Clothes Recognition Based on Lightweight Deep Learning Models

Yuchao Zhang, Minh Nguyen, Wei Qi Yan
Department of Computer and Information Sciences
Auckland University of Technology, New Zealand

ABSTRACT

This book chapter explores lightweight deep learning models for human clothes recognition in resource-constrained environments. Although most models achieve good classification performance, high computational complexity limits deployment on low-power devices. We compare three recent lightweight models—MNv4-Conv-S, YOLO11n, and YOLO12n—under identical conditions, with results showing classification accuracies between 82% and 86%, highlighting trade-offs in speed and resource usage. To better suit classification tasks, we propose an optimized variant of YOLO12n, named YOLO12n-LC, which removes the detection head and redundant modules. This not only reduces computational overhead but also improves classification accuracy to 90%. Compared to original YOLO12n, YOLO12n-LC has 2.1 million parameters, a 32% reduction, and its FLOPs decrease to 4.2G, with a significant reduction in inference latency. The redesigned model is much suitable for deployment on devices with limited resources. Our findings offer practical guidance for selecting and adapting models in applications such as intelligent retail, personalized recommendation, and industrial automation.

Keywords: Clothes recognition; Deep learning; Lightweight models; Attention mechanism; MobileNet; YOLO12n

INTRODUCTION

Human clothing recognition has become an important research topic in computer vision, with widespread applications in fashion design, automated retail, and smart manufacturing (Shin et al., 2023). As online shopping and digital production continue to grow, e-commerce platforms increasingly rely on intelligent systems to provide personalized recommendations, while manufacturers adopt automated systems to improve production management and inventory monitoring efficiency (Di, 2020; Liu et al., 2024). In these scenarios, it is essential to bridge the gap between image features and high-level semantics. Accurate and resource-efficient clothing recognition not only improves the user experience but also supports decision making in production, supply chain management, and various industries such as smart retail, fashion technology, and automated manufacturing.

The core function of a deep learning-based clothes recognition system is to accurately classify class categories, such as tops, pants, shoes, accessories, etc., by extracting and analyzing visual features from images (Yan, 2019; Yan, 2023, Yan, 2026). As fashion styles continue to evolve and consumers demand more personalized recommendations, the complexity of clothing recognition tasks also increases (Liu et al., 2018). These systems face a few challenges, including the wide diversity of clothing types, deformations caused by varying camera angles and poses, inconsistent lighting conditions, and cluttered backgrounds (Vijayaraj et al., 2022; Liu, 2018). This variability places higher demands on the generalizability and robustness of the model in real world scenarios (Zhou et al., 2022). Furthermore, as clothes recognition models grow in depth and complexity to improve accuracy, the computational and hardware requirements

for training and inference also become increasingly demanding, complicating deployment, particularly on low-power or edge devices.

In recent years, deep learning, especially Convolutional Neural Networks (CNNs)—has made remarkable strides in extracting complex patterns from images and improving classification performance (Liu, 2018; Fan et al., 2016; Abbas et al., 2024). The models like ResNet, MobileNet, and YOLO have achieved high accuracy in clothing recognition tasks (Donati et al., 2019). However, their computational complexity makes the deployment on low-power or resource-limited devices, such as smartphones, laptops, and edge devices, challenging (Zhou et al., 2022; Elleuch et al., 2019).

Deploying deep learning models in smart factories and on edge devices faces two main challenges: High cost and dependency on high-performance hardware or cloud APIs, as well as the need for efficient offline operation. Previous research often evaluates models under inconsistent conditions, typically using high-end hardware, which limits insights into deployment on resource-limited platforms. Furthermore, trade-offs between accuracy, inference time, memory usage, and energy consumption are often underexplored. This study addresses these issues by providing a unified evaluation of classic and modern lightweight models under consistent conditions, measuring their performance on both server and edge devices. We compare the latest models such as MNv4-Conv-S and YOLO12n with established models such as YOLO11n (Sapkota et al., 2024), ensuring standardized testing. Our work also takes considerations of real-world constraints such as offline inference, computational efficiency, and resource usage, offering a comprehensive reference for practical deployment.

Resource consumption is a critical factor in the deployment of deep learning models on resource-constrained devices, including memory usage, model size, parameter count, inference latency, and energy consumption. High-resolution inputs or deep architectures exacerbate memory demands, potentially preventing execution on low-end devices with limited RAM (e.g. <4 GB on smartphones). Computational complexity directly impacts inference latency and power consumption, which are particularly critical for mobile and embedded platforms, where battery life and real-time responsiveness are paramount. Hardware constraints, such as limited storage, CPU performance, and lack of GPU / NPU acceleration, further compound these challenges, requiring lightweight models that balance accuracy and efficiency (Gao, et al, 2024; Yang, et al. 2024). Techniques such as model pruning, quantization, and knowledge distillation are widely adopted to reduce computational load and memory requirements, enhancing deployability on edge devices (Liu et al., 2018).

This study focuses on lightweight models specifically designed for resource-constrained environments, including MobileNet and the YOLO series. These models are optimized to minimize parameter size and computational overhead while maintaining competitive accuracy. MobileNet, for example, reduces computation through separable convolutions in depth, making it well suited for low power applications (Shubathra et al., 2020). YOLO models are popular for real-time object detection and have also been applied to clothes recognition (Pan, Yan, 2018; Pan, Yan, 2020; Pan, et al, 2021), particularly where large volumes of image data must be processed efficiently (Liu et al., 2016). Despite of these advantages, the models still face challenges in recognizing clothes in diverse and complex conditions (Vijayaraj et al., 2022).

This book chapter aims to contribute to the deployment of deep learning models in environments with low computational resources (Kayed et al., 2020). We explore the performance of lightweight models, including MNv4-Conv-S, YOLO11n, and YOLO12n, while optimizing their feature extraction capabilities (Gu et al., 2023). Our goal is to minimize computational resource usage without compromising classification accuracy, allowing efficient operation on mobile and edge platforms (Kaur et al., 2023). The main contributions of this book chapter are as follows.

- We conduct a comprehensive comparison of the latest lightweight models (e.g. MNv4-Conv-S and YOLO12n) and classic models (e.g., YOLO11n) under identical training and testing environments. This combined evaluation reveals that while newer models show numerous advantages in various scenarios, they do not always outperform older models in general classification tasks.
- We propose a lightweight, classification-specific variant of YOLO12n, named YOLO12n-LC. Unlike the original YOLO12n, which is optimized for real-time object detection, YOLO12n-LC removes the detection head and neck modules while retaining the lightweight backbone. A global

average pooling layer and a fully connected classification head are added to adapt the classification model. This architectural change significantly reduces the parameters, inference latency, and computational complexity while preserving accuracy, making it suitable for mobile and edge deployment.

