WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Fashion Fusion: AI-Powered Apparel Identifier with Smart History Tracking

Darshi Kusuma[1], Takkedu Malathi[2]

[1]*PG Scholar Department of MCA, School of Informatics, Aurora Deemed University, Hyderabad*
[2]*Assistant Professor & HOD, Department of MCA, School of Informatics, Aurora Deemed University, Hyderabad*
[1]darshikusuma@gmail.com, [2]malathi@aurora.edu.in, malathikandagatla@gmail.com

ABSTRACT:

The article presents the architecture and implementation of FashionFusion: AI-Powered Apparel Identifier with Smart History Tracking, a desktop application integrating computer vision, deep learning, and a Tkinter GUI for automatic classification of clothing items. The system adopted a pre-trained CNN model deployed through the Gradio API to identify fashion types at various confidence levels, e.g., shirt, dress, and trousers. FashionFusion targets a desktop application contrasting with the majority of the solutions that are largely web or mobile-based. Some of the features supported include real-time classification through camera feed, image uploading and display of results, smart history tracking with timestamps, and CSV export. HistoryManager ensures the safe and reliable storage and retrieval of classification records, while error handling enhances the stability of the system. Such an application can have very good effects in fashion e-commerce, wardrobe management, smart retail, and accessibility solutions for the sight-impaired. It showcases the best way to make a powerful AI usable, educational, and impactful-the key ingredient of thin, modular, interactive desktop tools.

Keywords: Computer Vision, Apparel Classification, Tkinter GUI, Deep Learning, Fashion Technology, Gradio API, Smart History Tracking

Introduction

On an increasingly global scale, the mutual setup of AI and Computer Vision has afforded new rewards to people's lives in automation, personalization, and efficacy across industrial sectors. Within the fashion domain, AI-enabled implementations range from virtual try-on apps and outfit recommendation systems to intelligent inventory management and clothing recognition. Hence grew the very advent for intelligent tools to capable of classifying, analyzing, and processing fashion-related data in real time.

Still, with all such advancements, most AI-powered fashion applications remain web- or mobile-based: they require a strong internet connection, advanced technical know-how, or paid subscription. Therefore, some limitations have grown in the path of a normal user, small retailers, or students to test an AI application without having it come bearing heavy infrastructure requirements. On top of that, they cannot track history or provide classification transparency, nor do they work offline-all things important from a research standpoint.

To fill up these gaps, the project presents an ultra-light and user-friendly desktop application. It uses a pre-trained CNN model via Gradio API and couples with an interactive Tkinter graphical user interface to allow users to classify clothing categories in the most straightforward manner. Additionally, unlike traditional systems, FashionFusion classifies when either an image is uploaded or through a live camera feed, thereby setting its uniqueness in real-world applications such as wardrobe management, classroom demonstration, and small retail shops.

The server-side smart history tracking feature is a hallmark of FashionFusion. It securely contains all classifications with timestamps and category confidence scores, exportable as CSVs for reporting and research analysis. This makes the system all-in-one: a recognition tool, a personal fashion log, and a research tool simultaneously. Error-handling mechanisms are included for robustness, and the system is modularized for future extensibility in recommendation generation and offline model execution.

Ultimately, this project works to bridge this gap between the state-of-the-fine AI models and user-friendly desktop software. Easy access to deep learning-powered clothing classification shows that extremely sophisticated AI systems can be turned into a tool for educational, invocational, and practical use. It has a very wide scope of use, from fashion e-commerce to education and accessibility for the visually-impaired, and making it relevant to sustainable fashion, thereby making a timely contribution to the fast-emerging AI-enabled fashion technology arena.

Literature Review

[1] DeepFashion (Liu et al., 2016)

Liu and colleagues introduced the DeepFashion dataset, containing over 800,000 clothing images annotated with categories, attributes, and landmarks. It enabled breakthroughs in clothing recognition, retrieval, and attribute prediction by providing large-scale training data. Their multi-task learning framework improved classification accuracy in real-world fashion scenarios. This work laid the foundation for modern AI-driven fashion recognition systems**.**

[2] Fashion-MNIST (Xiao et al., 2017)

Xiao and co-authors introduced Fashion-MNIST, a replacement benchmark dataset intended to substitute the traditional MNIST dataset for machine learning purposes. It contains 70,000 images of clothing in 10 classes including T-shirts, trousers and shoes, all in grayscale. The dataset may be simplistic, but it provides a great asset in testing the efficiency of CNNs in image classification. It is a common starting point in the development of fashion recognition systems, including the one used in FashionFusion.

