Meesho Data Challenge 2024

Team: Neural Ninjas

Kushal Agrawal[1], Nachiketa Purohit[2], Alli Khadga Jyoth[3], and Ritu Singh[4]

[1]kushal12345kushal@gmail.com

[2]nachiketapuro@gmail.com

[3]khadgajyothalli@gmail.com

[4]ritutweets46@gmail.com

1 Source Code

The complete implementation of our approach is available at the following link: `Code`

2 Brief Summary

We developed a solution for Predicting Attributes from Product Images [1], focusing on accurately predicting attributes for a variety of e-commerce product categories. By combining robust vision-language models with a category-aware prediction framework, our method demonstrated strong capabilities in tackling multi-category, multi-attribute tasks. Notably, our approach achieved a **Score** [1] of **0.802** while maintaining an impressive **inference time** of only **0.05 seconds** per image on **1 NVIDIA T4 GPU**.

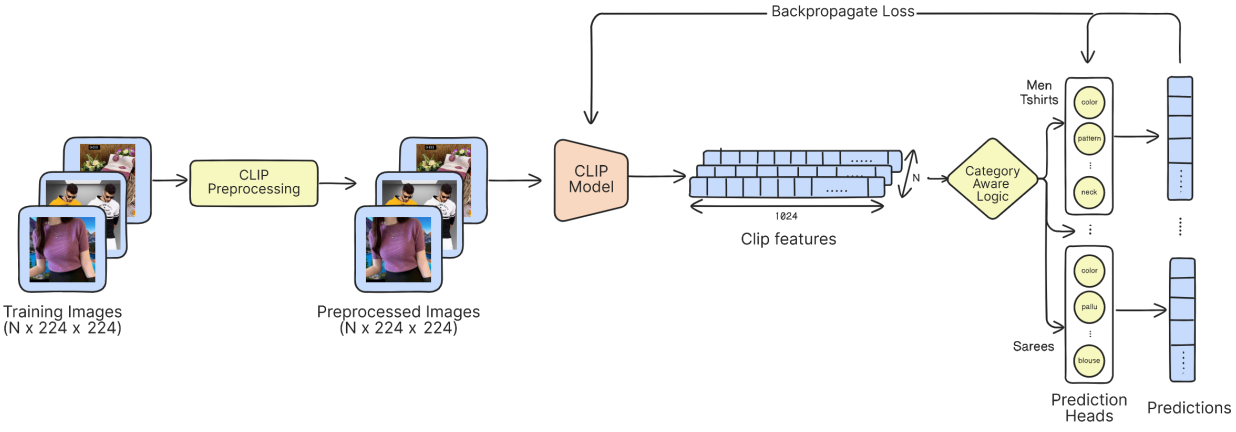

Figure 1: Training Pipeline of Final Approach

Our approach utilized a dual-backbone architecture, integrating two models: **CLIP** [2] **ViT-H/14-quickgelu** [3] (pretrained on **DFN-5B**) and **ConvNext-XXLarge** [4] (pretrained on **LAION-2B**). The core approach was our **Category-Aware Attribute Predictor**, which used MLP-based heads specifically designed for different category-attribute combinations. This system included flexible prediction paths, optimized layer, ReLU activations, and regularization techniques like layer normalization and dropout. To finalize predictions, we used an ensemble method, giving equal weight (0.5 each) to both models to combine their strengths.

We started with the ViT-H/14-quickgelu [3] model and MLP heads, which gave excellent results early on, performing well on validation metrics and public leaderboards. We got a **Private Score** of **0.800** without ensemble technique. We also tested other methods, including Visual Question Answering (VQA) with the Qwen2-VL-7B-instruct [5] model, which worked well for context but was slower and less accurate than our classification approach. Similarity-based methods, such as hash-based and deep feature based searches were also explored but didn't improve the results.

We put significant effort into optimizing the models, including hyperparameter tuning and ensemble experiments with ViT-H-14-quickgelu [3] and ConvNext-XXLarge [4]. Throughout, we focused on balancing accuracy with computational efficiency. Our best ensemble model takes **0.05 seconds** on **1 NVIDIA T4 GPU** to predict attribute values of an image . From this, we learned that simpler, well-tuned models often perform better than complex ones.

In summary, our solution shows that success comes from choosing the right models, designing thoughtful architectures, and carefully optimizing them. With this approach, we set a high standard for multi-category product classification in e-commerce while keeping the solution practical and efficient.

