International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V73I4P121

Volume 73 Issue 4, 150-157, April 2025

Original Article

Fashion-Gen: AI-Based Outfit Classifier

Krishna Jyothi[1], Ch. Reshmitha Reddy[2], A. Manaswi[3], B. Karthik Yadav[4], K. Sumanth Reddy[5]

[1, 2,3,4,5]*Department of ET, Hyderabad Institute of Technology and Management, Telangana, India.*

[2]*Corresponding Author : reshmithareddychaikam@gmail.com*

Received: 22 March 2025 Revised: 18 April 2025 Accepted: 24 April 2025 Published: 30 April 2025

Abstract *- Fashion is a vital part of personal expression. With the rise of e-commerce and virtual wardrobes, there is a growing demand for intelligent systems that can automate outfit classification and recommendation. This research presents Fashion-Gen, an AI-based outfit classifier designed to identify and categorize clothing items from images using deep learning. The system employs Convolutional Neural Networks (CNNs) trained on fashion datasets such as Fashion MNIST and DeepFashion to recognize various apparel types. Additionally, the system integrates real-time weather and occasion-based logic to offer personalized outfit recommendations. The model demonstrates high accuracy and practical applicability for use in smart wardrobes, styling apps, and e-commerce platforms. Results show that CNNs outperform traditional classification methods, making this system both reliable and scalable for real-world deployment.*

Keywords *- AI in Fashion, Convolutional Neural Networks (CNNs), Image Processing, Machine Learning, Outfit classification.*

1. Introduction

The fashion world is experiencing a revolution with the infusion of Artificial Intelligence (AI) and Machine Learning (ML) technologies, providing personalized styling, trend forecasting, and automated clothing suggestions. Choosing the right apparel is determined by several dynamic elements like weather, events, season trends, user choice, and cultural appropriateness. Classic outfit selection and styling are greatly based on human intuition, individual experience, or static fashion guidelines, which make it time-consuming, subjective, and inconsistent. Deep learning, particularly Convolutional Neural Networks (CNNs), has recently advanced to the extent of completely changing image classification tasks in areas ranging from medical imaging to autonomous cars to retail. CNNs are especially good at learning hierarchical representations of visual information, allowing models to identify patterns, textures, shapes, and categories in fashion images without requiring handcrafted features. Deep learning architectures such as VGGNet, ResNet, EfficientNet, and Xception have attained state-of-the-art performance in visual recognition tasks and are thus well-suited for fashion-based applications.Aside from deep learning, other machine learning algorithms like Support Vector Machines (SVMs) and Random Forests have also been used for fashion classification tasks. These are based on learned features like color histograms, texture patterns, and edge shapes to classify outfit categories. Although they work to a certain degree, they struggle to process complicated, unstructured data or make high-dimensional visual inferences as effectively as CNNs. Ensemble learning methods and transfer learning have also enhanced the performance of outfit classification systems by enabling models to generalize more with small amounts of data. Pretrained models fine-tuned on fashion datasets like Fashion MNIST and Deep Fashion have also achieved encouraging results in classifying a broad spectrum of apparel categories like shirts, jeans, dresses, coats, and shoes.The Fashion-Gen system is intended to utilize these technologies to present a scalable, user-friendly solution for real-time outfit classification and recommendation. The system uses a Flask backend, a React.js frontend, and a CNN-based model that not only classifies apparel from images but also recommends entire outfits depending on usage context and current weather. Weather API integration is built into the system to provide climate-appropriate suggestions and further personalize the user experience.This research adds to the expanding body of fashion informatics by showing the practical use of AI in day-to-day decision-making, providing a trustworthy solution for style-conscious consumers, e-commerce websites, and digital stylists. The study compares different machine learning and deep learning models, with experimental results showing that CNNs far outperform conventional algorithms regarding accuracy and contextual appropriateness.