- We take advantage of transfer learning (Elleuch et al., 2021) to simplify training and accelerate adaptation to specific tasks by using minimal data and computation. This improves the applicability of the model in resource-constrained settings (Vijayaraj et al., 2022).
- We evaluated the performance of the model under challenging real-world conditions, including variations in lighting, angles, and occlusions.

This study is organized around two main objectives. First, we conduct a comparative analysis of representative lightweight classification models, including MobileNetV3-Small, EfficientNet-Lite0, and ResNet18, in addition to task-specific models such as MNv4-Conv-S, YOLO11n, and YOLO12n. Although lightweight general-purpose models like MobileNetV3 and EfficientNet exhibit favorable inference speed and energy efficiency, the classification accuracy in our clothing dataset is significantly lower compared to MNv4-Conv-S and variants based on YOLO. Therefore, we select MNv4-Conv-S, YOLO11n, and YOLO12n as the main comparison baselines due to their performance and practical relevance in resource-constrained scenarios. Second, we propose a simplified variant of YOLO12n, called YOLO12n-LC, which removes redundant detection components and focuses solely on classification. This redesign reduces model size and computation, making it better suited for single-label clothing classification on low-power devices.

Our findings emphasize the importance of selecting and customizing models based on deployment needs. The results provide actionable insights for the application of lightweight vision models in smart factories, e-commerce platforms, and industrial automation, where accuracy must be balanced with resource efficiency.

RELATED WORK

Human clothing recognition, as a significant application in computer vision, encompasses a broad range of fields, from online shopping recommendations to intelligent factory automation management (Shin et al., 2023; Di, 2020). In recent years, with the rapid development of e-Commerce, clothing recognition has provided consumers with accurate recommendations, greatly improving the online shopping experience (Liu et al., 2024; Vijayaraj et al., 2022). Moreover, with the proliferation of applications such as virtual try-on, fashion design recommendations (Cong et al., 2024), and industrial automation, human clothing recognition has played an essential role in intelligent manufacturing by automatically identifying clothing categories (Zhou et al., 2022).

However, clothing recognition has multiple challenges, such as image artifacts, the diversity of clothing categories, complex backgrounds, and pose variations. These factors require our models to have a strong generalizability to cope with various lighting conditions, angle changes, and occlusions (Liu, 2018; Eshwar et al., 2016). With the diversification of fashion styles, the development of accurate, fast, and scalable clothes recognition models has become an urgent research problem, especially achieving efficient and accurate inference on devices limited by resources (Shin et al., 2023). Before the rise of deep learning, the recognition of human clothing was mainly based on specified feature extraction methods and shallow deep learning models (Nodari et al., 2012).

Conventional methods such as Local Binary Patterns (LBP), Histograms of Oriented Gradients (HOG), and Support Vector Machines (SVM) performed well in specific scenarios, but lacked generalization capabilities, particularly when addressing challenges such as varying lighting conditions and angle changes (Xu et al., 2022). Furthermore, machine learning methods were mainly based on assigned features, making it challenging to automatically extract and learn more discriminative high-level features while dealing with large-scale data sets and complex scenarios (Nodari et al., 2012; Wang, 2023). Furthermore, the computational efficiency of traditional methods was low, particularly in real-time applications, making it difficult to meet the demands of dynamic changes. Therefore, with increasing data volume and the demand

for classification accuracy and speed, deep learning methods have gradually become mainstream in the field of clothing recognition (Eshwar et al., 2016; Xiang et al., 2024).

The emergence of deep learning models, especially the introduction of Convolutional Neural Networks (CNNs), has greatly pushed the research progress in the recognition of human clothes (Cychnerski et al., 2017). CNN models, by stacking multiple convolutional kernels, can automatically extract multiple levels of visual features from the given images, including edges, textures, and complex shapes (Liu et al., 2018; Wang et al., 2023). Classic CNN architectures such as AlexNet and VGG have achieved excellent performance (Liu et al., 2018; Xu et al., 2022). In large-scale clothing datasets such as DeepFashion and Fashion-MNIST, deep learning models have significantly improved classification accuracy and robustness. Although CNNs have greatly improved accuracy through automated feature extraction, computational complexity and model size have limited applications in resource-constrained devices. In particular, large models such as ResNet and Inception are difficult to perform real-time inference on mobile devices and edge computing devices due to the large number of parameters and high computational complexity (Vijayaraj et al., 2022; Liu et al., 2016). These models are employed in high-performance servers and cloud computing environments. Furthermore, as deep learning models advance, the computational cost of training and deploying these models becomes increasingly demanding.

Another significant challenge is privacy. With the growing importance of data privacy, ensuring effective inference while maintaining user data security has become a key focus. Traditional cloud computing often requires uploading user data to remote servers for processing, which introduces risks during transmission and storage. In contrast, on-device inference processes data locally, significantly reducing the risk of data leakage (Donati et al., 2019). This approach is particularly beneficial in personal scenarios, such as virtual try-on and personalized clothing recommendations, where privacy protection is crucial (Liu et al., 2024). By minimizing data transfers and keeping sensitive information on the device, edge computing aligns with stringent data protection regulations and enhances user trust, making it a preferred solution for privacy-sensitive applications.

To overcome these limitations and meet the demands of low-power and resource-constrained devices, the market requires lightweight models capable of independent operation, reducing computational costs and memory usage while maintaining relatively high accuracy. MobileNet and SqueezeNet are typical examples of lightweight models (Shubathra et al., 2020). MobileNet, through separable convolutions in depth, significantly reduces the number of parameters and computations, making it suitable for applications on mobile and edge devices (Gu et al., 2023). MNv4-Conv-S further optimized inference speed and memory usage by introducing linear bottlenecks and residual connections, thus significantly reducing computational overhead while maintaining high accuracy (Gu et al., 2023). In addition, Babuc et al. proposed the Clothing-DAT model, which is based on MobileNet and optimized for clothing recognition on IoT devices (Shubathra et al., 2020). This model has been successfully deployed on smartphones and low-power devices, significantly improving inference speed and reducing computational costs. MNv4-Conv-S is a versatile model for the mobile ecosystem, featuring enhanced Squeeze-and-Excitation (SE) mechanisms and an optimized architecture. It achieves superior classification performance with lower FLOPs and memory usage, making it ideal for edge and IoT devices. Its balance of accuracy and efficiency makes it suitable for real-world applications (Qin et al., 2025). The application of these lightweight models has not only expanded the application scenarios of deep learning methods in resource-constrained devices but has also provided more possibilities for future edge computing (Islam et al., 2024).

