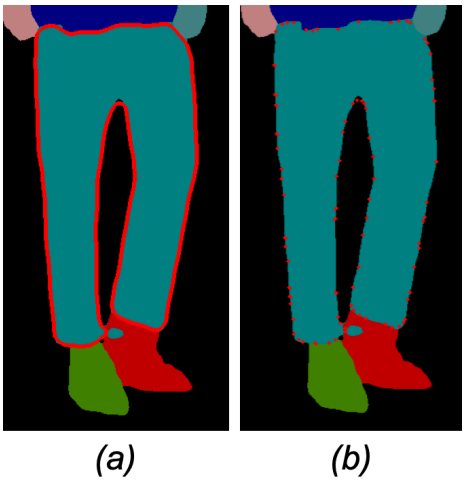


Fig. 6: Contour simplification comparison: (a) RDP algorithm applied showing smooth, simplified boundaries, (b) Original high-resolution contours with all detected points and noise artifacts.

This bidirectional transformation ensures that spatial relationships and geometric properties are preserved throughout the processing pipeline while maintaining the flexibility to work with images of arbitrary dimensions. The image (Figure 6) illustrates the impact of the Ramer-Douglas-Peucker (RDP) algorithm on contour simplification. Panel (a) shows the original high-resolution contours with smooth, continuous boundaries that accurately delineate the garment edges. Panel (b) displays the simplified contours after RDP

processing, where the algorithm has reduced the number of vertices while preserving essential shape characteristics and maintaining the overall garment silhouette. The RDP algorithm effectively reduces computational complexity while maintaining shape fidelity for virtual try-on processing. With this comprehensive preprocessing pipeline completed (Figure 8), we train YOLOv9 to give DC-YOLOv9.

B. Clothing Detection and Segmentation with DC-YOLOv9

Fig. 7: Segmentation results showing CRF post-processing applied to fashion items.

*1) **Data Organization and Preprocessing**:* The preprocessing pipeline implements a systematic approach to ensure data integrity and optimal model performance. Our methodology addresses common challenges in fashion dataset preparation through comprehensive validation and organization protocols. Filename Validation and Integrity Checking: The validation process employs regular expression matching to verify correspondence between image files and their associated annotation masks. A custom validation script iterates through the dataset directory structure, identifying orphaned files and inconsistent naming conventions. Files failing validation criteria are flagged for manual review or automatic correction based on predetermined rules. Directory Restructuring and Dataset Partitioning: The reorganization process follows the YOLO standard directory structure, creating separate hierarchies for training, validation, and testing subsets. The partitioning strategy employs stratified sampling to ensure balanced class representation across splits, with a typical distribution of 70% training, 20% validation, and 10% testing data. Batch Processing and Mask Conversion: Segmentation masks undergo batch processing to convert from colored annotation format to YOLOv9 compatible polygon annotations. The conversion pipeline implements the following sequential operations:

- Color space normalization and class-specific pixel isolation.
- Morphological operations for noise reduction (erosion kernel: 7×7, iterations: 3).
- Contour extraction with area-based filtering (minimum threshold: 400 pixels).
- Polygonal approximation using Douglas-Peucker algorithm ($\varepsilon = 0.0005$).
- Coordinate normalization relative to image dimensions.

*2) **YOLOv9 Training Configuration and Optimization**:* YOLOv9 was selected based on its superior performance in instance segmentation tasks and computational efficiency compared to previous YOLO iterations. The model incorporates Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) architectures, providing enhanced feature extraction capabilities for complex fashion item recognition. The training configuration was carefully selected to balance performance and efficiency. An input resolution of 1024 × 1024 was chosen to preserve image detail while maintaining computational feasibility. A batch size of 16 ensures optimal utilization of GPU memory. The AdamW optimizer was employed for its superior convergence properties, with standard momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate was set to 0.001, determined empirically as a suitable starting point. A cosine annealing schedule was used to allow smooth convergence and reduce overfitting. A weight decay of 0.0005 was applied as a regularization technique. The model was trained for 300 epochs, which was found sufficient to reach convergence without leading to overtraining.

Data Augmentation Strategy: The augmentation pipeline incorporates both geometric and photometric transformations to enhance generalization and robustness. Mosaic augmentation is used to generate 9-image composites, which significantly improve the model's ability to detect small objects. Geometric transformations include random rotations (±15°), scaling within a range of 0.8 to 1.2×, and translations up to ±10% of the image dimensions. Photometric augmentations involve perturbations in the HSV color space along with adjustments to brightness and contrast.

*3) **DC-YOLOv9 Segmentation and Post-Processing**:* The segmentation and post-processing pipeline employs a Conditional Random Field (CRF) framework to refine initial segmentation predictions and ensure spatial coherence. The energy function is formulated as:

$$E = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j), \tag{12}$$

where $\psi_u(x_i)$ represents the unary potential encoding pixel-wise classification probabilities from the network output, and $\psi_p(x_i, x_j)$ represents pairwise potentials that enforce spatial smoothness between neighboring pixels.