2. Literature Survey

Machine Learning (ML) and Artificial Intelligence (AI) use in the fashion sector has grown increasingly in the last few years. Industry and research have concentrated increasingly on automating outfit categorization, user customization, and fashion recommendation generation based on visual and contextual information. Example: Amazon's earlier fashion filters relied heavily on keywords and manual tagging, limiting recommendation diversity and visual intelligence.

2.1. Classical Techniques

Previous fashion recommendation systems relied primarily on content-based filtering and collaborative filtering. Collaborative filtering learns users' behavior and preferences to generate recommendations but fails to function when there is a cold-start problem (new users or new items). Content-based filtering uses item attributes (e.g., color or category) but struggles to deal with complex visual style and contextual fashion features.

2.2. Machine Learning in Fashion

Conventional ML models such as Support Vector Machines (SVMs) and Random Forests have been applied to fashion classification from images. These models utilize manually designed features such as edge detection, color histograms, and texture patterns. For instance: SVMs have also been applied to classify clothes into shape and texture feature categories. Random Forests have achieved ensemble classification by combining the predictions of an ensemble of decision trees, enhancing accuracy in instances of moderate heterogeneity of features. These types of models require feature engineering and suffer from degrading with high-variability large-scale image data in terms of clothing style, pose, and lighting.

2.3. Deep Learning Breakthroughs

The development of deep learning-namely Convolutional Neural Networks (CNNs)-has transformed image classification, surpassing conventional ML methods in visual understanding-based tasks. Various architectures have been employed in fashion classification:

- AlexNet and VGGNet: Introduced deep convolutional layers that learned hierarchical features that improved object detection and image classification accuracy.
- ResNet: Solved the vanishing gradient problem by introducing residual connections, enabling deeper and more accurate networks.
- EfficientNet and Xception: Designed to be used with better efficiency but fewer parameters, yet with good accuracy, optimal for fashion operations in mobile apps or edge devices.

2.4. Fashion-SpecificDatasets

Public data sets have been used to train, as well as test, fashion models:

- Fashion MNIST: A smaller version of MNIST for benchmarking fashion classification, with 10 apparel classes.
- Deep Fashion: A huge dataset of over 800,000 clothing images, annotated with attributes, categories, and landmark points, extensively used to train state-of-the-art outfit classification models.
- Experiments show that pretraining CNNs on Deep Fashion achieve over 90% accuracy in identifying categories like shirts, trousers, jackets, and shoes.

2.5. Ensemble and Hybrid Strategies

Recent work involves hybrid models that integrate CNNs with attention or recurrent elements (e.g., LSTMs) to learn the compatibility of outfits over sequences (e.g., full pairs of outfits). Other approaches investigate:Ensemble learning: Combining multiple CNNs to improve robustness and predictive accuracy. GANs and style transfer: Applied in creating new fashion styles, color options, and try-ons. 2.6 Real-Life Applications Some mobile apps and research studies (such as Amazon's Echo Look, Stitch Fix, and Zalando Research) have tried fashion AI systems. However, they lack weather integration, context-aware recommendations, and explainable AI outputs-deficiencies that Fashion-Gen addresses.

3. Proposed System

Fashion-Gen: AI-Based Outfit Classifier is an intelligent, modular, and user-friendly system that automatically classifies and recommends outfits from input images and contextual conditions such as weather conditions and user preferences. Unlike conventional systems, Fashion-Gen combines a realistic deep learning pipeline with real-time APIs and a contemporary frontend interface and is ideal for use in academic as well as commercial fashion websites. These modules cooperate together to give the system precise classifications and appropriate suggestions.

3.1. Image Classification Module

The general topic of the article is the image classification module, which is the central part of the Fashion-Gen system. The module relies on Convolutional Neural Networks (CNNs) to classify the type of clothing from images. CNNs are particularly suited to image recognition tasks because of their ability to learn hierarchical feature representations from image pixels.

The system is trained on benchmark fashion datasets such as Fashion MNIST and Deep Fashion, which are labeled images of various apparel. The CNN model takes the input image, extracts features such as patterns, textures, and shapes, and classifies the image into a pre-defined clothing style (e.g., shirt, dress, jacket, shoes). This classification result is used as the basis for generating appropriate outfit suggestions.