3 Exploratory Data Analysis and Data Preprocessing

Our exploratory data analysis highlighted several critical challenges that influenced our data preprocessing and feature engineering strategies:

1. **Class Distribution Analysis:** We observed a significant class imbalance in the training dataset, where certain attributes were heavily overrepresented while others were rare. This imbalance posed a risk of model bias towards majority classes, potentially leading to poor performance on minority classes.

2. **Missing Value Assessment:** A substantial portion of the attribute values in the training dataset were missing. This sparsity necessitated careful handling to ensure that the model could effectively learn without being adversely impacted by incomplete data.

3. **Label Consistency Analysis:** We identified cases where identical images were assigned conflicting labels. This label noise introduced ambiguity in the training data, which could adversely

affect model performance.

4. **Image Quality Assessment:** Our analysis revealed several distorted or low-quality images, which could potentially reduce the effectiveness of the model in learning meaningful features.

3.1 Data Cleaning Steps

To improve the dataset's quality, we experimented with several data cleaning and preprocessing approaches. However, **none of these methods** listed below resulted in improved validation accuracy or public leaderboard performance, and they were ultimately excluded from our final solution:

1. **Background Removal:** We attempted to remove image backgrounds to focus on primary objects, but this did not yield any noticeable improvements in model performance.

2. **Image Similarity-Based Label Correction:** Using features extracted with CLIP ViT-L/14 [3], we implemented an image similarity search. For each image, the labels were replaced using majority voting among the top 15 most similar images. This method failed to enhance results and occasionally introduced noise.

3. **Subset Creation and Balancing:** A balanced subset of the training data was created, addressing class imbalances through upsampling of minority classes. Unfortunately, this strategy also did not improve performance.

In our final solution, we did not use any of the above methodologies, as none aligned well with the dataset's inherent characteristics or contributed to performance gains.

3.2 Training and Validation Data Split Strategy

To ensure robust evaluation, we implemented a carefully designed training-validation split strategy:

- Random splitting was performed using PyTorch's random split functionality.
- Manual verification checks ensured stratification across classes to maintain consistent distributions in both training and validation sets.

This approach aimed to balance class representation across splits, mitigating risks associated with the dataset's inherent imbalance.

4 Feature Engineering

4.1 Image Preprocessing Pipeline

To maintain consistency and enhance input quality we used the respective preprocessing pipeline provided by open clip models:

- **Resolution Standardization:** All images were resized to fixed dimensions of 224px for ViT-H/14-quickgelu and 256px for ConvNext-XXLarge model.
- **Normalization:** Channel-wise normalization was applied based on OpenCLIP preprocessing standards to standardize pixel values.
- **Color Space Transformations:** All images were transformed to a standardized color space to minimize variations due to different capture environments.

4.2 Data Augmentation Strategy

To improve generalization and increase data diversity, we employed both basic and advanced augmentation techniques:

1. **Geometric Transformations:**
    - Random cropping with size preservation.
    - Horizontal flipping with a probability of 0.5.
    - Rotation within defined angles to simulate real-world variability.
2. **Color Space Augmentations:**
    - Applied color jittering with controlled parameters to introduce variations in hue, saturation, and brightness.
    - Adjusted brightness and contrast to enhance robustness to different lighting conditions.
3. **Advanced Techniques:**
    - **Mixup:** Combined two images and their labels using adaptive alpha parameters to create blended samples.
    - **CutMix:** Randomly replaced a region of one image with a patch from another, controlling the size and location of the region.

5 Model Selection

In tackling the problem of Product Attribute prediction, we systematically evaluated various cutting-edge methodologies. Each method was scrutinized for its performance, scalability, and suitability for the task, ultimately leading us to a robust ensemble-based solution. In section 5.1 we give a detailed account of the methods explored, including their key features, rationale, and outcomes. The final model selection is given in section 5.2. Please refer Table 1 for the leaderboard results.