YOLO series of models, as lightweight object detection models, have been widely utilized in clothes recognition. In particular, versions such as YOLO11n and YOLO12n have achieved real-time inference capabilities through streamlined network architectures (Liu et al., 2016; Lyu et al., 2024). Although YOLO11n demonstrated strong performance in real-time processing scenarios, YOLO12n introduced additional architectural improvements, including enhanced attention mechanisms and feature fusion modules, resulting in better accuracy and robustness for complex classification tasks. The existing studies have shown that YOLO models perform well in handling multicategory classification tasks, especially in scenarios requiring the efficient processing of large volumes of image data (Shubathra et al., 2020).

In recent years, attention mechanisms have been introduced to improve the performance of deep learning models by allowing the network to focus on relevant informative regions in images, particularly in scenarios involving complex backgrounds and pose variations (Gu et al., 2023; Shubathra et al., 2020). The Squeeze-and-Excitation (SE) module is a widely adopted example that enhances the representation of features by recalibrating channel-wise feature responses through global average pooling and fully connected layers (Gu et al., 2023). For example, Gu et al. incorporated SE into YOLOv4-Tiny and demonstrated improved generalization when classifying complex clothing images (Shubathra et al., 2020). Similarly, Lyu et al. applied attention mechanisms to enhance the robustness of recognition in scenes with occlusions and pose variation (Lyu et al., 2024). However, despite the improved accuracy, attention modules also introduce additional computational and memory overhead. For resource-constrained environments, such as mobile devices or edge devices, this complexity can undermine the advantages of lightweight architectures, reducing inference speed and increasing energy consumption. Therefore, it is crucial to strike a balance between accuracy and efficiency, particularly when designing compact models intended for practical deployment.

To address these trade-offs, recent studies have turned to cluster-based visual learners and general-purpose lightweight architectures as alternative pathways to achieve robust and efficient visual understanding. For example, Wang et al. (Wang et al., 2022) proposed a deep-nearest centroids approach to allow class-wise discrimination with reduced supervision. Liang et al. extended this idea to the segmentation domain with ClustSeg (Liang et al., 2023), and further developed ClusterFormer (Liang et al., 2023), which integrates clustering into transformer attention mechanisms for universal visual learning. These approaches demonstrate that incorporating clustering principles can significantly improve generalization with minimal overhead.

Simutanously, advances in lightweight transformer-based models have further broadened the design space for efficient image understanding. LightViT (Huang et al., 2022) introduces a convolution-free architecture that achieves a strong balance between accuracy and model size. DINOv2 (Oquab et al., 2023) explores self-supervised clustering to learn transferable representations without relying on labeled data, while Wen et al. (Wen et al., 2023) leverage prototypical clustering to improve short-handed classification performance. These efforts inspire the design of compact yet high-performing classifiers suited for real-world deployment, especially in scenarios where both generalization and computational efficiency are required.

Moreover, deploying lightweight models on resource-constrained devices presents a few challenges. Firstly, lightweight models, while reducing computational costs and model size, often underperform in complex scenarios, such as those involving diverse backgrounds, occlusions, and varied clothes classes, highlighting the need for improved robustness and generalization capabilities. Secondly, hardware limitations, including limited memory, computational power, and GPU availability, frequently result in slower inference speeds and reduced accuracy, particularly in real-time applications where responsiveness is critical. Third, device diversity and compatibility issues complicate deployment, requiring models to be adaptable across various hardware configurations. Moreover, integrating lightweight models into existing devices presents additional hurdles. Security and privacy further compound the challenges, as on-device inference must ensure robust data protection and comply with regulations such as GDPR. Finally, inference efficiency and response time are critical, as limited computational resources on edge devices directly affect performance. Addressing these challenges requires balancing resource consumption and model performance by optimizing model structures for fast and efficient inference, adopting hardware-aware strategies, and ensuring reliable deployment across diverse real-world scenarios.

To address these challenges, this study evaluates and optimizes lightweight models for clothing recognition, with a particular focus on their suitability for deployment on resource-constrained platforms. We perform a standardized comparison of three representative models: MNv4-Conv-S, YOLO11n, and YOLO12n. Although all three demonstrate promising classification accuracy around 84%, trade-offs between inference speed, model complexity, and resource consumption persist. Based on this analysis, we introduce a classification-specific variant of YOLO12n, named YOLO12n-LC. Unlike the original YOLO12n, which is designed for real-time object detection, YOLO12n-LC removes the detection head and

related components, streamlining the architecture for single-label classification tasks. This simplification significantly reduces computational demands and memory usage, making the model more suitable for mobile and embedded deployment. In our data set, YOLO12n-LC achieves a classification accuracy of 90% with notably lower computational overhead, outperforming baseline models and showing strong potential for real-world deployment in resource-limited environments.