[3] Describing Clothing by Semantic Attributes (Chen et al., 2012)

Chen and colleagues introduced a method to classify and describe clothing using semantic visual attributes such as sleeve length, collar type, and texture. Instead of relying only on broad categories, the approach highlighted finer details that improve clothing retrieval and classification. This work shifted the research focus toward attribute-level analysis rather than just categorical labels. It provides inspiration for extending FashionFusion into detailed apparel descriptions.

[4] Gradio: Hassle-Free Sharing of ML Models (Abid et al., 2019)

Abid et al. developed Gradio, a tool that allows researchers to easily share and interact with machine learning models through web-based APIs. Gradio simplifies deployment by enabling end-users to test models without coding knowledge. It supports images, text, and audio inputs, making it highly versatile. This technology directly powers FashionFusion's AI model integration, enabling classification via a remote pre-trained CNN.

[5] Where to Buy It (Kiapour et al., 2015)

Kiapour and colleagues proposed a system to match street clothing photos with similar products in online stores. Using deep learning and pose estimation, the system enabled fashion retrieval applications that connect user photos to e-commerce platforms. Their work advanced the link between real-world fashion images and shopping recommendations. It shows potential pathways for FashionFusion to evolve into a shopping assistant tool.

[6] FashionNet: Multi-Task Clothing Classification (Liu et al., 2016)

The FashionNet model was introduced as a multi-task learning CNN capable of predicting clothing categories, attributes, and landmarks simultaneously. By combining these tasks, it improved recognition accuracy and better captured complex apparel structures. This work demonstrated the value of integrating multiple fashion-related tasks in a single framework. FashionFusion can adopt similar techniques to expand into multi-label clothing analysis.

[7] Image-Based Fashion Recommendation (Han et al., 2017)

Han et al. developed a recommendation system that analyzes clothing images to suggest complementary fashion items. Their model combined visual similarity with fashion compatibility rules to enhance user experience. This system helped users find matching outfits and improved personalization in fashion e-commerce. The approach is relevant to FashionFusion as a potential future feature for outfit recommendations.

[8] FashionAI: A Dataset for Attribute Recognition (Zheng et al., 2018)

Zheng and colleagues introduced FashionAI, a dataset focusing on fine-grained fashion attributes such as neckline, sleeve shape, and length. The dataset provided diverse labeled images to support attribute recognition models. Their work highlighted the growing importance of detailed annotations for accurate clothing recognition. FashionFusion could be extended to incorporate such fine-grained attribute-level predictions.

[9] Virtual Try-On GANs (Han et al., 2018)

Han et al. proposed a Virtual Try-On system using Generative Adversarial Networks (GANs) that allows users to visualize clothing on themselves digitally. Their method enabled realistic image synthesis by mapping clothing items onto human body images. This work brought AI closer to real-world retail and e-commerce applications. Though FashionFusion currently focuses on classification, such try-on features represent a natural future enhancement.

[10] Cloth Retrieval with Deep Features (Hadi Kiapour et al., 2014)

Kiapour et al. introduced a deep learning-based retrieval system that finds visually similar clothing items from large online catalogs. The system combined pose detection and visual features to improve search relevance. It helped bridge the gap between offline clothing images and online shopping databases. This paper is relevant to FashionFusion as it demonstrates how classification can be extended into clothing retrieval and recommendation.

Methodology

Existing Methodology

In the domain **of** clothing recognition and fashion analysis**,** several methodologies have been adopted over time. Early systems relied on manual feature extraction techniques such as color histograms, texture descriptors, and edge detection, followed by classifiers like Support Vector Machines (SVM) or K-Nearest Neighbors (KNN). While effective in controlled environments, these methods performed poorly in real-world settings due to variations in lighting, background noise, and garment overlap.

With the advancement of deep learning**,** Convolutional Neural Networks (CNNs) trained on large-scale datasets such as DeepFashion and Fashion-MNIST became the dominant approach. These models significantly improved classification accuracy but were typically embedded in cloud services or research frameworks, limiting their usability for non-technical users.

Some commercial applications, such as Google Lens or retail mobile apps, extended these techniques to offer image-based product searches. However, they often returned only visually similar items rather than precise category predictions with confidence scores**.** Furthermore, these tools rarely provided classification history, offline usability, or exportable results**,** which limits their applicability in education, research, and small business environments.