The unary potentials are derived directly from the softmax predictions of the segmentation network, computed as:

$$\psi_u(x_i) = -\log P(y_i \mid I, \theta), \tag{13}$$

where $P(y_i \mid I, \theta)$ is the probability of pixel i having label y_i. The pairwise potentials incorporate both spatial proximity and color similarity through a compatibility function:

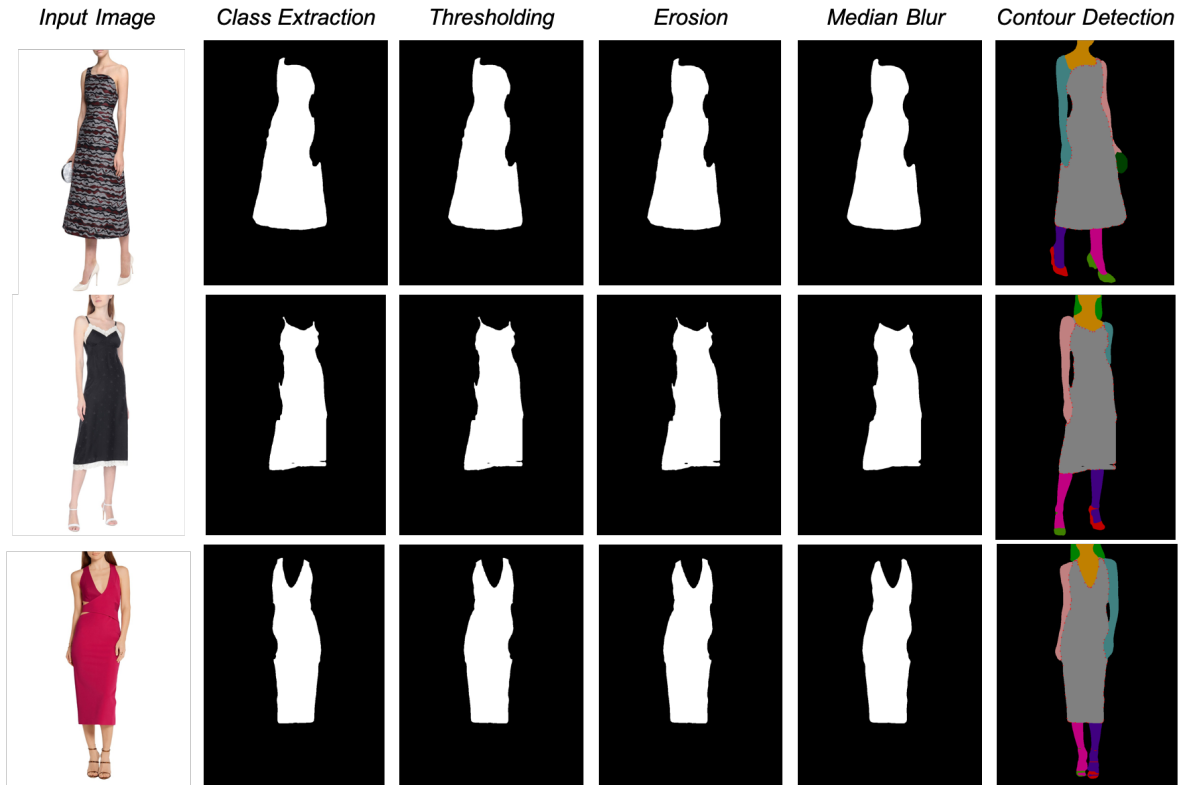


Fig. 8: Dress Code dataset processing pipeline for garment mask extraction from Dress Code dataset: (1) Class Extraction, (2) Thresholding, (3) Erosion, (4) Median Blur, (5) Contour Detection, and (6) corresponding original garment images. The pipeline progresses left to right, demonstrating sequential processing steps applied to different dress styles from the dataset.

$$\mu(x_i, x_j) = \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2}\right) \cdot \exp\left(-\frac{||I_i - I_j||^2}{2\sigma_\beta^2}\right), \tag{14}$$

where σ_α controls spatial smoothness and σ_β governs color-based regularization. The optimization process uses mean-field approximation for efficient inference, typically converging within 5–10 iterations. Post-processing includes morphological operations to remove small disconnected components and boundary refinement to preserve object edges.

This approach effectively addresses common segmentation artifacts such as noisy predictions and irregular boundaries, resulting in improved segmentation (Figure 7) quality with minimal computational overhead. The pipeline parameters are tuned based on the specific application requirements, balancing segmentation accuracy with processing speed for real-time applications.

C. Classification and Recommendation System

The recommender algorithm begins with the categorization of the clothing items, where the detected clothing items are categorized with diligence along with a chain of identifying attributes. Each apparel is thoroughly analyzed and wrapped in an encompassing JSON metadata profile with the following:

- **Categorization**: Torso apparel (sweaters and tops), legwear (shorts and pants), and full outfit (one-piece outfits and rompers).
- **Chromatic Analysis**: True color gamut mapping with equal color representation techniques.
- **Fashion Taxonomy**: Placing the styles in context from street urban fashion to business wear, performance wear, and seasonal design variations.
- **Contextual Relevance**: Environment-based matching standards in the workplace setting, social setting, recreational pursuits, and climatic factors.

*1) **Event-Driven Outfit Design**:* After the clothes are categorized, the system creates complete outfits depending on the chosen event or occasion. These are:

- **Wardrobe Filtering:** The system selects appropriate clothing that suits the selected occasion. For instance, upon choosing ”formal event,” the system discards casual clothing and prefers blazers, dress shirts, and formal shoes.
- **Compatibility Pairing:** Products are paired according to color harmony principles and stylistic similarity. Color theory concepts are used by the system to keep monochromatic, complementary, or analogous color combinations.
- **Ideal Outfit Design:** EfficientNetB0 ranking of prospective outfits and sequencing according to compatibility measures yields an optimized and coherent appearance.