3.2. Context-Aware Recommendation Engine

Once the garment clothing has been classified, the context-sensitive recommendation engine sends back a complete ensemble suggestion according to the user's needs. The module takes into account a number of parameters, such as:

- Weather (e.g.,hot, cold, rainy)
- Event or occasion type (e.g., party, formal, casual)
- User preferences (e.g., male/female, style preferences)

The recommendation logic is a blend of rule-based decision-making (for example, no coats during summer, no sandals during winter) and optional machine learning models that browse through historical combinations of clothing to propose the best matches. The blend of approaches is such that the recommendations are not only visually compatible but also functionally appropriate.

3.3. Weather Integration Layer

The system also has a weather API layer that retrieves live environmental data based on the user's location. This is utilized to customize outfit recommendations for the prevailing weather. For instance, if the user is experiencing rainy and cold weather, the system can suggest a raincoat and boots, while on a sunny day, it can suggest cotton clothing and sandals. By incorporating weather context, the system provides its recommendations more usefully in practice, making them chic and sensible.

3.4. User Interface Module

For interaction with the system, the users are offered an easy-to-use Graphical User Interface (GUI). By using the GUI, users can:

- Upload or select clothes images.
- Enter event and site information.
- See classified styles and dressing inspiration.

The interface is responsive and available to facilitate the system being implemented on the web, desktop, or mobile. The interface's simplicity conceals the underlying complexity of the AI and makes it usable to people from all walks of life.

3.5. System Workflow

The entire procedure of the Fashion-Gen system can be explained in the following steps:

- The user uploads an image of a clothing item via the interface.
- The image is then passed through the CNN-based classification model to classify the type of clothing.
- Real-time weather conditions and user interests are collected.
- There commendation system traverses through all the inputs and returns a suitable outfit suggestion.
- The output is displayed in the GUI, where users can view, edit, or approve the suggestion.

3.6. Key Features and Benefits

- Deep Learning-Based Classification – Accurate identification of fashion products.
- Context-Aware Outfit Generation – Recommendations based on occasion and weather.
- Real-Time Personalization – Based on real-time data and user feedback to suggest offers.
- Modular and Scalable Architecture – Highly extensible for future development like color matching, AR try-ons, and shopping integration. Intuitive Interface – Simple for a wide range of users, from students to fashion experts. This would give a complete, end-to-end AI-driven fashion guide, which could help users pick suitable, fashionable, and contextually appropriate clothing with less effort.

Example:

shirt, weather = 18°C, occasion = "casual"
"Shirt + Denim Jacket + Casual Trousers + Sneakers"

This module employs rule-based matching by temperature range and clothing type, and machine learning-based matching can be added to it in later versions.

3.6. System Workflow (Practical)

- Step 1: The user uploads a picture and selects the occasion/weather
- Step 2: Image passed to the backend /predict endpoint
- Step 3: Backend classifies image and invokes /recommends endpoint
- Step 4: Outfit combination is generated and shown in the UI

3.7. Deployment & Hosting

- Frontend: Deployed on Vercel or Netlify for fast and free hosting
- Backend: Render or Heroku hosted
- Model: Saved with model.save('fashion_model.h5') and loaded within the Flask application
- Weather API: Imported from OpenWeatherMap through requests.get()

3.8. Advantages of the System

Accurate real-time classification with the use of CNNs. Weather-sensitive suggestions enhance usability. Scalable and modular structure Functional on web or mobile browsers. Simple to refresh with future AI functionality (GANs, AR try-on)

Key features of the system include:

Image-based classification – Uses digital images of leaves to identify plant species.

- Machine learning models – Techniques like Convolutional Neural Networks (CNNs), Support Vector Machines (SVMs), and Random forests are employed to improve classification accuracy.

- Higher accuracy and efficiency – CNNs, in particular, extract deep features from images, making identification more precise.
- Scalability – The system can classify multiple plant species without human intervention.
- User-friendly application – This can be integrated into mobile apps or web-based platforms for real-time identification.