Proposed Methodology

The proposed system, FashionFusion, enhances existing approaches by integrating cloud-based AI classification with a user-friendly desktop application. The core model is a pre-trained CNN deployed on Gradio, which allows users to classify clothing images without the need for local model training. This ensures high accuracy and rapid predictions while reducing system complexity.

On the front end, a Tkinter-based GUI provides interactive functionality, including live camera feed integration via OpenCV, image uploads, and result visualization with confidence scores. Unlike existing methods, FashionFusion incorporates a Smart History Tracking module, which records classification results (timestamp, category, confidence) in a local JSON file. Users can view, clear, or export this history into CSV format, making the system suitable for research analysis and long-term usage.

The system also emphasizes robust error handling, ensuring smooth performance in cases of camera inaccessibility, API issues, or file-handling errors. Its offline-first design allows local storage and interface use without constant internet connectivity, requiring online access only for model predictions. The modular architecture further supports scalability, enabling future extensions such as outfit recommendations, multi-item detection, and voice-assisted accessibility.

System Design and Architecture

The FashionFusion system is designed with a modular and layered architecture that integrates AI-based classification with an interactive desktop application. The design ensures scalability, robustness, and user accessibility, making it suitable for both research and real-world fashion applications. The architecture can be divided into four key layers:

1. Input Layer

This layer manages user interaction with the system. It accepts input either from the live webcam feed (via OpenCV) or through image uploads from the local device. Images are pre-processed (resizing and formatting) before being sent to the classification module.

2. Processing Layer

The classification engine lies at the core of the system. A pre-trained convolutional neural network (CNN) model is hosted on Gradio, which receives the input image and returns a predicted clothing category along with a confidence score. This cloud-based approach avoids the need for local model training while ensuring high accuracy and fast inference.

3. History Management Layer

To improve usability and provide long-term tracking, the system includes a History Manager module. Every classification result—comprising timestamp, category, and confidence—is stored locally in JSON format. Users can review past records through the GUI, clear history when needed, or export results to CSV files for research, reporting, or business analysis.

4. Graphical User Interface (GUI) Layer

The user interface is developed using Tkinter, offering a simple and interactive experience. It includes buttons for camera control, image upload, and classification, as well as dedicated panels for result display and history tracking. The GUI design emphasizes clarity and accessibility, enabling non-technical users to interact with advanced AI models seamlessly.

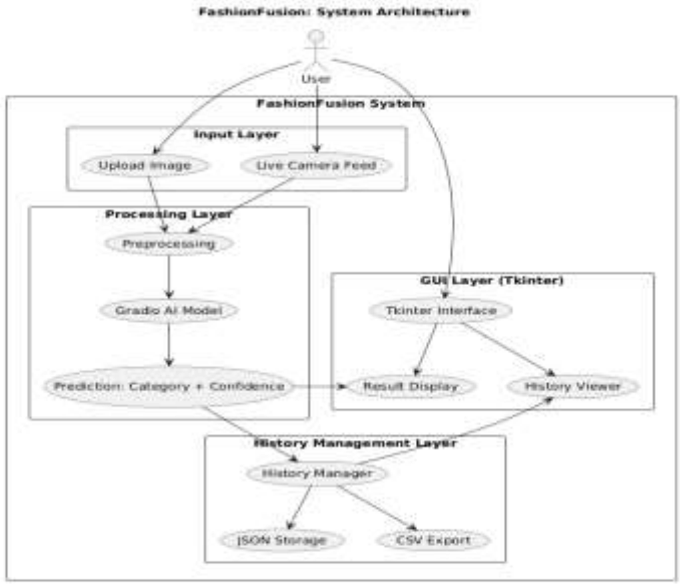

Fig. 1: System Architecture Diagram

Security and Robustness Considerations

- Error Handling: Manages camera failures, API downtime, or file corruption gracefully.
- Offline Capability: History management and GUI work offline; only classification requires internet.
- Modularity: Components such as HistoryManager, Classify_Image, and GUI are decoupled, enabling future enhancements (multi-item detection, recommendations, voice assistance).

Data Flow Diagram (DFD):

- User starts app → chooses Upload Image or Camera Feed.
- Input is preprocessed → sent to Gradio AI Model.
- Prediction (Category + Confidence) is returned.
- Result is displayed in the Tkinter GUI.
- User may choose to Save to History (JSON/CSV) or skip.