Approach Type	Model & Technique	Public Score	Private Score
VQA using VLM	Finetuned Qwen2VL-7B instruct model using VQA	0.551	0.583
Image Similarity Based Search with majority voting	Hashing	0.337	0.342
	SeResNext model and Faiss	0.670	0.669
	Swin Transformer and Faiss	0.606	0.605
	Frozen ClipViT B/32	0.723	0.724
	Frozen ClipViT L/14	0.777	0.778
Classification Based w/ MLP Head	ClipViT-B/32	0.765	0.765
	ClipViT-L/14	0.770	0.771
	ClipViT-L/14 optimal params	0.785	0.785
	ClipViT-L/14 w/background removal	0.782	0.779
	Coca	0.797	0.794
	ConvNext-XXLarge	0.801	0.799
	ViT-H/14-quickgelu	**0.806**	**0.800**
Ensemble Based	ViT-H/14-quickgelu + Coca	0.804	0.801
	ConvNext-XXLarge + ViT-H/14	**0.807**	**0.802**

Table 1: Comparison of different approaches and their corresponding model performances.

5.1 Explored Approaches

5.1.1 Visual Question Answering (VQA)

We initially framed the problem as a Visual Question Answering (VQA) task, leveraging the state-of-the-art model **Qwen2-VL-7B-Instruct** [5], key features of the model are:

- Pre-trained on VQA and OCR datasets
- Vision-Language Alignment
- Dynamic Resolution

To improve performance, we used carefully crafted prompts to predict the attributes directly from the images. While this approach demonstrated the potential of contextual reasoning, it yielded a **Score** of **0.58**.

Challenges: However, due to the high computational complexity and limited scalability of this method, we decided to explore alternative approaches that would be more computationally efficient and better suited for large-scale applications.

5.1.2 Image Similarity Search

1. Hashing-Based Similarity Search [6]
    - A lightweight technique that uses image hashing algorithms to retrieve visually similar images quickly. Attributes were predicted using majority voting from the top-k retrieved samples.

- This method provided a computationally inexpensive baseline for exploring similarity-based approaches, suitable for initial experimentation.

Challenges and Exclusion: Hashing failed to capture the nuanced details necessary for distinguishing complex fashion attributes, resulting in poor accuracy.

2. Deep Feature-Based Cosine Similarity

   - In this approach we utilized various Deep Learning models i.e. SeResNext [7], Swin Transformer [8], CLIP ViT [3] etc.
   - **Key Features:** Features were extracted from these models after pretraining, and FAISS [9] indexing with cosine similarity was used to retrieve top-k similar samples for a test image. Predictions were made through majority voting on non-null attribute values from these top-k samples.
   - **Rationale:** This approach capitalized on the representational power of pre- trained models and the efficiency of FAISS indexing. Notably, CLIP ViT outperformed other models due to its contrastive pretraining, which aligns image-text representations and enhances feature quality.

   Challenges and Exclusion: While CLIP ViT showed strong performance in similarity search, this method lacked scalability and robustness compared to classification-based approaches, particularly when dealing with a large variety of fashion attributes

5.1.3 Classification-Based Methods

We explored several models during our experimentation. CLIP ViT-B/32 [3] was tested for its fast inference, though its feature representations were only moderately accurate. CLIP ViT-L/14 [3] showed improved feature extraction and was effective for fine-grained attribute prediction. We also tried ConvNext-XXLarge [4], a scalable ConvNet with strong classification performance, and CoCa [10], a multi-modal model leveraging vision and text inputs for generalization. However, we selected CLIP ViT-H/14-quickgelu [3] and ConvNext-XXLarge [4] as our final model due to their state-of-the-art accuracy, making it the best fit for our needs. Please refer 1 for the training pipeline.

Key Features: The models were fine-tuned with **41 MLP heads**, one for each unique (category_attribute_name). These heads were trained alongside the backbone weights, framing the problem as a multi-class classification task. The loss function updated both the MLP weights and the backbone to achieve optimal feature representation and attribute prediction.

Rationale for Exploration: Classification directly aligns with the problem's requirements, offering a structured approach to multi-attribute prediction. Models like **ViT-H/14-quickgelu** and