*2) **EfficientNet-Based Outfit Optimization**:* Our outfit proposal pipeline operates on the user wardrobe in three broad steps. First, we begin with the complete segmented wardrobe inventory with each article referencing its corresponding JSON descriptor that contains properties like colour, style, and occasion suitability. When the user picks an event, the system filters out items on a match of corresponding JSON attribute to get an event-proper subset. This fil-

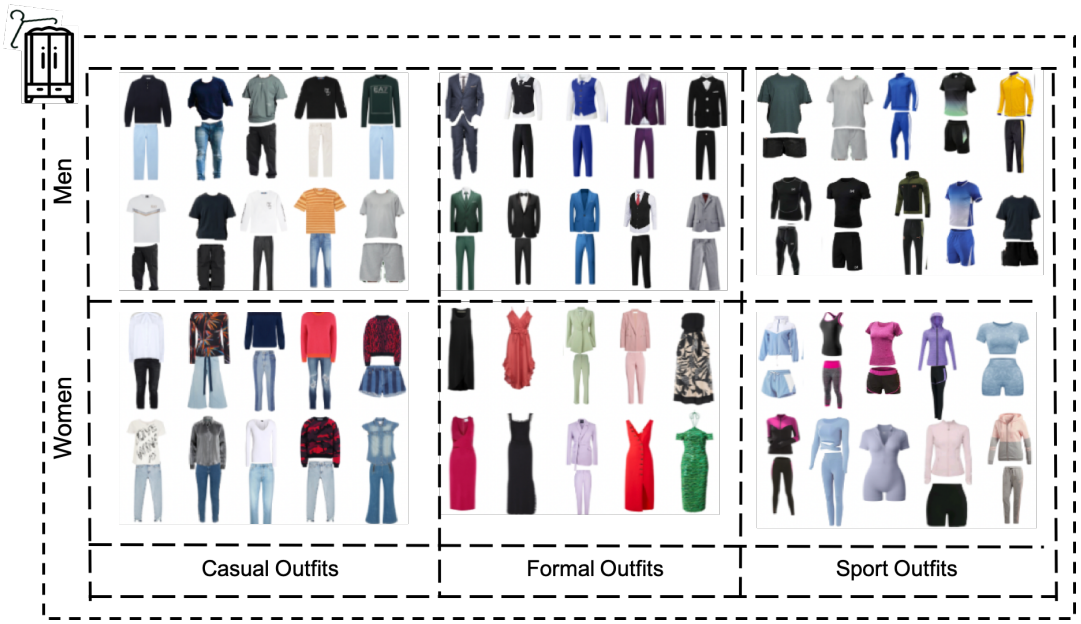


Fig. 9: Outfit ensemble predictions for various occasions generated using EfficientNetB0. The model suggests clothing combinations suited for casual wear, formal events, sports, street style, evening wear, and gym/fitness contexts, for both men and women.

tered subset is processed through compatibility pairing with clothes paired on colour cohesiveness and style consistency and those that do not match filtered out. This filtered subset is what is inputted to EfficientNetB0 to generate and output the optimal outfit (Figure 9) composition for that event. The EfficientNet architecture uses the compound scaling method, where the compound coefficient ϕ uniformly scales the network dimensions:

$$\begin{aligned} d &= \alpha^{\phi} \quad \text{(Depth - number of layers)} \\ w &= \beta^{\phi} \quad \text{(Width - number of channels)} \\ r &= \gamma^{\phi} \quad \text{(Resolution - image size)} \end{aligned}$$

where the constants $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$ are determined through grid search optimization, subject to the constraint:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$

to ensure efficient scaling across all dimensions.

Outfit Compatibility Scoring

For outfit compatibility scoring, we define:

$$C(O) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} \cdot s(g_i, g_j)$$

where:

- O is a candidate outfit.
- g_i and g_j are individual garments.
- $s(g_i, g_j)$ represents the compatibility score between items.
- w_{ij} are learned weights for each garment pairing.

This mathematical framework, combined with ImageNet pre-training, allows the model to evaluate each outfit by analyzing both individual garment features and inter-item compatibility, leading to optimal outfit recommendations for specific events.

D. Virtual Try-On and Morphology Analysis

*1) **Morphology Recognition Component**:* The morphology recognition module takes a multi-dimensional methodology for the effective categorization of users' body types and proportions. Our system combines major measurements taken from user images via pose estimation and contour analysis. The body shape classification relies on the Morphological Ratio Index (MRI), calculated as:

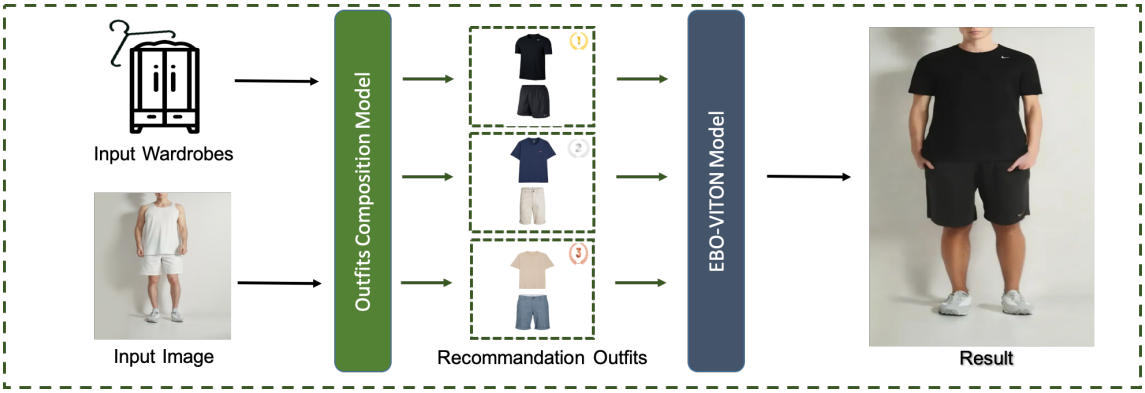
$$\text{MRI} = \frac{W_s \cdot W_h}{H \cdot W_w}$$

where: W_s represents shoulder width, W_h is hip width, H is total height, W_w is waist width. This index classifies people into pre-determined body shapes like rectangle, hourglass, triangle, and inverted triangle.