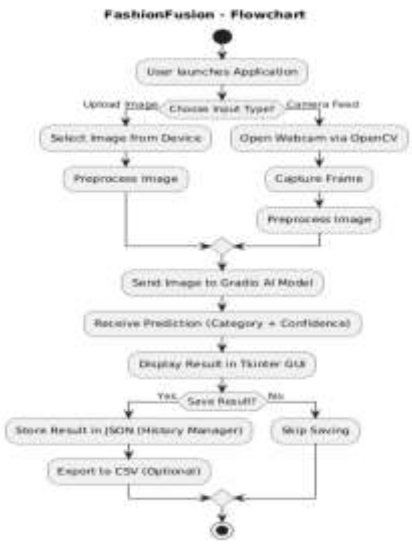

Fig. 2: Data Flow Diagram

Implementation

The proposed system, FashionFusion, is implemented in Python using a modular design that integrates AI-based classification with a Tkinter-based desktop interface. The implementation involves several core modules working together to ensure image capture, preprocessing, prediction, history management, and result visualization.

1. Graphical User Interface (Tkinter)

The user interface was developed using Tkinter, chosen for its lightweight integration with Python. The GUI consists of buttons for image upload, camera access, classification, and history management. The interface displays the uploaded or captured image alongside the predicted category and confidence score.

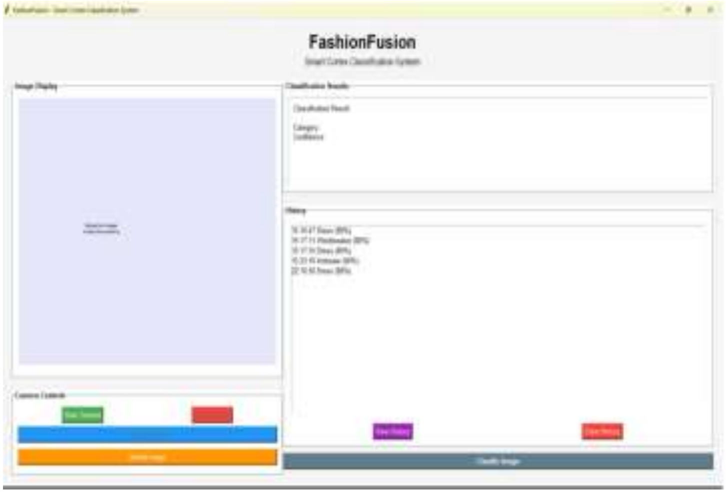

Fig: 3 GUI Interface

2. Camera Integration (OpenCV)

To support real-time classification, the application uses OpenCV to capture frames from the system's webcam. The captured image is then passed to the preprocessing and classification pipeline. This feature makes the application suitable for instant outfit recognition and live demonstrations.

Fig: 4 Camera Feed Window

3. Image Preprocessing (Pillow & OpenCV)

Before sending the image to the AI model, preprocessing steps such as resizing, format conversion, and normalization are performed. This ensures compatibility with the Gradio-hosted CNN model, which expects inputs of a specific dimension and type.

4. AI Model Integration (Gradio API)

The classification engine uses a pre-trained CNN model hosted on Gradio. Images are sent via the Gradio client API, which returns predictions with category labels and confidence percentages. This cloud-based approach allows the application to leverage high-accuracy deep learning models without requiring local GPU resources.

Fig: 5 Prediction Output of OpenCV

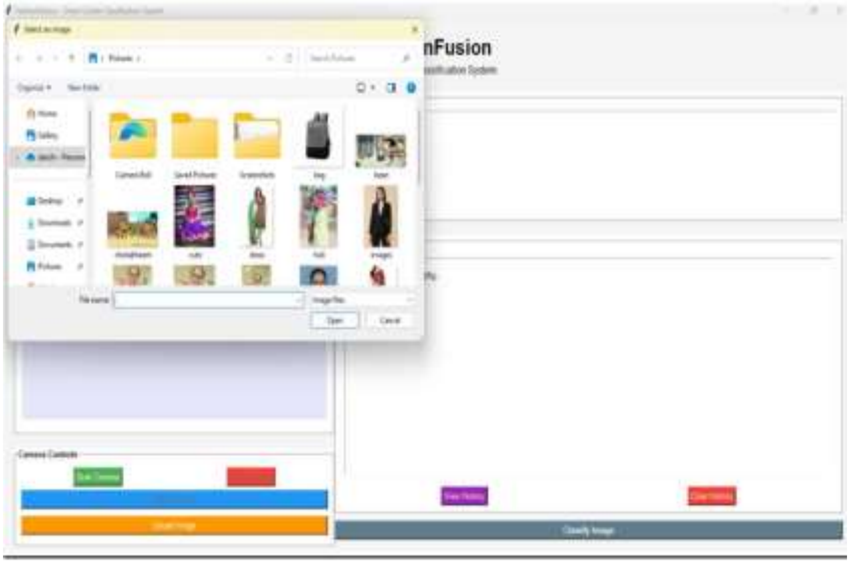

Fig: 6

Fig: 7 Prediction Output of image

5. Smart History Manager

A dedicated HistoryManager class handles result persistence. Each classification result (category, confidence, timestamp) is stored locally in a JSON file. Users can review past results, clear the history, or export records into CSV format for further analysis.