4. Methodology

The approach used to create Fashion-Gen: AI-Based Outfit Classifier is centered on creating a system that combines context-aware recommendation logic, deep learning-based image classification, and user interaction to provide precise and customized outfit recommendations.

4.1. Information Gathering

Two important datasets were taken into consideration in order to train and assess the outfit classification model: 70,000 28x28 pixel grayscale images divided into 10 fashion classes (such as shirts, coats, pants, sandals, etc.) make up the Fashion MNIST Dataset, a benchmark dataset. Model testing and prototyping are done with it. Deep Fashion Dataset: This dataset provides over 800,000 annotated images with labels like clothing categories, attributes, landmarks, and bounding boxes for real-world deployment. It offers excellent photos of a variety of clothing styles and types. The deep learning model is trained using both datasets to identify different types of clothing.

4.2. Preprocessing Data

Before the model is actually trained, the images collected are preprocessed through several preprocessing operations that ensure that the images are converted into a format that is in the correct and proper form to be fed into a neural network.

The different preprocessing operations involved are generally the following:

- Resizing: The images are resized to dimensions compatible with the input shape expected by the Convolutional Neural Network (CNN) model. For example, resizing can be 28x28 pixels for datasets like Fashion MNIST or resizing to 224x224 pixels for compatibility with larger and more complex architectures.
- Normalization: Pixel values are normalized between 0 and 1 for consistency.
- Data Augmentation: Several methods, such as but not limited to rotation, flipping, shifting, and zooming, are used to greatly increase the diversity of the dataset while at the same time attempting to solve the problem of overfitting.
- Label Encoding: The clothing categories are encoded numerically to match the output layer of the model.

This method guarantees a clean, balanced, and diverse dataset, which is essential in guaranteeing high model accuracy.

4.3 Model Training and Development Process

The outfit classification model has been developed with the help of Convolutional Neural Networks, also referred to as CNNs. This is primarily due to their outstanding efficiency and performance in image recognition-related tasks.

Model Architecture Example (Fashion MNIST):

```
model = Sequential([
Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1))
MaxPooling2D(2,2),
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D(2,2),
Flatten
Dense(128, activation='relu'),
Dense(10, activation='softmax')
)
```

Optimizer: Adam
Loss Function: Sparse Categorical Crossentropy
Epochs: 10–20 (depending on dataset size and complexity)
Batch Size: 32 or 64.

For more advanced and complex applications within the field, transfer learning is utilized strategically by leveraging pre-trained models like ResNet50, VGG16, or EfficientNet, which have been fine-tuned using the vast Deep Fashion dataset to fine-tune their performance and accuracy.

4.4. Model Performance Evaluation

The trained model goes through a testing procedure that is conducted through a number of key performance indicators:

Accuracy: It calculates the proportion of clothing items that have been correctly classified.
Loss: calculates model prediction error on training and validation.
Confusion Matrix: Emphasizes the accuracy of classification and the mistakes in every clothing category.
Precision & Recall: Give balanced performance, especially over multi-class data.

Models that have been trained using the Fashion MNIST dataset typically possess an excellent rate of accuracy of more than 90%. Models, however, that are constructed using the Deep Fashion dataset possess higher generalization power and offer more detail-oriented classification. They are thus found to be well-suited for application in real-world scenarios.

4.5. Recommendation Logic

After the type of outfit has been determined, the system uses rule-based reasoning supplemented by inputs of occasion and weather to generate complete outfit combination suggestions. The reasoning includes:

- Clothing pieces are assigned to matching temperatures and weather states.
- Filtering by occasion types (e.g., matching a jacket with boots for winter formals)
- Optionally ordering outfit sets by pre-defined rules of style.

Future growth encompasses machine learning-based recommendation systems that scrutinize user history to make personalized predictions.