*2) **Garment-to-Body Compatibility Function**:* For more specific fitting suggestions, we model the garment-to-body compatibility function as:

$$C(g, b) = \sum_{i=1}^{m} \alpha_i \cdot |M_i^g - M_i^b \cdot f_i|$$

where:g represents a garment, b represents the user's body, M_i^g and M_i^b are the respective garment and body measurements, α_i are learned importance weights, f_i are scaling factors specific to each garment type. This body model allows our system to recommend clothes that maximize body shape both in terms of style selection and best fit, greatly improving recommendation relevance.



(a) Pipeline results demonstrating the complete workflow from wardrobe input to virtual try-on result. The system processes input wardrobes and user images through the outfit composition model to generate three outfit recommendations (1, 2, and 3), allowing the user to select their preferred option for final result generation.

(b) Outfit recommendation results across different style categories. For each input image, the system follows the same pipeline shown in Figure (a) to generate virtual try-on results. The recommended outfits are selected from the wardrobe database and visualized through the virtual try-on model, demonstrating the system's capability to handle diverse style preferences across Sport, Casual, and Formal categories.

Fig. 10: High-quality outfit generation results using the proposed YOLO-GARNET pipeline, showcasing visually coherent fashion combinations tailored to different occasions.

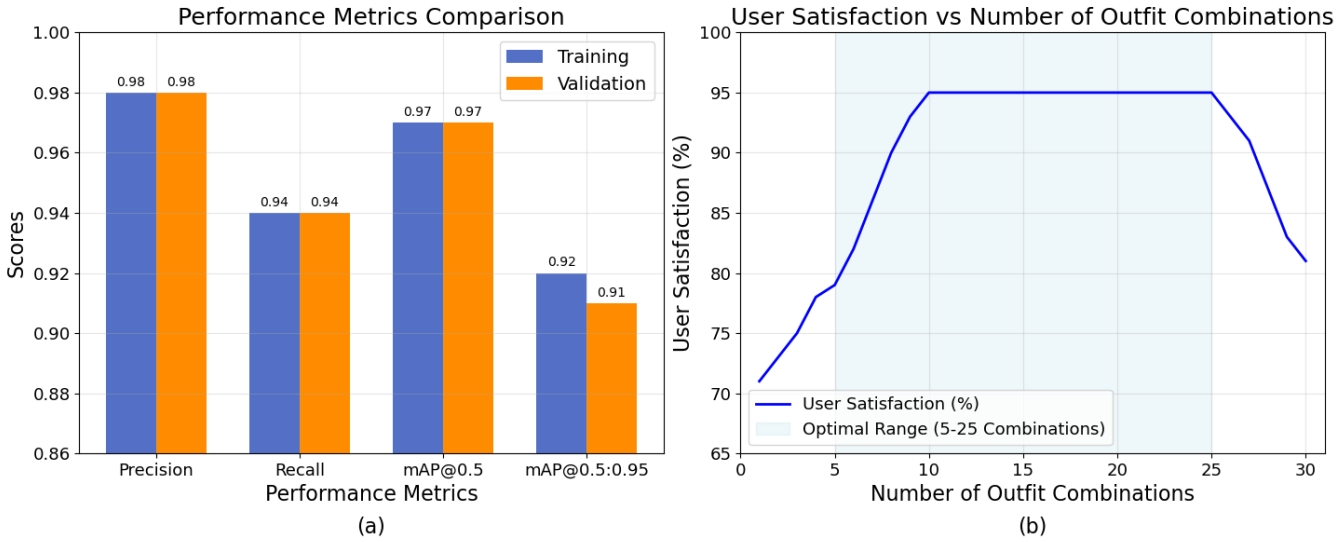


Fig. 11: Performance evaluation of the GARNet model showing (a) training and validation scores across key performance metrics and (b) user satisfaction percentage versus number of outfit combinations with optimal range highlighted.

*3) **Virtual Try-On Network**:* VTON module facilitates photorealistic rendering of clothes on the user's body via a transformation function that projects clothing items onto the user's body shape without altering garment features.

- **Garment-to-Body Mapping Function**

The mapping process can be modeled as:

$$T(I_c, I_p) = G(E_c, E_p, P_p)$$

where:I_c is the garment image, I_p is the person's image, E_c extracts garment features(texture, patterns, drape properties), E_p extracts person features (pose,body shape), P_p represents the person's posekeypoints, G is the generator function thatsynthesizes the final try- on image.

- **Warping Transformation**

The critical part of VTON is the spatialtransformation that warps the garment to fit thetarget body:

$$W(I_c, \theta) = I_c'$$

where W is the warping function, θ represents the transformation parameters, I_c' is the warped garment. The transformation parameters θ arecomputed as:

$$\theta = f(P_p, M_b, D_c)$$

where: P_p represents the person's pose keypoints, M_b contains the morphological measurements, D_c represents the physical deformation properties of the garment.