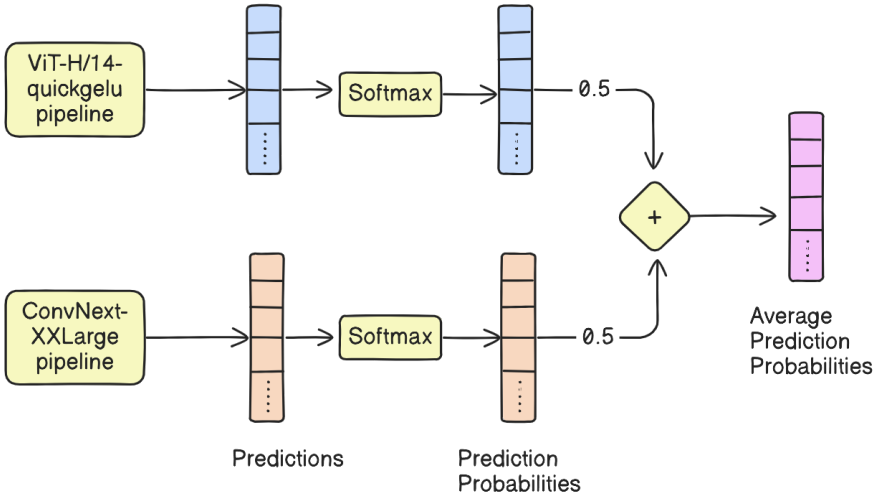

Figure 2: Ensemble Approach Inference

ConvNext-XXLarge excelled due to their large capacity and efficient architectures, achieving more than 0.8 public leaderboard scores.

Challenges and Exclusion: While single models like ViT-H/14-quickgelu performed exceptionally well, the diverse nature of fashion attributes highlighted the potential for further improvement via model ensembling.

5.2 Final Ensemble Methodology

5.2.1 Model Architecture:

Our final model is a weighted ensemble of two cutting-edge architectures: ViT-H/14-quickgelu and ConvNext-XXLarge. The predictions from these models were combined using a weighted averaging mechanism to optimize overall performance 2. Details of architecture can be viewed from Table 2.

The ensemble effectively leverages the complementary strengths of these two architectures; ViT-H/14-quickgelu's fine-grained detail extraction and ConvNext-XXLarge's robustness in handling classification complexities.

5.2.2 Rationale for Selection:

The ensemble was selected as the final model due to its exceptional performance across multiple metrics:

- Achieved the highest validation accuracy and top public score of **0.807** in the evaluation phase.
- Demonstrated a unique ability to handle the multi-attribute prediction complexity, offering both scalability and accuracy.
- By combining the distinct strengths of ViT-H/14-quickgelu and ConvNext-XXLarge, the model provided a balanced solution capable of addressing diverse prediction challenges with consistency.

5.2.3 Outcome:

The weighted ensemble emerged as the definitive solution, outperforming alternative approaches. Its superior performance and robustness made it the ideal choice for deployment.

5.2.4 Balancing Accuracy and Scalability:

Early methods, such as VQA and similarity search, provided valuable insights into understanding data distribution and attributes. However, they lacked scalability for large-scale attribute prediction.

5.2.5 State-of-the-Art Architectures:

The classification-based ensemble leveraged advanced architectures to ensure high performance, making it particularly suited for large and complex datasets like Meesho's e-commerce platform.

Architecture	**Component**	**Description**
ViT-H/14-quickgelu	Patch Embeddings	14×14 pixel patches, flattened and projected with added positional embeddings.
	Transformer Encoder	Multi-head self-attention with FFN, QuickGELU activation, and layer normalization.
	QuickGELU Activation	Efficient approximation of GELU activation function.
	MLP Head	Multi-layer perceptron with dropout and softmax for classification.
ConvNext-XXLarge	Stem Layer	4×4 non-overlapping convolutions (stride 4) with Layer Normalization.
	ConvNext Blocks	7×7 depthwise separable convolutions followed by pointwise convolutions, with GELU and Layer Normalization.
	Block Architecture	Each block contains: - Depthwise Conv (k=7×7) - Layer Normalization - Pointwise Conv (expansion ratio 4) - GELU activation - Pointwise Conv (projection) - Stochastic Depth for regularization
	Network Stages	Four progressive stages with increasing channel dimensions: - Stage 1: 384 channels - Stage 2: 768 channels - Stage 3: 1536 channels - Stage 4: 3072 channels Each stage downsamples feature maps by 2×
	Global Processing	Global Average Pooling after final stage.
Prediction Head		Linear layer with Layer Normalization.

Table 2: Architecture details of ViT-H/14-quickgelu and ConvNext-XXLarge

5.3 Hyperparameter Tuning

Hyperparameter fine-tuning was performed using a exhaustive search approach. We conducted hyperparameter tuning for the parameters listed in Table 3, identifying their optimal values through systematic experimentation. The final values were chosen after multiple iterations of training to ensure the best trade-off between model performance and computational efficiency.