METHODOLOGY

Figure 1. Class Distribution of Clothing Dataset

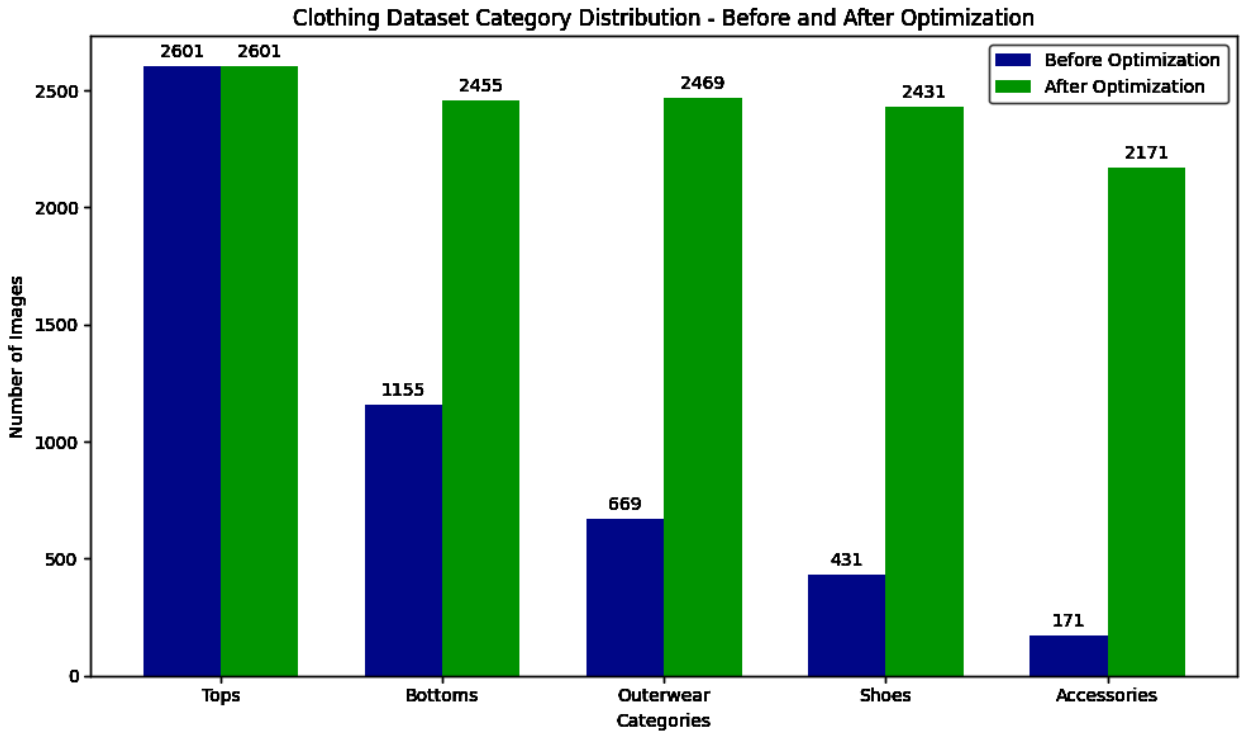


Figure 2. Class Distribution of Clothing Dataset– Before and After Optimization

Dataset Selection and Description

In this study, we used the Kaggle Clothes dataset, which initially contains more than 5,000 labeled images in 20 clothing classes. To streamline the classification task and reduce complexity, we merged semantically related subcategories and discarded those with insufficient samples or limited relevance to our objectives. As a result, the dataset was reorganized into five major classes: Tops, bottoms, shoes, outerwear, and

accessories. Each image has a resolution of 400 × 533 pixels and reflects real-world scenarios with varied lighting, backgrounds, and poses.

Despite this consolidation, class imbalance remained prominent, with the number of images in the largest class (tops) being more than three times that of the smallest class (accessories), as shown in Figure 2. Hence, we adopted a multistage data balancing strategy. First, we applied conventional augmentation techniques, random rotation, horizontal flipping, scaling, and color jittering, focused on underrepresented classes. Second, we introduced oversampling by duplicating augmented samples to ensure a more uniform distribution. Lastly, to further increase the diversity of the data and improve generalization, we incorporated additional samples from external public datasets, followed by consistent preprocessing and augmentation.

This hybrid approach not only enriched the visual diversity of minority classes, but also enhanced the overall representativeness of the data set. Consequently, it improved the robustness and fairness of model evaluation, particularly in addressing performance degradation typically observed in imbalanced classification tasks.

Model Selection

This study selected three recent lightweight models: MNv4-Conv-S, YOLO11n, and YOLO12n, and an improved variant, YOLO12n-LC, with the following characteristics.

Table 1. Model Parameter Comparison

Models	**Number of Parameters (M)**	**FLOPs (G)**
MNv4-Conv-S	3.8	0.3
YOLO11n	2.9	10.4
YOLO12n	2.8	6.5
YOLO12n-LC	**2.1**	**4.2**

MNv4-Conv-S is a lightweight model in the MobileNetV4 series, designed for resource-constrained mobile devices. It incorporates Squeeze-and-Excitation (SE) modules, hybrid convolutions, and Universal Inverted Bottleneck (UIB) modules, combined with an optimized Neural Architecture Search (NAS) strategy and Multi-Query Attention (MQA) module. These enhancements significantly improve feature extraction and classification accuracy. In the ImageNet-1K dataset, MNv4-Conv-S achieves a precision of 73.8% Top-1 with only 3.8M parameters and 0.2G MACs, which requires just 2.4 milliseconds of inference time on a Pixel 6 CPU. Demonstrating near-Pareto-optimal efficiency across CPUs, GPUs, and EdgeTPUs, this model is ideal for real-time image processing and object recognition tasks, particularly in privacy-preserving offline deployment scenarios. However, its computational demands are slightly higher than those of MNv4-Conv-S, necessitating careful optimization in resource-limited environments (Qin et al., 2025).

The YOLO family (You Only Look Once) is widely recognized for its real-time object detection capability that combines high accuracy and fast inference through a unified architecture. Each YOLO version typically includes a lightweight variant (e.g., YOLOv3-tiny, YOLOv4-tiny) that is specifically optimized for hardware efficiency. These “tiny” versions significantly reduce the number of parameters and floating-point operations (FLOPs), enabling real-time inference on edge devices and mobile devices. As the YOLO series evolved, newer models not only improved detection accuracy but also introduced architectural optimizations aimed at reducing memory usage and increasing inference speed. This trend continues with the development of YOLO11n and YOLO12n, which are tailored for low-resource environments without compromising performance.

YOLO11n is a lightweight variant introduced in the YOLO11 family, designed to meet the growing demand for efficient real-time inference on resource-limited devices. With approximately 2.9 million parameters, YOLO11n incorporates architectural improvements such as enhanced feature fusion and basic

attention mechanisms. These modifications help the model capture discriminative features while maintaining low computational complexity. YOLO11n performs well in multiclass image classification tasks, particularly in industrial and retail scenarios where balancing accuracy and speed is essential. Its design reflects a careful trade-off between performance and efficiency, making it an effective baseline for compact computer vision applications (Sapkota et al., 2024).

YOLO12n builds on the strengths of YOLO11n and introduces further enhancements in both the backbone and the overall architecture. It integrates lightweight transformer-inspired modules and advanced channel-wise processing, further improving feature extraction and representational power. Although the parameter count remains modest at approximately 2.8 million and FLOPs are limited to approximately 6.5G, YOLO12n achieves higher classification accuracy (up to 86%) and better F1 scores in multiple test sets. This makes YOLO12n not only more robust in handling challenging image variations but also more stable across training runs. Its performance demonstrates that high accuracy and low resource usage are not mutually exclusive, positioning YOLO12n as an ideal choice for deployment in embedded, mobile, and smart factory environments.

Evaluation Metrics

To assess the classification performance and the efficiency of the model deployment, we take use of standard evaluation metrics, including Precision (P), Recall (R), F1 Score, mean average precision (mAP), inference latency, and energy consumption. These metrics provide comprehensive insights into both predictive accuracy and practical deployment performance on resource-constrained devices.

- **Precision (P)**: Measures the proportion of true positives among all predicted positives.

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

- **Recall (R)**: Measures the proportion of true positives among all actual positives.

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

- **F1 Score**: The harmonic mean of precision and recall.

$$F1 = \frac{2\times TP}{2\times TP+FP+FN} \tag{3}$$

- **mAP (Mean Average Precision)**: For multi-class classification, mAP is calculated by averaging the AP (area under the precision–recall curve) across all classes:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{4}$$

- **Inference Latency**: Measured as the average time (in milliseconds) taken to process a single image, this indicates the responsiveness of the model on the target devices.