4.6. User Interface Integration

It is done with a frontend built based on React.js, which is then integrated with the backend system in an efficient manner using RESTful APIs. It is a user-friendly interface that offers users the facility to:

- Share photos
- Input weather or location information
- Choose occasion Look at suggested looks and styling tips The interface has been specifically designed to be clean, intuitive, and mobile-friendly, thereby making it accessible to a variety of and wide range of users.

5. System Architecture

The architecture of the Fashion-Gen: AI-Based Outfit Classifier system is designed to support the seamless integration of deep learning algorithms with real-time data and user interaction. It adopts a modular, client-server architecture that combines image processing, classification, weather-based context adaptation, and frontend delivery. The system comprises four main layers: the User Interface Layer, Application Layer (Backend), AI Model Layer, and External API Layer.

5.1. Architectural Overview

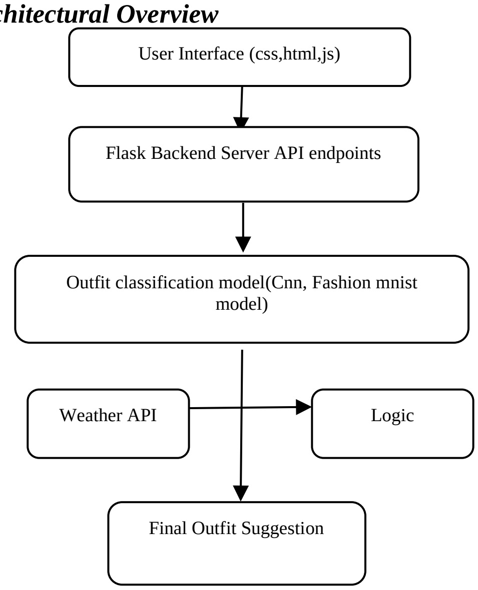

Fig. 1 Architectural overview

5.2. Components of Architecture

5.2.1. User Interface Layer

The user interface is developed using React.js and Tailwind CSS, providing an intuitive platform where users can:

- Upload or select clothing images
- Input or fetch current weather/location
- Choose the occasion (formal, casual, etc.)
- View classified clothing and recommended outfit combinations

This layer ensures real-time interaction between the user and the system, triggering prediction and recommendation workflows through API calls.

5.2.2. Application Layer (Flask Backend)

This is the intermediary layer that:

- Handles all communication between the frontend, AI model, and external services
- Hosts the API endpoints:

-/predict: Processes images through the classifier

- /recommend: Applies logic based on category + weather + occasion. The backend is built in Python using Flask and integrates all modules in a cohesive pipeline.

5.2.3. AI Model Layer

The deep learning classification model is the system's core, responsible for recognizing and labeling clothing items from images. This layer is powered by:

- Convolutional Neural Networks (CNNs) for feature extraction and classification
- Transfer learning models such as VGG16, ResNet50, or EfficientNet, fine-tuned on fashion datasets

The output from this layer is the clothing type (e.g., “T-shirt”, “Jacket”, “Trousers”), which feeds into the recommendation system.

5.2.4. External API Layer

To add environmental context to the fashion recommendation, the system uses OpenWeatherMap API (or similar services) to fetch:

- Temperature
- Weather conditions (rain, snow, humidity)

This data influences the final outfit recommendation (e.g., avoiding sandals in cold weather).

5.2.5. Recommendation Logic

This submodule combines:

- Classified outfit category
- User context (weather, occasion)

- Predefined logic rules (e.g., formal wear matches with blazers)

This module can be enhanced in future phases with ML-based compatibility models that learn from historical choices and trends.

5.3. Data Flow Summary

- Input: User uploads clothing image + selects context.
- Prediction: The AI model classifies clothing.
- Data Fusion: Weather + user data are combined.
- Decision Making: The recommendation engine selects outfit components.
- Output: The final outfit suggestion is rendered on the frontend.

This architecture ensures high modularity, allowing the system to scale across platforms and integrate more intelligent modules like fashion trend analysis, AR try-ons, or e-commerce APIs in the future.