Hyperparameter	**Values**
clip_lr	1e-5, 1e-4, 5e-6
predictor_lr	5e-5, 5e-4, 1e-3
weight_decay	0.001, 0.05, 0.1
beta1	0.9, 0.95
beta2	0.999, 0.9999
hidden_dim	256, 768, 1024
dropout_rate	0.1, 0.3, 0.4, 0.5
num_hidden_layers	1, 2, 3

Table 3: Parameter grid for hyperparameter tuning.

5.4 Final Hyperparameters

The final hyperparameters for the training of **Both OpenCLIP [2] Models with MLP head** are summarized in the table 4, along with their justifications:

Hyperparameter	**Value**	**Justification**
clip_lr	$1e-5$	Chosen to prevent overfitting and ensure stable training while fine-tuning the CLIP [2] model.
predictor_lr	$5e-5$	Slightly higher learning rate to allow focused optimization of the final classification layers.
weight_decay	0.001	Regularization to prevent overfitting.
beta1	0.9	Standard choice for first momentum term in Adam optimizer to accelerate convergence.
beta2	0.999	Standard choice for second momentum term in Adam optimizer for stability.
hidden_dim	256	Hidden dimension for MLP head, balancing model complexity and computational efficiency.
dropout_rate	0.1	Dropout used in MLP layers for regularization to prevent overfitting.
num_hidden_layers	1	Number of hidden layers in the MLP head, set to 1 for simplicity and efficiency.

Table 4: Final hyperparameters with justification

The performance comparison of various modeling approaches, including similarity-based (S) and classification-based (C) methods, as well as techniques incorporating background removal (BR) and ensemble strategies (E), is summarized in the accompanying figure. This visualization highlights the results on both public and private scores, providing a clear assessment of each method's effectiveness. For a detailed analysis of these results, please refer to the figure 3, which underscores the impact of

different strategies on overall model performance.