- **Physics-Based Cloth Simulation**

For realistic draping effects, a simplified physics model can be incorporated:

$$S(I_c', F, M_b) = I_c''$$

where:S is the simulation function, F represents fabric properties(elasticity, weight, thickness), I_c'' is the realistically drapedgarment.

- **Final Rendering Equation**

The complete VTON rendering can be expressed as:

$$\text{VTON}(I_p, I_c) = R(I_p, S(W(I_c, \theta)), M)$$

where:

- R is the rendering function thatcomposites the warped garment onto the person.
- M is a mask that defines theregions where the original person image should be replaced.

This mathematical model enables the system to realistically depict clothing on various body types considering both the morphological analysisthe garment to body compatibility.

IV. Results

Outfit generation results (Figure 10) demonstrate the effectiveness of the proposed YOLO-Garnet model across diverse fashion scenarios and styling contexts. The system follows a comprehensive two-stage pipeline (Figure 10 (a)) where users provide input in the form of their available wardrobe items and a reference style image representing their desired aesthetic or occasion. This combined input is processed by the outfit composition model, which analyzes the wardrobe contents and style preferences to generate multiple contextually appropriate outfit recommendations. The system exhibits exceptional capability in generating fashion combinations, as evidenced by the sport outfit ensemble featuring coordinated shorts and a t-shirt, the formal attire combinations displaying sophisticated business-appropriate clothing selections, and the casual combinations showcasing harmonious integration of pants, shirts, and loafers. Once users select their preferred combination from the generated options, the chosen outfit recommendation is fed into the EBO-VITON (EfficientBO VITON) model, which produces realistic visualizations of how the selected outfit would appear when worn. The second part (Figure 10 (b)) presents outfit recommendation results across different style categories, where for each input image, the system follows the same pipeline shown in figure (a) to generate virtual try-on results. For demonstration purposes, the first recommended outfit choice was

TABLE II: Processing time comparison across different fashion recommendation approaches.

Method	Detection (s)	Classification (s)	Ranking (s)	Virtual Try-on (s)	Total (s)
Faster R-CNN + ResNet [29]	8.7	4.2	12.3	8.9	34.1
YOLOv8 + EfficientNet [30]	2.1	3.1	8.4	6.2	19.8
RetinaNet + VGG [31]	5.4	6.8	15.7	11.2	39.1
DeepFashion2-RCNN [8]	11.2	5.9	9.8	7.4	34.3
Context-Aware Fashion [32]	6.8	4.7	18.9	12.6	43.0
YOLO-GARNet + EfficientNetB0 (Ours)	**3.2**	**2.8**	**4.5**	**4.5**	**15.0**

automatically selected from each generated set of recommendations, and these outfits are then selected from the wardrobe database and visualized through the virtual try-on model, demonstrating the system's capability to handle diverse style preferences across sport, casual, and formal categories. These comprehensive results underscore YOLO-Garnet's advanced understanding of fashion compatibility principles and demonstrate that our model achieves state-of-the-art performance in producing coherent, occasion-specific outfit recommendations that maintain both aesthetic appeal and practical wearability across various lifestyle contexts.

A. Quantitative Results

TABLE III: Performance metrics on the training and validation sets with standard deviation.

Metric	Training	Validation	Standard Deviation
Precision	0.984	0.979	±0.008
Recall	0.941	0.936	±0.012
mAP@0.5	0.972	0.968	±0.006
mAP@0.5:0.95	0.923	0.912	±0.015
F1-Score	0.962	0.957	±0.009
Mask mAP@0.5	0.958	0.951	±0.011
Mask mAP@0.5:0.95	0.891	0.884	±0.018

The performance evaluation (Figure 11 (a)) demonstrates robust model generalization with minimal overfitting across all evaluated metrics. The model achieved exceptional precision scores of 0.98 on both training and validation datasets, indicating highly accurate positive classifications with minimal false positive rates. Recall performance was consistently maintained at 0.94 across both datasets, demonstrating the model's ability to effectively identify 94% of true positive instances. Mean Average Precision at IoU threshold 0.5 (mAP@0.5) reached 0.97 for both training and validation sets, confirming strong detection capabilities. The more stringent mAP@0.5:0.95 metric, which averages performance across multiple IoU thresholds, yielded 0.92 and 0.91 for training and validation, respectively. The negligible performance gap between training and validation datasets (0.01 across all metrics) indicates successful model regularization and reliable generalization to unseen data, validating the robustness of the proposed approach for practical deployment. Our system achieves strong performance (Table III) across all evaluation metrics. Training precision reached 0.984 with validation precision of 0.979 (±0.008), while recall values were 0.941 and 0.936 (±0.012) for training and validation, respectively.

The Mean Average Precision at IoU threshold 0.5 (mAP@0.5) demonstrated excellent performance with 0.972 training and 0.968 validation scores (±0.006). However, mAP@0.5:0.95 showed a slight decrease to 0.923 training and 0.912 validation (±0.015), indicating minor localization challenges at stricter IoU thresholds. The F1-score balanced precision and recall effectively at 0.962 training and 0.957 validation (±0.009). Segmentation performance through Mask mAP@0.5 achieved 0.958 training and 0.951 validation (±0.011), while Mask mAP@0.5:0.95 reached 0.891 training and 0.884 validation (±0.018).