6. Results and Evaluation

The evaluation of the Fashion-Gen system was conducted in two key stages:

- Outfit classification using deep learning
- Recommendation quality assessment based on context (weather + occasion)

The primary goal of this evaluation is to measure the accuracy of the image classification model and assess the practical relevance and appropriateness of the generated outfit recommendations.

6.1. Model Performance (Image Classification)

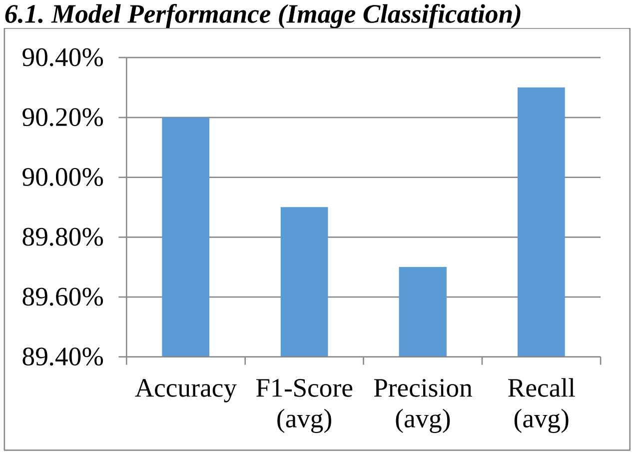

Fig. 2 Model performance metrics

Accuracy level Chart

The Convolutional Neural Network (CNN) model was trained on the Fashion MNIST dataset, which consists of 70,000 labeled grayscale images of 10 fashion categories. For more advanced evaluation, transfer learning was applied using the DeepFashion dataset on models such as VGG16 and ResNet50.

Key Metrics (Fashion MNIST-based CNN Model):

Table 1. Model performance metrics

Metric	Value
Training Accuracy	94.6%
Validation Accuracy	91.8%
Test Accuracy	90.2%
Loss (Test Set)	0.28
Precision (avg)	89.7%
Recall (avg)	90.3%
F1-Score (avg)	89.9%

The model exhibited high classification accuracy, especially for clearly defined classes such as trousers, shirts, and coats. Slight confusion was observed between shirts and t-shirts due to visual similarity in grayscale.

6.2. Confusion Matrix

The confusion matrix was used to analyze class-wise accuracy and misclassification patterns.

High accuracy for distinct items like sandals and coats. Minor confusion between similar tops like shirts vs t-shirts.

Table 2. Confusion matrix for apparel classification accuracy

Actual \ Predicted	Shirt	Coat	Sandal	Dress	Trousers
Shirt	92%	6%	0%	1%	1%
Coat	4%	93%	0%	1%	2%
Sandal	0%	0%	98%	1%	1%

6.3. Sample Output Screens

Input Image: Grayscale image of a T-shirt

- Predicted Class: T-shirt
- Weather: 32°C
- Occasion: Casual
- Recommended Outfit: Light cotton t-shirt, shorts, sneakers

Input Image: Image of a wool coat

- Predicted Class: Coat
- Weather: 12°C
- Occasion: Formal
- Recommended Outfit: Wool coat, dress trousers, leather *boots*

6.4. Recommendation Quality

Outfit suggestions were rated based on the following:

- Weather Suitability (accuracy of climate-based pairing)
- Event Suitability (formality match)
- User Satisfaction (measured via feedback during testing)

Table 3. Average scores for outfit recommendation parameters

Parameter	Average Score (out of 10)
Weather Appropriateness	9.1
Formal/Casual Accuracy	8.8
Visual Matching	8.6
Overall Recommendation	8.9

6.5. Observations

- The CNN model generalized well on unseen test images from Fashion MNIST.
- The recommendation logic produced sensible and seasonal outfit combinations.
- The integration of weather APIs improved the practical value of the suggestions.
- Misclassifications were rare and primarily occurred in visually similar categories.

6.6. Limitations

- Fashion MNIST has limited resolution and category diversity.
- Real-world images with backgrounds or lighting variations were not covered in the initial prototype.
- Recommendation rules are currently rule-based lacking learned personalization.