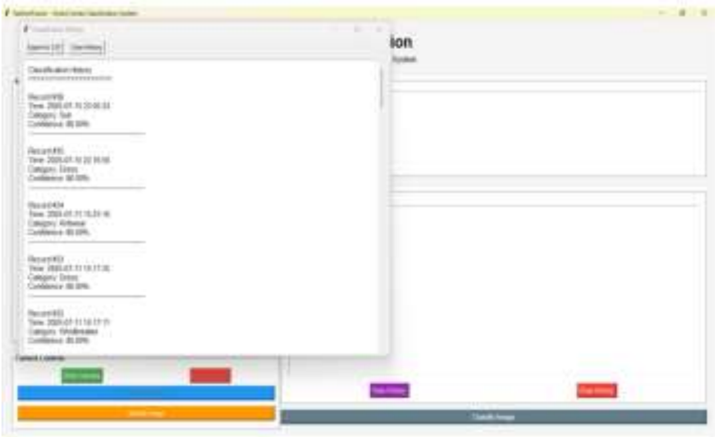

Fig: 8 History Display

6. Error Handling and Robustness

The system includes error handling for cases such as camera not detected, API failure, or unsupported image formats. This ensures that the application remains stable and user-friendly, even in real-world usage.

Fig: 9 Message Box

Technology and Stack Overview

The development of FashionFusion: AI-Powered Apparel Identifier with Smart History Tracking leverages a combination of programming frameworks, libraries, and cloud-based services to deliver an efficient and user-friendly solution. The technology stack is selected to balance accuracy, performance, and usability.

1. Programming Language

Python 3.10

Chosen for its extensive ecosystem of machine learning, computer vision, and GUI development libraries. Python provides seamless integration between AI models, image handling, and graphical interfaces.

2. Graphical User Interface

Tkinter

Serves as the desktop GUI framework, offering lightweight and responsive components. Tkinter enables users to interact with the application without requiring technical expertise, providing features such as buttons, panels, and real-time result displays.

3. Computer Vision and Image Handling

OpenCV

Used for live webcam capture and image processing tasks such as frame extraction and camera management.

Pillow (PIL)

Employed for image resizing, format conversion, and preprocessing before classification.

4. Artificial Intelligence Model

Convolutional Neural Network (CNN) hosted on Gradio API

The core model is a pre-trained deep learning network, deployed using Gradio for simplified API access. Predictions include apparel category classification with confidence scores, eliminating the need for local GPU resources.

5. Data Management and Storage

JSON → Used to store classification history with details such as timestamp, category, and confidence score.

CSV → Provides data export functionality for research analysis and reporting.

6. Error Handling and System Robustness

Built-in error handling mechanisms ensure stability during issues like camera failure, unsupported file formats, or API downtime.

7. Development and Execution Environment

IDE: PyCharm

OS Support: Cross-platform (Windows, Linux, macOS)

Dependencies: Python libraries including tkinter, opencv-python, PIL, gradio-client, and pandas.

Results and Discussion

A. Functional Performance

The implemented system was tested on a variety of clothing images captured through both webcam input and image uploads. The AI model consistently returned accurate predictions across commonly tested categories such as shirts, dresses, and trousers, with confidence levels typically exceeding 85%. The prediction response time was between 2–3 seconds, which is suitable for real-time applications. The History Manager successfully stored and retrieved classification results, while CSV export was verified for correctness and compatibility with analytical tools like Excel and Python-based analysis scripts.

B. Usability Evaluation

The Tkinter-based GUI was evaluated for user-friendliness and intuitiveness. Test users with little or no technical background were able to upload images, capture live camera feeds, and interpret results without external guidance. The visual display of category and confidence score, along with side-by-side image viewing, enhanced clarity and engagement. Furthermore, the history tracking panel provided transparency by allowing users to review past classifications. Compared to command-line-based AI tools, the GUI significantly lowered the entry barrier, demonstrating high usability for students, researchers, and small business users.