- **Energy Consumption**: Estimated as the average power usage during inference multiplied by the inference time per sample:

$$Energy = Power\ (W) \times Latency\ (s) \tag{5}$$

Pertaining to on-device testing, power usage is obtained via external hardware monitors or software tools. *TP*, *FP* and *FN* denote true positives, false positives and false negatives, respectively. AP_i is the average precision for class *i*, and *n* is the number of classes. These metrics ensure a balanced evaluation across accuracy, speed, and efficiency, especially critical for real-world deployment on edge devices.

Loss Function and Optimization Algorithm

To ensure effective learning during classification, all models in this study utilize the cross-entropy loss function, a widely adopted metric for multiclass classification. It is defined as:

$$H\left(y^{(i)}, \widehat{y^{(i)}}\right) = -\sum_{j=1}^{q} y_j^{(i)} \log \widehat{y_j^{(i)}} \tag{6}$$

where $y^{(i)}$ is the true one-hot encoded label of the i-th sample, $\hat{y}^{(i)}$ is the predicted probability distribution, and q denotes the number of target classes. This loss measures the divergence between the predicted and actual distributions, guiding the model to improve classification accuracy. To optimize the loss and accelerate convergence, we adopt the Adam optimizer, an adaptive learning algorithm that combines the advantages of momentum and RMSProp. Its key benefits include stable convergence and dynamic adjustment of learning rates between parameters. The update rule is as follows:

$$\theta = \theta - \eta \cdot \frac{m^t}{\sqrt{v^t} + \varepsilon} \tag{7}$$

where θ represents the model weights, η is the learning rate, m^t and v^t are the first and second moment estimates of the gradients, and ε prevents division by zero. This strategy improves stability and performance, especially in large-scale image datasets.

Activation Function and Network Design

All models adopt the Rectified Linear Unit (ReLU) activation function in their hidden layers:

$$ReLU(x) = max(0, x) \tag{8}$$

ReLU introduces non-linearity while avoiding the vanishing gradient problem common in deeper networks by using sigmoid or tanh functions. By preserving positive gradients and zeroing out negatives, ReLU accelerates training and facilitates deeper architectures. This is especially beneficial for image-based tasks such as clothes recognition, which require capturing complex features. For the output layer, we make use of the Softmax activation function to generate a normalized probability distribution over all classes:

$$Softmax(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{q} \exp(z_j)} \tag{9}$$

This enables the network to interpret the output as class probabilities, which is essential for assigning clothing categories such as Tops, Bottoms, Shoes, etc.

In summary, the combination of cross-entropy loss, Adam optimizer, ReLU activation in hidden layers, and Softmax output provides a solid, efficient, and interpretable foundation for training deep learning models in clothing classification tasks. These design choices contribute to the balance between accuracy, training efficiency, and deployment feasibility in resource-constrained devices.

Data Preprocessing and Augmentation

To improve model generalization and performance, data preprocessing and augmentation were performed on the data set. Before feeding the data into the model, all image pixels were normalized to the interval [0,1] to ensure a consistent distribution across all input data, facilitating faster convergence and preventing negative impacts caused by different data ranges.

To improve the generalization of the model, especially for minority classes, three data enhancement methods were applied. Random horizontal flipping was applied to simulate angle changes that occur during real-world image capture, allowing the model to maintain good classification capability under different image angles. Random rotations were applied to recognize clothes that are inclined or varies in pose, thus improving robustness to multi-pose images of clothes. Random scaling was used to simulate variations. This allows the model to extract key features from images of different sizes, increasing adaptability to various sizes of clothes.

The introduction of data augmentation not only increased the diversity of training data, but also effectively mitigated the overfitting problem. By continually generating variations of images, the model can learn more patterns, thereby improving its generalization capability.

In this book chapter, class imbalance was a significant challenge, particularly for minority classes. To address this, we applied an oversampling approach to balance the minority classes, primarily targeting

clothes classes with fewer images (e.g. Accessories and Outerwear). By replicating these images, they were given sufficient representation in the training set. In addition to oversampling, data augmentation further expanded the minority-class samples by generating variations in angle, size, and rotation. The combination of these methods ensured that the model learned more representative features of minority classes, effectively mitigating the negative effects of class imbalance.
Class weight adjustment was used in the loss function, assigning higher weights to minority classes to ensure accurate classification in the presence of imbalanced classes.

Pruning and Quantization

Pruning and quantization are two essential techniques for optimizing deep learning models for deployment on resource-constrained platforms, such as edge devices and mobile terminals. Pruning aims to reduce model size and inference latency by removing redundant components, such as unimportant weights, filters, or entire channels, from the network. This not only decreases the number of parameters but also simplifies the computational graph, accelerating inference without severely compromising accuracy. Quantization, on the contrary, compresses the model by converting high-precision floating point operations (e.g., FP32) into lower-precision formats (e.g., INT8), significantly reducing memory bandwidth requirements and improving execution efficiency on hardware accelerators that support integer operations.

These techniques provide a clear direction to further improve the feasibility of the deployment of YOLO12n. Although YOLO12n already exhibits a strong balance between accuracy and efficiency, its architecture, originally designed for object detection, still includes components that are unnecessary for single-label classification tasks.

Our optimized variant, YOLO12n-LC, simplifies the original structure by removing the detection head and neck, retaining only the lightweight backbone, and introducing a classification head. Although this modification already reduces computational complexity, applying structured pruning (e.g., channel or layer-wise pruning) can further eliminate redundant computations, particularly in early convolutional stages. Furthermore, integrating post-training quantization or quantization-aware training (QAT) would allow YOLO12n-LC to operate with INT8 precision, thereby lowering memory consumption and improving inference speed on real-time hardware such as ARM-based processors or NPUs.

These techniques are particularly valuable for applications requiring fast low power classification, such as intelligent manufacturing, wearable devices, or mobile recommendation systems. In future work, we plan to explore hardware-aware pruning strategies and integer quantization pipelines to further compress the YOLO12n-LC model. The goal is to deliver a high-performance, low-latency classification model that retains accuracy while being highly suitable for deployment in real-world, resource-limited environments.

EXPERIMENTS

In this section, we present the experimental setup, including data pre-processing, training procedures, and evaluation metrics. The primary goal is to assess the effectiveness of three lightweight deep learning models: MNv4-Conv-S, YOLO11n, and YOLO12n for human clothing classification in resource-constrained environments, such as mobile and embedded devices. We compare the models in terms of classification accuracy, inference time, and computational efficiency to identify the most suitable architecture for deployment.