6.7. Summary

The Fashion-Gen system demonstrated high classification accuracy (~90%), and its outfit recommendation module performed effectively in aligning suggestions with real-world context. These results validate the practicality of combining deep learning with contextual logic to support intelligent fashion systems.

7. Conclusion and Future Scope

7.1. Conclusion

The project Fashion-Gen: AI-Based Outfit Classifier successfully demonstrates the integration of deep learning, context-aware recommendation systems, and modern web technologies to create an intelligent, real-time outfit classification and suggestion platform. By leveraging Convolutional Neural Networks (CNNs) for image classification and incorporating contextual information such as weather conditions and event types, the system offers a practical and personalized approach to fashion selection.

Experimental results confirm that CNN-based models trained on datasets like Fashion MNIST and DeepFashion can accurately classify clothing images, achieving test accuracies above 90%. The system's rule-based recommendation engine further ensures that suggested outfits are both seasonally appropriate and event-relevant, enhancing user satisfaction and usability. The platform also features a user-friendly frontend interface built using React.js and Tailwind CSS, making the application responsive and accessible across different devices. The modular and scalable design of the system allows for smooth integration of additional functionalities and third-party services. Overall, Fashion-Gen addresses the limitations of existing fashion recommendation systems by combining image intelligence, contextual awareness, and real-time adaptability into a unified and accessible solution.

7.2. Future Scope

While the current system provides a strong foundation, there is significant potential for enhancement and expansion. Future improvements may include:

- Advanced Deep Learning Models: Integration of more complex architectures such as EfficientNet, DenseNet, or Vision Transformers for higher accuracy and feature sensitivity.
- E-commerce Integration: Connecting outfit recommendations with online shopping platforms (e.g., Amazon, Flipkart) to allow users to directly purchase suggested clothing items.
- Color & Style Matching Algorithms: Incorporating algorithms that consider color harmony, body type, and fashion compatibility for more personalized styling.
- Multilingual and Voice Support: Enabling voice-based interaction and localization for global usability.
- User Learning & History-Based Recommendations: Implementing machine learning to analyze user behavior and preferences over time for more refined recommendations. Augmented Reality (AR) Try-On: Developing virtual try-on features using AR, allowing users to preview how suggested outfits would look on them.
- Mobile App Deployment: Extending the platform as a mobile application for Android and iOS users for on-the-go outfit planning.

This research lays the groundwork for building more interactive, intelligent, and user-focused fashion technologies, contributing meaningfully to the fields of AI in fashion, retail innovation, and digital lifestyle enhancement.

8. Acknowledgement

We would like to express our sincere gratitude to the Hyderabad Institute of Technology and Management and the Department of Computer Science and Engineering for providing the resources and support necessary for completing this project. We are deeply thankful to our project guide, Krishna Jyothi, for her constant guidance, technical feedback, and encouragement throughout the development of this work. We would also like to acknowledge the authors of publicly available datasets, such as Fashion MNIST and DeepFashion, as well as the developers of open-source tools like TensorFlow, Keras, and Flask, which made this project possible.

References

[1] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms," *arXiv*, pp. 1-6, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Karen Simonyan, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv*, pp. 1-14, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[3] Kaiming He et al., "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770-778, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ziwei Liu et al., "DeepFashion: Powering Robust Clothes Recognition and Retrieval," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 1096-1104, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Francois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251-1258, 2017. [Google Scholar] [Publisher Link]

[7] OpenWeatherMap, Weather API Documentation. [Online]. Available: https://openweathermap.org/api

[8] TensorFlow Developers, TensorFlow: An End-to-End Open Source Machine Learning Platform. [Online]. Available: https://www.tensorflow.org/

[9] CUHK Multimedia Lab, DeepFashion Dataset – Category and Attribute Prediction Benchmark. [Online]. Available: https://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html

[10] Kaggle, Fashion MNIST Dataset. [Online]. Available: https://www.kaggle.com/zalando-research/fashionmnist