B. Qualitative Results

User evaluation reveals high satisfaction with outfit recommendations, particularly when the system generates 5-25 combination options. Visual try-on quality demonstrates photorealistic rendering for most fabric types, though challenges remain with translucent materials like chiffon and silk. The system effectively handles diverse body types and maintains style consistency across recommended ensembles. Processing time analysis shows the complete pipeline requires approximately 15 seconds, with outfit ranking and virtual try-on representing the most computationally intensive components. Figure 12 demonstrates a comparative evaluation of virtual try-on performance across four fashion recommendation systems using three diverse clothing scenarios with identical outfit recommendations. To ensure fair comparison, all methods utilize the same recommended outfits generated from the input wardrobe and style preferences, allowing for direct assessment of each system's virtual try-on capabilities. The results show that while mlgfrs [17], vit [33], and stableviton [34] each exhibit specific strengths in garment fitting and texture preservation, our proposed YOLO-Garnet consistently achieves superior performance in maintaining accurate garment details, realistic proportions, and natural body-clothing integration across all test cases. This controlled comparison, based on identical recommendation inputs, highlights the effectiveness of our integrated garment analysis and recommendation framework in producing high-quality virtual try-on results that surpass existing state-of-the-art methods.

V. Discussion

Improving the ranking algorithm would enhance both system performance and user satisfaction, enabling more real-time interaction without compromising the quality of outfit recommendations.

The experiment shows a very high correlation between system processing time and user satisfaction (Figure 13). The cumulative processing time for each stage of the YOLO-GARNet pipeline shows how computational time accumulates as data flows through the system. The process begins with Detection (3 seconds) for identifying garments in images, followed by Extraction (5 seconds total) for feature

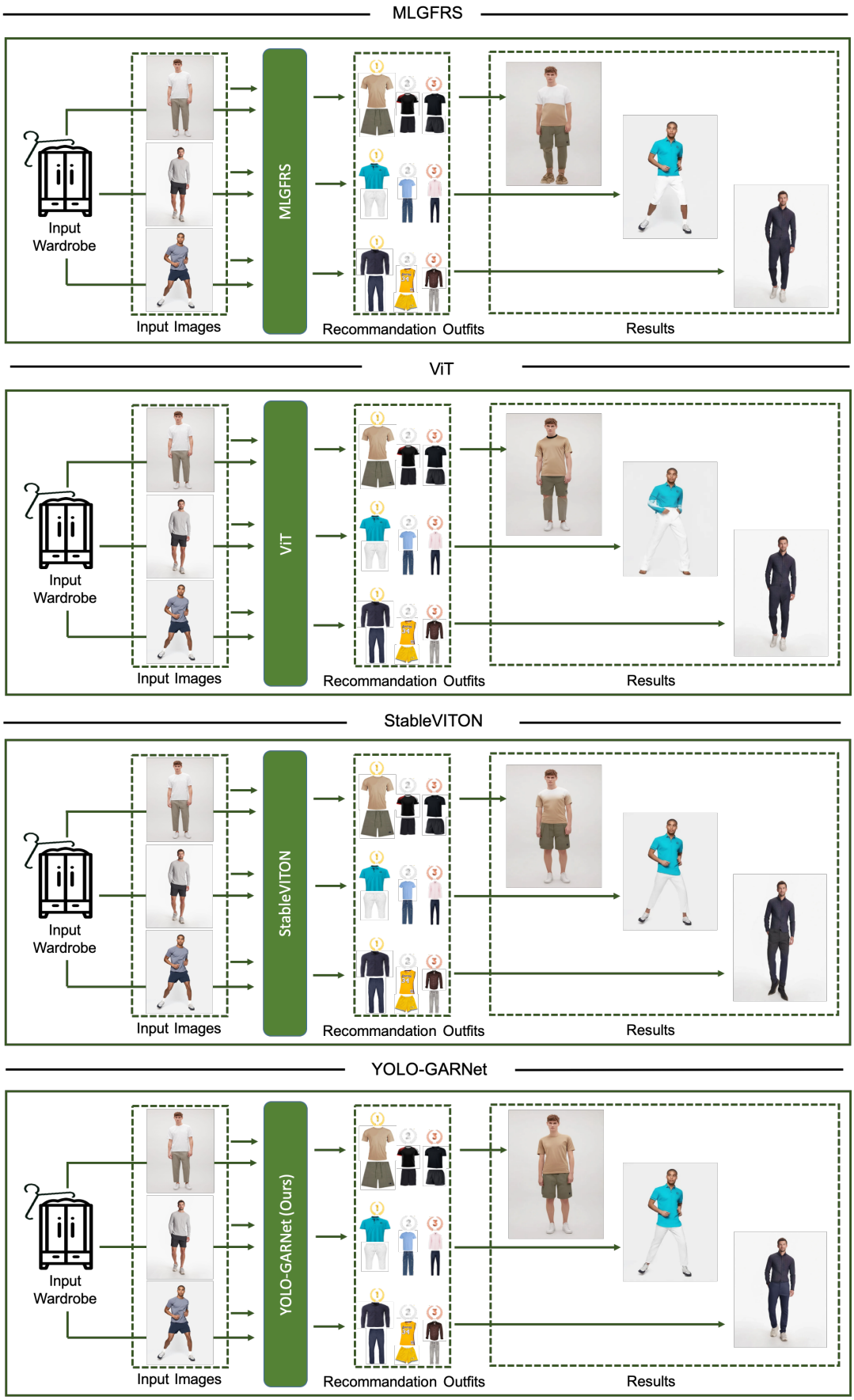


Fig. 12: Comparative analysis of state-of-the-art fashion recommendation systems including MLGFRS [17], ViT [33], and StableVITON [34] against our proposed YOLO-GARNet model, showcasing the effectiveness of our integrated garment analysis and recommendation model.