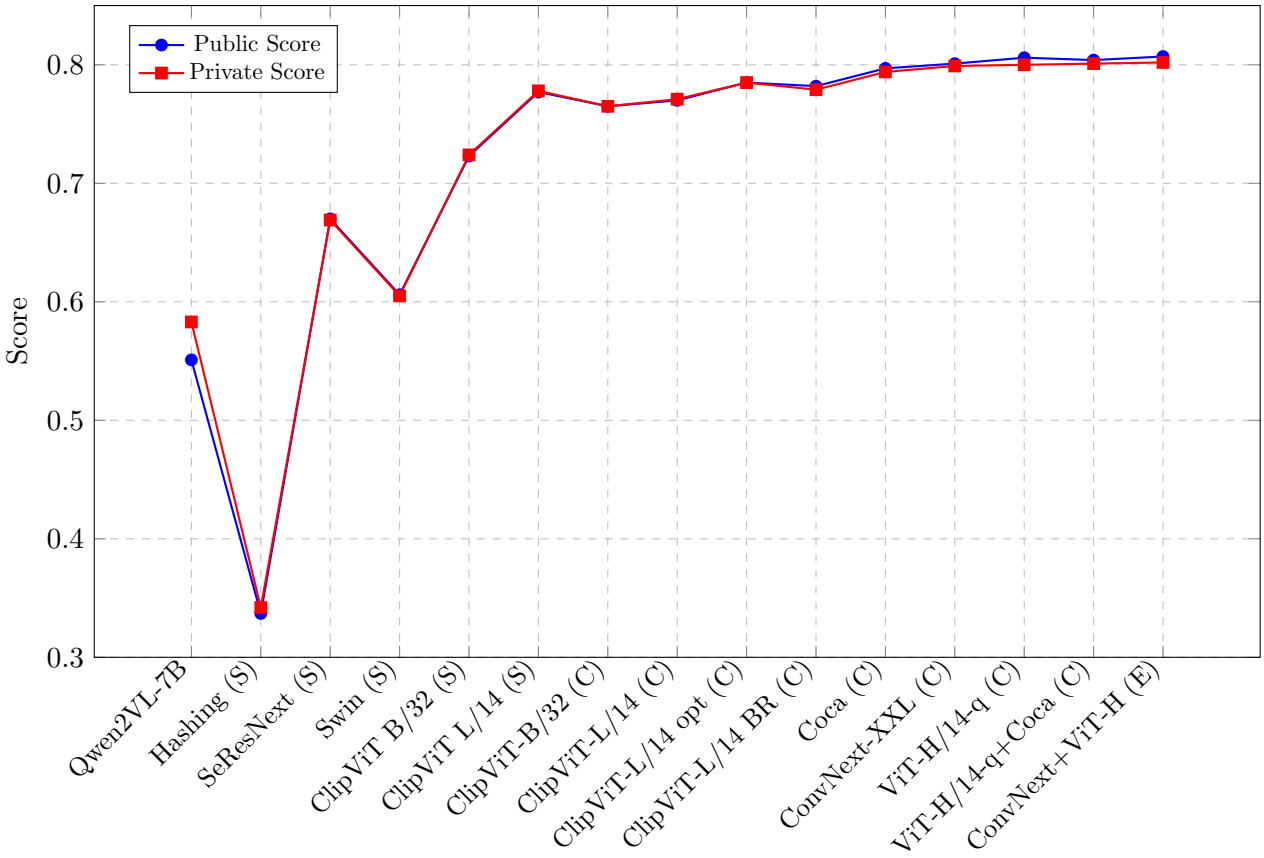

Figure 3: Comparison of model performances on public and private scores. Note: (S) indicates Similarity-based approach, (C) indicates Classification-based approach, BR stands for background removal and (E) indicates Ensemble.

6 Novelty and Innovation

6.1 Ensemble Design:

The methodology combines two advanced models: CLIP ViT-H/14-quickgelu and ConvNext-XXLarge into an ensemble framework. By unfreezing all layers of the CLIP model, it adapts fully to the specific dataset, enabling more accurate extraction of fine-grained features for predicting product attributes.

6.2 Category-Specific MLP Classifiers:

For each category-attribute combination, such as 'Sarees_color' or 'Men Tshirts_color', a separate multi-layer perceptron (MLP) head was trained. This approach ensures the model can handle imbalanced data effectively by isolating predictions for different attribute groups, reducing the impact of dominant classes.

6.3 Selective Loss Calculation:

During training, instances with null target labels are excluded from the loss computation. This prevents the model from learning irrelevant patterns and ensures the predictions remain accurate and focused on meaningful data.

7 Training Details

7.1 Hardware Configuration

- **Training Environment**: NVIDIA A100 GPU with 40GB memory
- **Inference Environment**: NVIDIA T4 GPU with 15GB memory

7.2 Training Optimizations

- **Mixed Precision Training**:
  - Enabled using PyTorch's autocast for CUDA.
  - Applied gradient scaling to prevent underflow during FP16 operations.
  - Maintained FP32 precision for critical computations.
- **Data Loading Optimizations:** Persistent workers and prefetching with a factor of 2 enabled efficient data pipelines.
- **Model Compilation:** PyTorch 2.0+ compilation was used for automatic optimization of computational graphs, improving training speed.

7.2.1 Learning Rate Scheduling

- Separate learning rates for the CLIP model and the MLP predictor components.
- MultiStepLR scheduler with milestones at epochs 4, 6, and 10.
- A gamma decay factor of 0.1 applied for systematic learning rate reduction.

The total training time for the method was approximately **4.5 hours**, achieved through the efficient use of NVIDIA A100 GPUs and optimizations like mixed precision training and streamlined data pipelines. The pipeline ensures stability and resource efficiency, with effective learning rate scheduling and **Weights & Biases** logging for comprehensive metric tracking. Regular checkpointing secures model states and configurations, enabling smooth recovery and reproducibility. These strategies together provide a reliable and efficient framework for accurate model training.

8 Evaluation Metrics

8.1 Chosen Metrics

The performance of our model is evaluated using accuracy as the primary metric. Accuracy is chosen as it provides a clear and interpretable measure of the model's overall classification performance, reflecting the proportion of correct predictions across all categories. The final leaderboard score was calculated using the following formula:

$$\text{attribute\_f1\_score} = \frac{2 \cdot (\text{Micro\_F1} \cdot \text{Macro\_F1})}{\text{Micro\_F1} + \text{Macro\_F1}}$$

$$\text{score} = \frac{1}{\text{no\_of\_categories}} \sum_{j=1}^{m} \left(\frac{\sum_{i=1}^{n} \text{attribute\_f1\_score}}{\text{no\_of\_attributes\_of\_category\_j}} \right) \tag{1}$$

For training the model, we employed **cross-entropy loss**. Cross-entropy loss is commonly used in classification tasks as it effectively penalizes incorrect classifications, with a stronger penalty for predictions that are farther from the true label. This loss function is well-suited for multi-class classification problems and ensures that the model is optimized towards minimizing misclassifications during training.