To further improve performance under limited resources, we propose an optimized version of YOLO12n, named YOLO12n-LC, by removing unnecessary detection components and tailoring the architecture for single-label classification. We conducted comparative experiments between YOLO12n-LC and the original models, demonstrating that YOLO12n-LC not only reduces computational overhead but also improves classification accuracy, achieving better suitability for real-world applications.

Experimental Environment

Our experiments were carried out in the following hardware and software environment, shown in Table 2 and Table 3. In terms of software configuration, we took advantage of the latest version of the PyTorch deep learning framework, which supports GPU acceleration, running on Ubuntu 20.04 LTS. PyTorch's dynamic computational graph support allows flexible model and hyperparameter adjustments to meet various experimental needs.

Table 2. Experimental Hardware Configuration

Hardware Component	**Specification**
CPU	Intel(R) Xeon(R) @ 2.00 GHz, 4 cores, 2 threads, AVX-512 support
Memory	32 GB RAM
GPU	2 × Tesla T4 (15,360 MiB each), CUDA 12.4

Table 3. Experimental Software Configuration

Software Component	**Version / Description**
Operating System	Ubuntu 20.04 LTS
Deep Learning Framework	PyTorch 2.0
Python Version	Python 3.9

Although the experimental hardware configuration is well defined, the presence or absence of GPU support has a significant impact on training and inference efficiency. On devices with GPU support, both model training speed and inference time are significantly shortened. This difference is particularly unsuitable for YOLO, where inference time is greatly extended in the absence of GPU support, making it unsuitable for real-time applications.

The parallel computing capability of GPUs allows efficient parameter updates in large batch training, reducing overall training time. For example, MNv4-Conv-S completes a 100-epoch training session in just 1 hour with GPU support; without GPU, the training time can exceed 10 hours. Similarly, YOLO performs better in the GPU environment, with significantly improved inference efficiency, making it well suited for scenarios that require efficient processing of large-scale data.

Data Processing and Augmentation

Kaggle Clothing Dataset was utilized in this book chapter. To enhance the generalizability of the model in complex clothing scenarios, a series of data enhancement techniques were used, including random horizontal flipping, brightness adjustment, and random cropping. These enhancements artificially introduce variations in training images, allowing the model to learn a wider range of image features, thereby exhibiting stronger robustness when facing clothing images under different poses, angles, and lighting conditions.

Model Selection and Parameter Setting

In our experiments, three lightweight models were selected for comparative analysis: MNv4-Conv-S, YOLO11n, and YOLO12n. The first round of training was conducted under consistent conditions to evaluate the baseline performance of each model in terms of classification accuracy, inference time, and computational efficiency.

Based on the results, YOLO12n demonstrated the best overall performance and was selected for further optimization. To better adapt it to single-label clothing classification tasks, we proposed a simplified variant named YOLO12n-LC. Unlike the original YOLO12n, which was designed for real-time object detection, YOLO12n-LC removes the detection head and related modules, retaining only the backbone and adding a

classification head. This redesign significantly reduces computational complexity and model size, making it more suitable for deployment on edge devices and mobile devices.

The improved YOLO12n-LC achieved the highest classification accuracy (90%) among all models tested, while maintaining low resource consumption. This shows its practical value for lightweight applications where efficient and accurate classification is required.

In this experiment, we utilized the Adam optimizer due to its adaptive learning rate mechanism, which ensures stable convergence and efficient training. The initial learning rate was established at 0.001, and an exponential decay with a factor of 0.9 was applied every 10 epochs to enhance generalization and reduce the risk of overfitting. A batch size of 32 was applied to balance computational efficiency and memory consumption. Each model was trained for 100 epochs, with validation loss continuously monitored to dynamically adjust the learning rate and guide the training process.

All experiments were conducted in consistent settings to ensure fair comparison between the models. Our goal was to evaluate and compare the performance of MNv4-Conv-S, YOLO11n, YOLO12n, and YOLO12n-LC under identical conditions in terms of convergence speed, classification accuracy, and computational cost.

To ensure the reliability and statistical significance of the experimental results, we conducted four independent experiments for each model, recording key metrics such as accuracy, inference time, and F1 scores during each run. We calculated the average values of these metrics and adopted 95% confidence interval to assess the stability and consistency of the experimental results.

The study revealed that data augmentation significantly helped mitigate the issue of class imbalance. Multiple experiments comparing different augmentation strategies showed that combining multiple enhancement strategies led to significant improvements in overall accuracy and F1 scores, particularly reducing misclassification for minority classes such as accessories and outerwear.