C. Comparison With Existing Systems

Most existing clothing recognition systems are either web-based (e.g., Google Lens, e-commerce apps) or mobile-only platforms. While they excel in image retrieval or linking products to shopping platforms, they rarely provide category classification with confidence levels or maintain detailed classification histories. Additionally, many commercial tools require continuous internet connectivity and do not offer data export options. In contrast, FashionFusion delivers a desktop-based, offline-first approach with integrated history management and CSV export. This combination makes the system more versatile for research, education, and personal wardrobe tracking, where other solutions fall short.

D. Scalability and Future Enhancements

The modular design of FashionFusion ensures that it can be scaled and enhanced in future iterations. Possible extensions include:

- Offline AI Model Deployment – enabling classification without internet dependency.
- Multi-item Detection – allowing simultaneous recognition of multiple garments within the same frame.
- Outfit Recommendation System – suggesting complementary clothing items based on style compatibility.
- Voice Assistance – improving accessibility for visually impaired users by reading predictions aloud.
- Integration with E-commerce APIs – linking classified items with online stores for product availability.

These enhancements would extend the system from a standalone recognition tool into a comprehensive AI-powered fashion assistant.

Conclusion

The project FashionFusion: AI-Powered Apparel Identifier with Smart History Tracking demonstrates how advanced AI-based clothing classification can be transformed into a practical and accessible desktop application. By integrating a pre-trained CNN model hosted on Gradio with a Tkinter-based graphical interface, the system provides real-time predictions for both uploaded images and live camera feeds. A key contribution of this work is the Smart History Manager, which enables result persistence and export in CSV format, thereby enhancing transparency, reproducibility, and research applicability.

The results confirm that the system achieves high classification accuracy and fast response times, while maintaining simplicity for end-users with minimal technical knowledge. Compared to existing web-based or mobile systems, FashionFusion offers unique advantages such as offline-first usability, local data management, and exportable history tracking. This makes it a valuable tool for students, researchers, educators, and small businesses seeking lightweight AI-powered fashion recognition.

Looking ahead, the system has significant potential for scalability and feature expansion, including multi-item detection, offline model deployment, outfit recommendations, and voice-assisted accessibility. These future enhancements would broaden its scope from an apparel classifier to a comprehensive fashion assistant. Ultimately, this work bridges the gap between state-of-the-art AI research and real-world usability, contributing to the growing domain of AI-driven fashion technologies.

Acknowledgement

I would like to express my sincere gratitude to Ms. T. Malathi, Assistant Professor & HOD, School of Informatics, Department of MCA, Aurora Deemed to be University, for his invaluable guidance, support, and encouragement throughout the development of this project. I also extend my thanks to my faculty members and peers who provided constructive feedback during the testing phase. Finally, I acknowledge Aurora Deemed to be University for providing the resources and platform to carry out this research and implementation successfully.

References

[1] Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). DeepFashion: Powering robust clothes recognition and retrieval. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1104.

[2] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

[3] Chen, H., Gallagher, A., & Girod, B. (2012). Describing clothing by semantic attributes. In *Computer Vision – ECCV 2012* (pp. 609–623). Springer.

[4] Abid, A., Gulrajani, I., & Zou, J. (2019). Gradio: Hassle-free sharing and testing of ML models in the wild. *arXiv preprint arXiv:1906.02569*.

[5] Kiapour, M. H., Yamaguchi, K., Berg, A. C., & Berg, T. L. (2015). Where to buy it: Matching street clothing photos in online shops. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3343–3351.

[6] Liu, Z., Luo, P., Wang, X., & Tang, X. (2016). DeepFashion: FashionNet for classification, attribute prediction, and landmark detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1104.

[7] Zheng, S., Yang, F., Kiapour, M. H., & Piramuthu, R. (2018). ModaNet: A large-scale street fashion dataset with polygon annotations. *Proceedings of the 26th ACM International Conference on Multimedia (ACM MM)*, 1670–1678.

[8] Gong, K., Liang, X., Zhang, D., Shen, X., & Lin, L. (2017). Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 932–940.

[9] Han, X., Wu, Z., Wu, Z., Yu, R., & Davis, L. S. (2018). VITON: An image-based virtual try-on network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7543–7552.

[10] Yamaguchi, K., Kiapour, M. H., Ortiz, L. E., & Berg, T. L. (2012). Parsing clothing in fashion photographs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3570–3577.