8.2 Results

For Attribute Level F1-scores, please refer Table 5

8.3 Error Analysis

8.3.1 Overall Performance Summary

The average F1-score across all categories is 0.928, with Men Tshirts performing the best (average micro F1: 0.971) and Sarees the worst (average micro F1: 0.776).

8.3.2 Category-wise Analysis

Sarees (Most Challenging Category) Sarees exhibit significant challenges, particularly in the color attribute, which shows the poorest performance with a micro F1 score of 0.589 and a macro F1 of 0.511. This discrepancy suggests class imbalance issues. Additionally, attributes like pallu details, border features, and print/pattern types also perform poorly with micro F1 scores around 0.7. The complexity of color patterns, cultural variations in naming conventions, and overlapping design features contribute to these difficulties.

Category	Attribute	Micro-F1-score	Macro-F1-score	Harmonic Mean
Kurtis	Color	0.948	0.910	0.948
	Fit Shape	0.946	0.941	0.946
	Length	0.928	0.925	0.928
	Occasion	0.973	0.897	0.973
	Ornamentation	0.962	0.953	0.962
	Pattern	0.972	0.972	0.972
	Print/Pattern Type	0.969	0.969	0.969
	Sleeve Length	0.982	0.966	0.982
	Sleeve Styling	0.996	0.972	0.996
Men Tshirts	Color	0.890	0.898	0.890
	Neck	0.998	0.998	0.998
	Pattern	0.995	0.995	0.995
	Print/Pattern Type	0.974	0.959	0.974
	Sleeve Length	0.999	0.995	0.999
Sarees	Blouse Pattern	0.781	0.641	0.781
	Border	0.722	0.679	0.722
	Border Width	0.915	0.869	0.915
	Color	0.589	0.511	0.589
	Occasion	0.735	0.567	0.735
	Ornamentation	0.914	0.678	0.914
	Pallu Details	0.681	0.570	0.681
	Pattern	0.865	0.570	0.865
	Print/Pattern Type	0.715	0.632	0.715
	Transparency	0.843	0.496	0.843
Women Tops & Tunics	Color	0.942	0.927	0.942
	Fit Shape	0.939	0.930	0.939
	Length	0.971	0.971	0.971
	Neck Collar	0.956	0.944	0.956
	Occasion	0.991	0.844	0.991
	Pattern	0.985	0.964	0.985
	Print/Pattern Type	0.956	0.901	0.956
	Sleeve Length	0.979	0.974	0.979
	Sleeve Styling	0.970	0.960	0.970
	Surface Styling	0.934	0.905	0.934
Women Tshirts	Color	0.951	0.943	0.951
	Fit Shape	0.962	0.908	0.962
	Length	0.964	0.947	0.964
	Pattern	0.991	0.966	0.991
	Print/Pattern Type	0.924	0.919	0.924
	Sleeve Length	0.981	0.935	0.981
	Sleeve Styling	0.993	0.967	0.993
	Surface Styling	0.982	0.880	0.982
Overall F1-Score		0.928		

Table 5: Category and Attribute-level F1 Scores on Validation Data (0.3% of Training Data)

Men Tshirts (Best Performing Category) Men Tshirts show excellent performance, with micro F1 scores exceeding 0.99 for sleeve length, neck, and pattern. The consistency between micro and macro F1 scores indicates a well-balanced dataset. The simplified attribute structure, more standardized design features, and limited variation in attributes likely contribute to the category's success.

Women's Categories (Kurtis, Tops & Tunics, Tshirts) Women's categories generally perform well in structural attributes such as sleeve length and fit, with strong results in pattern recognition. However, subjective attributes like occasion and style show slightly lower performance, reflecting potential ambiguities in classification.

8.3.3 Attribute-wise Analysis

The best-performing attributes, with micro F1 scores greater than 0.95, include sleeve styling, pattern recognition, sleeve length, and neck/collar features. Conversely, challenging attributes such as color classification in Sarees, blouse pattern, border features, and pallu details show micro F1 scores below 0.80, suggesting these areas require further refinement.