Experimental Results and Analysis

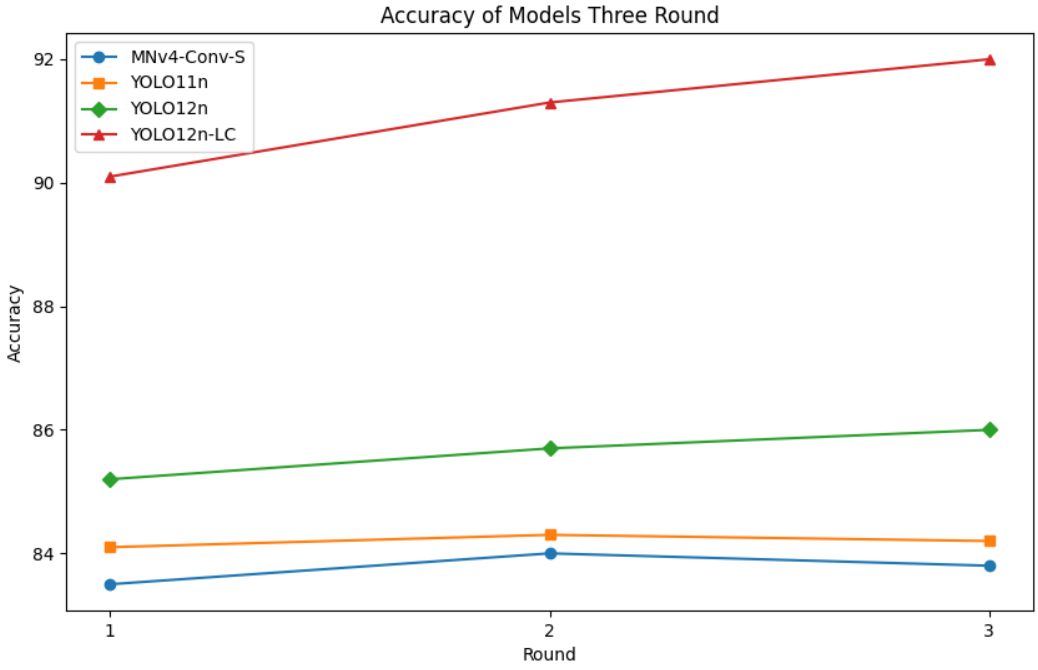


Figure 3. Three-round Accuracy After 100 Epochs

Fig. 3 illustrates the accuracy trends of four models—MNv4-Conv-S, YOLO11n, YOLO12n, and the proposed YOLO12n-LC—over three rounds of training. The three baseline models exhibit relatively close and limited improvements: MNv4-Conv-S increases slightly from 83.1% to 84.0%, YOLO11n from 84.1% to 84.5%, and YOLO12n from 85.2% to 86.0%, indicating a limited optimization potential without further architectural modifications. In contrast, YOLO12n-LC, a YOLO12n-specific optimized classification variant, consistently outperforms all other models, with an accuracy ranging from 90.1% to 92.0%. By removing the detection head and redundant components, YOLO12n-LC focuses exclusively on single-label classification, leading to better utilization of training data, improved generalization and training efficiency. The widening performance gap across rounds highlights the effectiveness of its task-aligned architectural simplification under resource-constrained conditions.

These results indicate that although baseline lightweight models perform reasonably well in clothes classification tasks, task-specific structural simplification—such as removing redundant components based on classification requirements, as carried out in the redesigned YOLO12n-LC—can significantly enhance overall performance. This task-aligned optimization not only improves classification accuracy and generalization but also improves the practicality of the model for real-world applications, particularly on mobile and edge devices with limited computational resources.

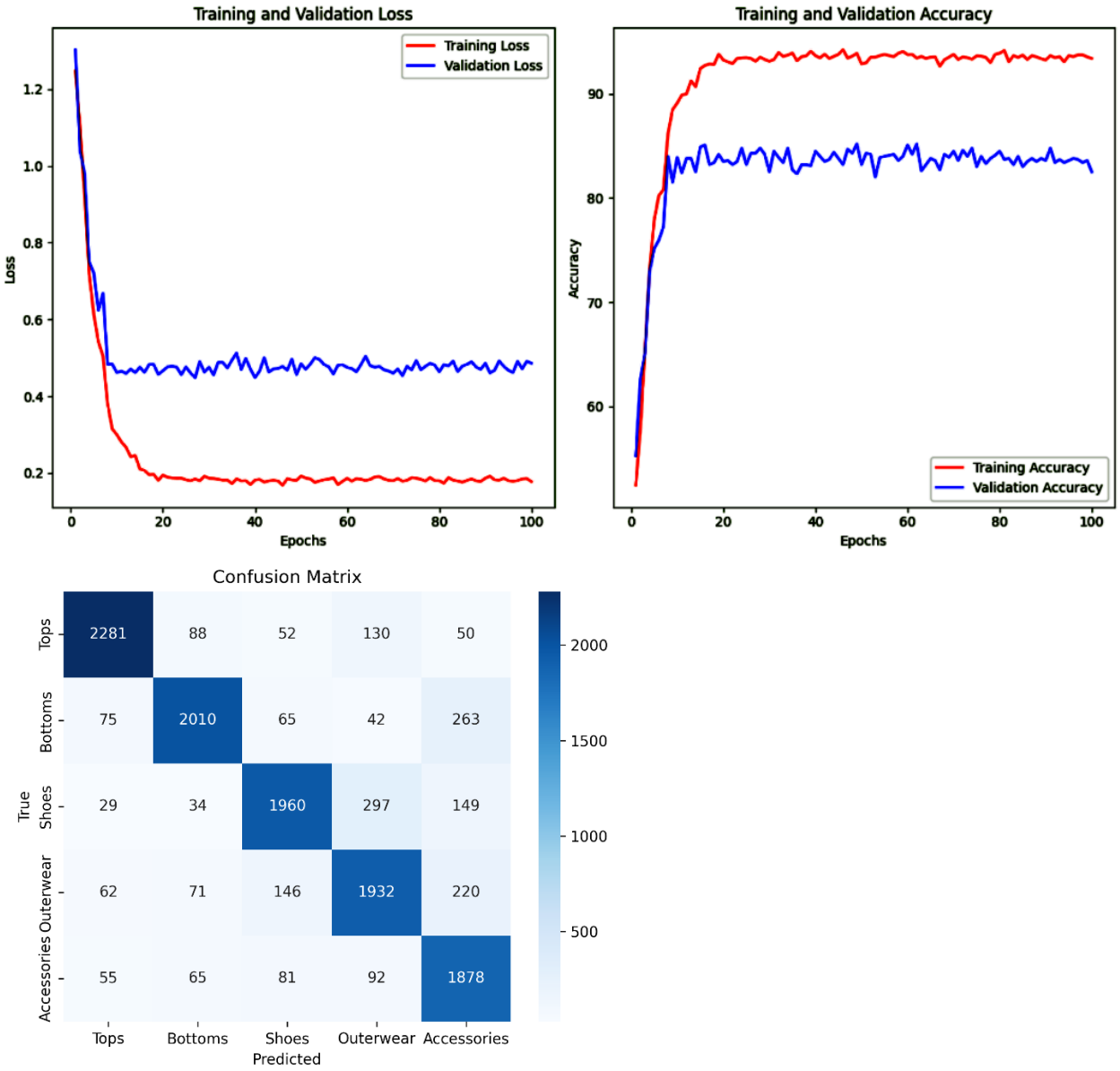


Figure 4. Performance of MNv4-Conv-S: Accuracy Curve and Confusion Matrix after Training for 100 Epochs

Table 4. Per-class Precision, Recall, and F1-score for MNv4-Conv-S

Class	**Precision**	**Recall**	**F1-score**	**Support**
Tops	0.89	0.88	0.88	2601
Bottoms	0.89	0.82	0.85	2455
Shoes	0.85	0.79	0.82	2469
Outerwears	0.79	0.79	0.79	2431
Accessories	0.76	0.87	0.81	2171
Average	**0.84**	**0.83**	**0.83**	**12127**

Fig.4 presents the performance of the MNv4-Conv-S model after 100 training epochs, including training and validation loss curves, accuracy curves, and the confusion matrix. As shown in the loss and accuracy plots, the model converges rapidly within the first 10 epochs and maintains stable performance thereafter. Although training accuracy continues to increase and eventually stabilizes, validation accuracy plateaus at approximately 83%, indicating a potential generalization gap. This discrepancy suggests that while the model learns effectively from the training set, its ability to generalize to unseen data remains limited.