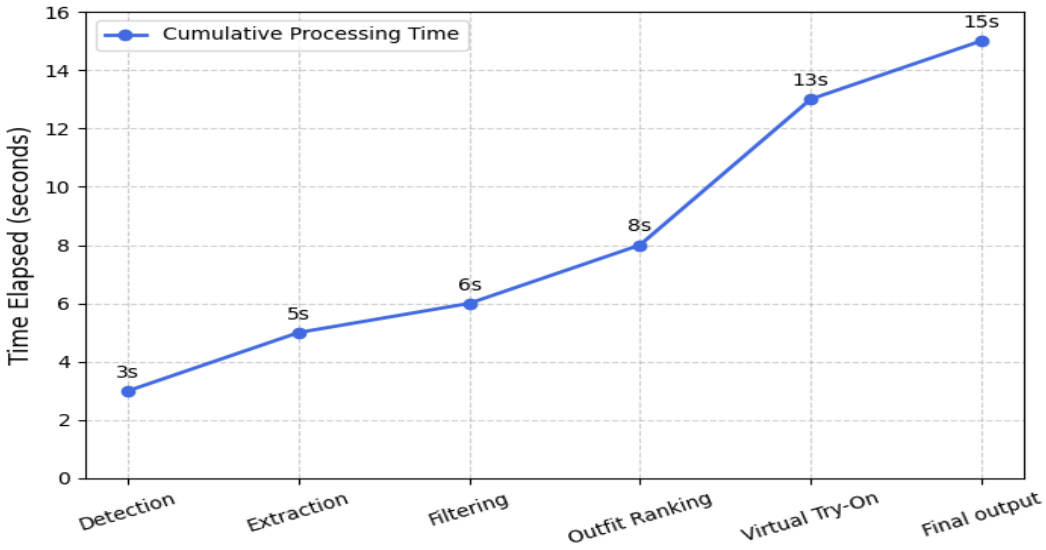


Fig. 13: Cumulative Processing Time Across End-to-End Fashion Recommendation Pipeline

extraction and segmentation. The Filtering stage reaches 6 seconds cumulative time for processing and refining detected garments. Outfit Ranking takes the processing time to 8 seconds as the system evaluates and scores potential clothing combinations. The Virtual Try-On phase significantly increases processing time to 13 seconds, reflecting the computational complexity of generating realistic virtual fitting results. Finally, the Final Output stage completes at 15 seconds total processing time for delivering the complete fashion recommendation. The entire pipeline takes approximately 15 seconds to process, and the most computationally expensive operations are outfit ranking and virtual try-on. User satisfaction (Figure 11 (b)) is highest between 5-25 outfit combinations, which represents the optimal balance between offering variety and avoiding decision paralysis. Comparative analysis (Table II) demonstrates the superior computational efficiency of the proposed approach. YOLO-GARNet achieves significantly faster detection times (3.2s) compared to traditional two-stage detectors like Faster R-CNN (8.7s) and DeepFashion2-RCNN (11.2s). The optimized EfficientNetB0 classification component outperforms heavier architectures, requiring only 2.8s versus VGG-based approaches that demand 6.8s. Most notably, the ranking algorithm demonstrates exceptional efficiency at 4.5s, representing a 64% improvement over YOLOv8-based systems (8.4s) and a 76% improvement over Context-Aware Fashion methods (18.9s).

With good precision (0.98) and recall (0.94) values, the system excels in garment detection and identification. Uniform outfit (Figure 11 (b)) suggestions are thus achieved, although the slight drop in mAP@0.5:0.95 (0.91 validation) indicates minor localization issues at stricter IoU thresholds. While garment detection remains robust for most subjects, better boundary detection would improve visual realism in the virtual try-on, particularly for garments with strong silhouettes.

A. System Limitations and Future Enhancements

Despite excellent performance, the system faces technical limitations such as suboptimal rendering of materials like chiffon or silk in the virtual try-on and difficulties with layered clothing or accessories fitting. Data processing and cleaning are also significant challenges, as garments need to be accurately labeled and matched to facilitate effective recommendations. Future work will focus on further developing the cloth simulation model for more realistic fabric behavior and adding a more sophisticated layering system. Additionally, incorporating accessories into the recommendation system and reducing user latency will enhance the user experience and improve recommendation compatibility with individual preferences.

1) Challenge: Determining Appropriateness of Events for Particular Clothes: Traditional models like YOLOv9 and EfficientNetB0, designed for object detection and general image classification, lack contextual understanding for event-appropriate clothing selection. **Solution**: Implementation of specialized Python libraries such as fashion-ai-analysis and VisualFashionAttributePrediction enables extraction of rich clothing features including color, pattern, and style attributes, allowing the system to determine appropriate occasions for each outfit and improve recommendation relevance.

2) Scalability and Latency in Recommendation Systems: The recommendation system's response time (T_{response}) scales linearly with the number of garments (N_{garments}) and required compatibility checks (C_{outfits}), and is expressed as:

$$T_{\text{response}} = \frac{N_{\text{garments}}}{C_{\text{outfits}}} \tag{15}$$

Complex wardrobes with numerous potential combinations can create computational bottlenecks and slower response times.

Solution: Implementation of pre-filtering processes that select promising garment combinations based on predefined compatibility rules significantly reduces C_{outfits}. Additionally, optimization algorithms such as Multi-Objective Differential

Evolution (MODE) can accelerate EfficientNetB0 convergence, resulting in faster and more accurate outfit suggestions.

3) Improving Outfit Recommendation Visualization: **Challenge:** Traditional list-based outfit presentations fail to provide an intuitive understanding of the ensemble's appearance, which can reduce user engagement and confidence in system recommendations.