8.3.4 Micro vs Macro F1 Score Disparities

The most significant disparities are observed in Sarees, particularly in attributes like pattern (0.865 vs 0.570) and ornamentation (0.914 vs 0.678), highlighting class imbalance. Women's Tops & Tunics also show a noticeable disparity in the occasion attribute (0.991 vs 0.844). In contrast, Men Tshirts exhibit minimal disparities between micro and macro F1 scores, demonstrating a more balanced dataset.

9 Conclusion

9.1 Summary of Results

The experimental evaluation revealed the strengths and limitations of various approaches for product attribute prediction. Table 1 highlights the performance of all methods, with classification-based approaches demonstrating superior accuracy and scalability. Notable insights include:

- The **ViT-H/14-quickgelu** model achieved the best individual performance, with a public leaderboard score of 0.806.
- The final **ensemble** method (ViT-H/14-quickgelu + ConvNext-XXLarge) outperformed all other models, achieving the highest scores of 0.807 on the public leaderboard and 0.802 on the private leaderboard.
- Visual Question Answering (VQA)-based approaches showed potential for contextual reasoning but lacked scalability and robustness.
- Image similarity-based search using CLIP ViT models exhibited strong performance in retrieval tasks, but it struggled to match the robustness of classification-based approaches for complex attribute prediction.

- Fine-tuning MLP heads on top of pre-trained architectures proved effective for multi-class classification, with minimal overfitting and consistent results across validation datasets.

The ensemble methodology provided a balanced solution, leveraging the complementary strengths of the ViT and ConvNext architectures to address the diverse challenges of fashion attribute prediction.

9.2 Limitations and Future Work

9.2.1 Best Single Model: ViT-H/14-quickgelu

Limitations

- **Sensitivity to Noise:** While the model excels in fine-grained attribute extraction, its performance degrades with images containing poor lighting, occlusions, or cluttered backgrounds.
- **Computational Cost:** ViT-H/14-quickgelu is resource-intensive, requiring substantial computational resources for both training and inference.
- **Limited Robustness Across Categories:** Generalization is uneven across attribute categories, particularly for underrepresented or rare attributes.

Future Work

- **Robust Feature Learning:** Incorporate advanced data augmentation techniques or adversarial training to improve resilience to noisy or diverse image conditions.
- **Improved Fine-Tuning Strategies:** Investigate low-rank adaptation (LoRA) and other efficient fine-tuning techniques to optimize resource utilization.
- **Cross-Domain Generalization:** Pre-train the model on a broader dataset, including diverse and synthetic samples, to improve performance on unseen data.

9.2.2 Best Ensemble Model: ViT-H/14-quickgelu + ConvNext-XXLarge

Limitations

- **Increased Computational Complexity:** The ensemble approach requires more memory and inference time, posing challenges for real-time or resource-constrained environments.
- **Deployment Challenges:** Deploying two large models in production increases operational costs and requires careful optimization.
- **Redundancy in Feature Representations:** Some overlap in features from the two models may introduce unnecessary complexity without proportional performance gains.

Future Work

- **Optimized Ensemble Strategies:** Leverage knowledge distillation to condense the ensemble's combined knowledge into a single lightweight model for faster inference.
- **Dynamic Ensemble Weights:** Develop adaptive weighting mechanisms to optimize model contributions based on input characteristics during inference.
- **Model Compression:** Apply pruning or quantization techniques to reduce the memory and computational requirements of the ensemble.
- **Multi-Modal Learning:** Incorporate textual metadata or attribute descriptions to improve predictions through multi-modal fusion.

These improvements aim to enhance the practical applicability, scalability, and robustness of the proposed methodology in large-scale e-commerce environments.

References

[1] https://www.meesho.io/ai/data-challenge, [Accessed 18-11-2024].

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[4] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, "LAION-5b: An open large-scale dataset for training next generation image-text models," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. [Online]. Available: https://openreview.net/forum?id=M3Y74vmsMcY

[5] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," 2024. [Online]. Available: https://arxiv.org/abs/2409.12191

[6] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014. [Online]. Available: https://arxiv.org/abs/1408.2927

[7] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2019. [Online]. Available: https://arxiv.org/abs/1709.01507

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021. [Online]. Available: https://arxiv.org/abs/2103.14030

[9] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[10] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, "Coca: Contrastive captioners are image-text foundation models," 2022. [Online]. Available: https://arxiv.org/abs/2205.01917