Solution: Integration of Virtual Try-On (VTO) functionality allows users to visualize suggested clothing ensembles on their own bodies by overlaying garments onto user photographs or digital avatars. This provides realistic previews that significantly enhance the user experience and strengthen the system's value proposition.

VI. Conclusion

This system architecture demonstrates our innovative one-stop solution that revolutionizes online fashion shopping by seamlessly integrating advanced clothing detection, intelligent recommendation algorithms, and photorealistic virtual try-on capabilities. The end-to-end pipeline leverages cutting-edge technologies including YOLOv9 for precise garment segmentation and classification, EfficientNetB0 for personalized style recommendations, and Virtual Try-On Network (VTON) for immersive fashion visualization, creating a unified platform that transforms the traditional e-commerce experience into an interactive, personalized, and visually engaging shopping journey.

References

[1] V. Aakash, A. Akmal, A. Chandrababu, and A. Vijay, "A comprehensive survey on ai-driven fashion technologies: Clothing detection, recommendation systems, and virtual try-on solutions," *International Journal of Advances in Engineering and Management*, vol. 2024, no. 11, p. 6, 2024.

[2] S. Elsayed, L. Brinkmeyer, and L. Schmidt-Thieme, "End-to-end image-based fashion recommendation," in *Workshop on Recommender Systems in Fashion and Retail*. Springer, 2022, pp. 109–119.

[3] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[4] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1096–1104.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778, 2016.

[6] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[7] D. Morelli, M. Fincato, M. Cornia, F. Landi, F. Cesari, and R. Cucchiara, "Dress code: High-resolution multi-category virtual try-on," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2231–2235.

[8] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5337–5345.

[9] S. Abdellaoui and I. Kachbal, "Apparel e-commerce background matting," *Int. J. Adv. Res. Eng. Technol.(IJARET)*, vol. 12, no. 3, pp. 421–429, 2021.

[10] E. A. Said and K. Ilham, "Deep residual network for high-resolution background matting," *Stud. Inf. Control*, vol. 30, no. 3, pp. 51–59, 2021.

[11] S. El Abdellaoui and I. Kachbal, "Deep background matting," in *The Proceedings of the International Conference on Smart City Applications*. Springer, 2022, pp. 523–532.

[12] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European conference on computer vision*. Springer, 2024, pp. 1–21.

[13] I. Kachbal, S. El Abdellaoui, and K. Arhid, "Revolutionizing fashion recommendations: A deep dive into deep learning-based recommender systems," in *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security*, 2024, pp. 1–8.

[14] I. Kachbal, S. E. Abdellaoui, and K. Arhid, "Fashion recommendation systems: From single items to complete outfits," *International Journal of Computer Engineering and Data Science (IJCEDS)*, vol. 4, no. 1, pp. 27–40, Apr. 2025.

[15] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," pp. 173–182, 2017.

[16] E. Purificato, L. Boratto, and E. W. De Luca, "User modeling and user profiling: A comprehensive survey," *arXiv preprint arXiv:2402.09660*, 2024.

[17] B. Suvarna and S. Balakrishna, "Enhanced content-based fashion recommendation system through deep ensemble classifier with transfer learning," *Fashion and Textiles*, vol. 11, no. 1, p. 24, 2024.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[19] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023.

[20] R. G. Baldovino, A. J. P. Vidad, R. P. B. Abastillas, N. T. Bugtai, E. P. Dadios, R. R. P. Vicerra, A. A. Bandala, A. R. See, and N. R. Roxas Jr, "Comprehensive analysis on ultralytics-supported yolo models for detection and recognition of large office objects for indoor navigation," *Procedia Computer Science*, vol. 246, pp. 3851–3858, 2024.

[21] X. Yang, X. Du, and M. Wang, "Learning to match on graph for fashion compatibility modeling," in *Proceedings of the AAAI Conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 287–294.

[22] B. S. Priya, R. L. S. Ram, M. Manojkumar, T. R. Vennam, V. Murthy, and G. B. Regulwar, "Efficientnetb0-based deep learning approach for crop growth stage classification," in *2024 2nd International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR)*. IEEE, 2024, pp. 1–6.

[23] G. Sharma, K. Abraham *et al.*, "Transfer learning approach for multilabel fine grain image attribute extraction," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2022, pp. 1521–1524.

[24] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[25] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450.

[26] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[27] S. A. Zhang Shuai, Yao Lina and T. Yi, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[28] A. Aayesha, M. Afzaal, and J. Neidhardt, "User experience of recommender system: A user study of social-aware fashion recommendations system," in *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, 2024, pp. 356–361.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," vol. 28, 2015.

[30] R. Varghese and M. Sambath, "Yolov8: A novel object detection algorithm with enhanced performance and robustness," in *2024 International conference on advances in data engineering and intelligent computing systems (ADICS)*. IEEE, 2024, pp. 1–6.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[32] M. Muthivhi, T. van Zyl, and H. Wang, "Multi-modal recommendation system with auxiliary information," in *Southern African Conference for Artificial Intelligence Research*. Springer, 2022, pp. 108–122.

[33] H. M. Abd Alaziz, H. Elmannai, H. Saleh, M. Hadjouni, A. M. Anter, A. Koura, and M. Kayed, "Enhancing fashion classification with vision transformer (vit) and developing recommendation fashion systems using dinova2," *Electronics*, vol. 12, no. 20, p. 4263, 2023.

[34] J. Kim, G. Gu, M. Park, S. Park, and J. Choo, "Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 8176–8185.