MNv4-Conv-S demonstrates high classification accuracy for dominant categories such as Tops and Bottoms, with 2,281 and 2,010 correct predictions, respectively. However, it struggles with more visually ambiguous classes, such as Outerwear, where 130 samples are misclassified as Tops and 146 as Shoes. Similarly, Shoes are often confused with Outerwear (297 cases) and Accessories (149 cases), indicating that the model's feature extraction capability may be insufficient to capture subtle distinctions between classes. These misclassifications underscore the challenges of discriminating between categories with overlapping visual traits.

In general, these results reflect both the strengths and limitations of MNv4-Conv-S. Its lightweight architecture and rapid convergence make it a promising candidate for deployment in resource-constrained environments where inference speed and computational efficiency are critical. However, its limited representational capacity, particularly for capturing high-level semantic differences, affects its performance on visually similar categories. Compared to more advanced architectures, MNv4-Conv-S offers a reasonable trade-off between efficiency and accuracy, but further improvements in feature representation may be necessary to improve performance on complex classification tasks.

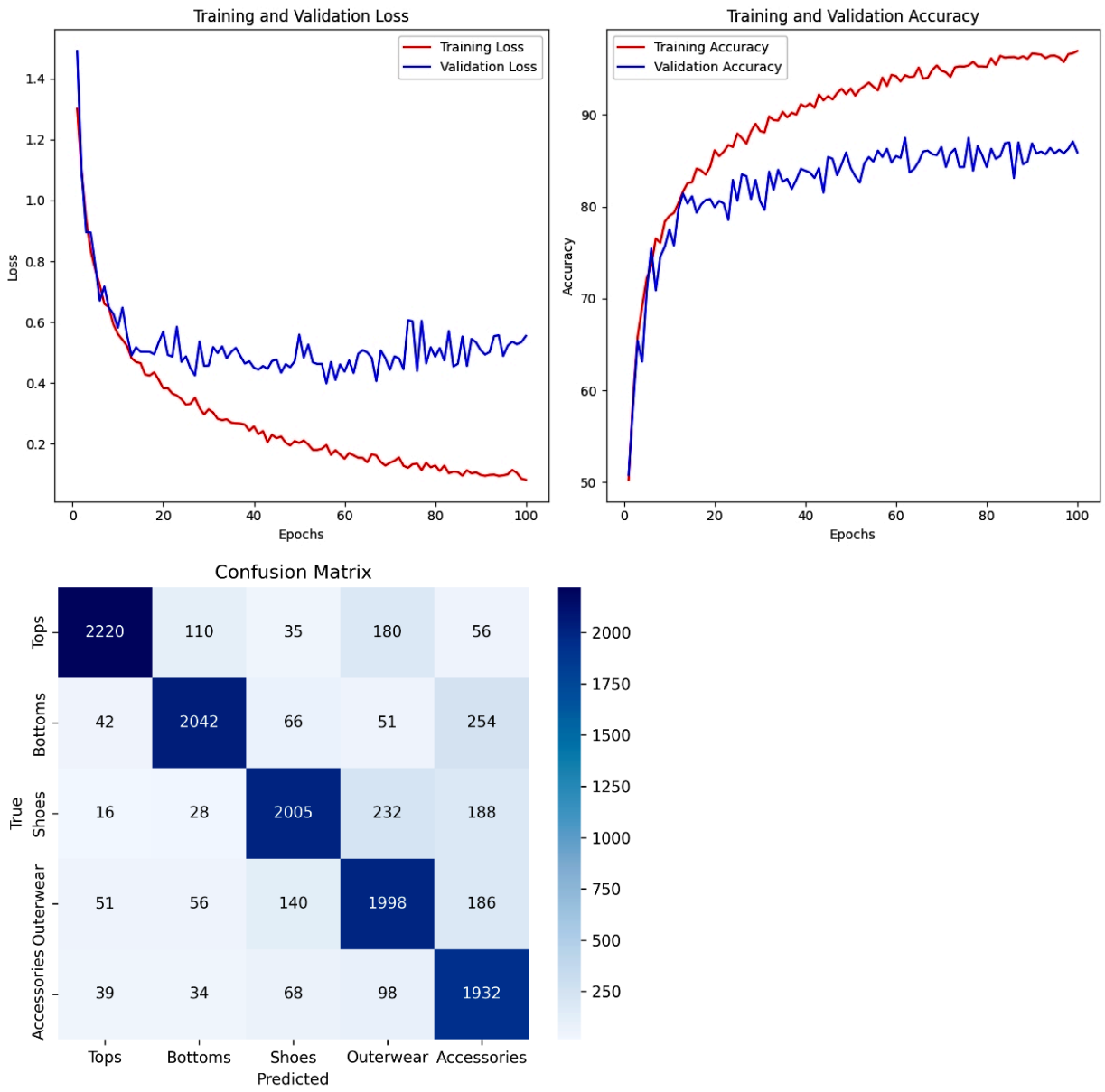


Figure 5. Performance of YOLO11n: Accuracy Curve and Confusion Matrix After Training for 100 Epochs

Table 5. Per-class Precision, Recall, and F1-score for YOLO11n

Class	**Precision**	**Recall**	**F1-score**	**Support**
Tops	0.91	0.85	0.88	2601
Bottoms	0.89	0.83	0.86	2455
Shoes	0.87	0.81	0.84	2469
Outerwears	0.77	0.82	0.79	2431
Accessories	0.75	0.89	0.81	2171
Average	**0.84**	**0.84**	**0.84**	**12127**

Fig.5 presents the performance of the YOLO11n model after 100 training epochs, showing the training and validation loss curves, the accuracy curves, and the confusion matrix. The training loss steadily decreases over time, while the validation loss plateaus with noticeable fluctuations after approximately 30

epochs, suggesting that the model may begin to overfit the training data despite its relatively compact architecture. The training accuracy continues to improve and reaches around 90%, while the validation accuracy stabilizes at approximately 84%, indicating a potential generalization gap between the training and validation sets.

The confusion matrix further illustrates the performance of the model in different clothing categories. The model correctly classifies 2,220 samples from the Tops category and 2,042 samples from the Bottoms category, reflecting strong recognition performance for dominant classes. However, Outerwears present notable confusion: 232 samples are misclassified as Shoes and 186 as Accessories, while 104 are confused with Tops and Bottoms. Similarly, 188 Shoes are misclassified as Accessories and 188 Accessories as Outerwears, revealing difficulties in distinguishing visually similar or functionally adjacent categories.
Compared to MNv4-Conv-S, YOLO11n maintains a slightly higher average accuracy but exhibits a greater degree of misclassification for mid- and low-frequency classes. These findings suggest that though YOLO11n benefits from a compact design and efficient training, its ability to extract fine-grained features for ambiguous categories may require further refinement, particularly for deployment in scenarios demanding high inter-class discrimination.

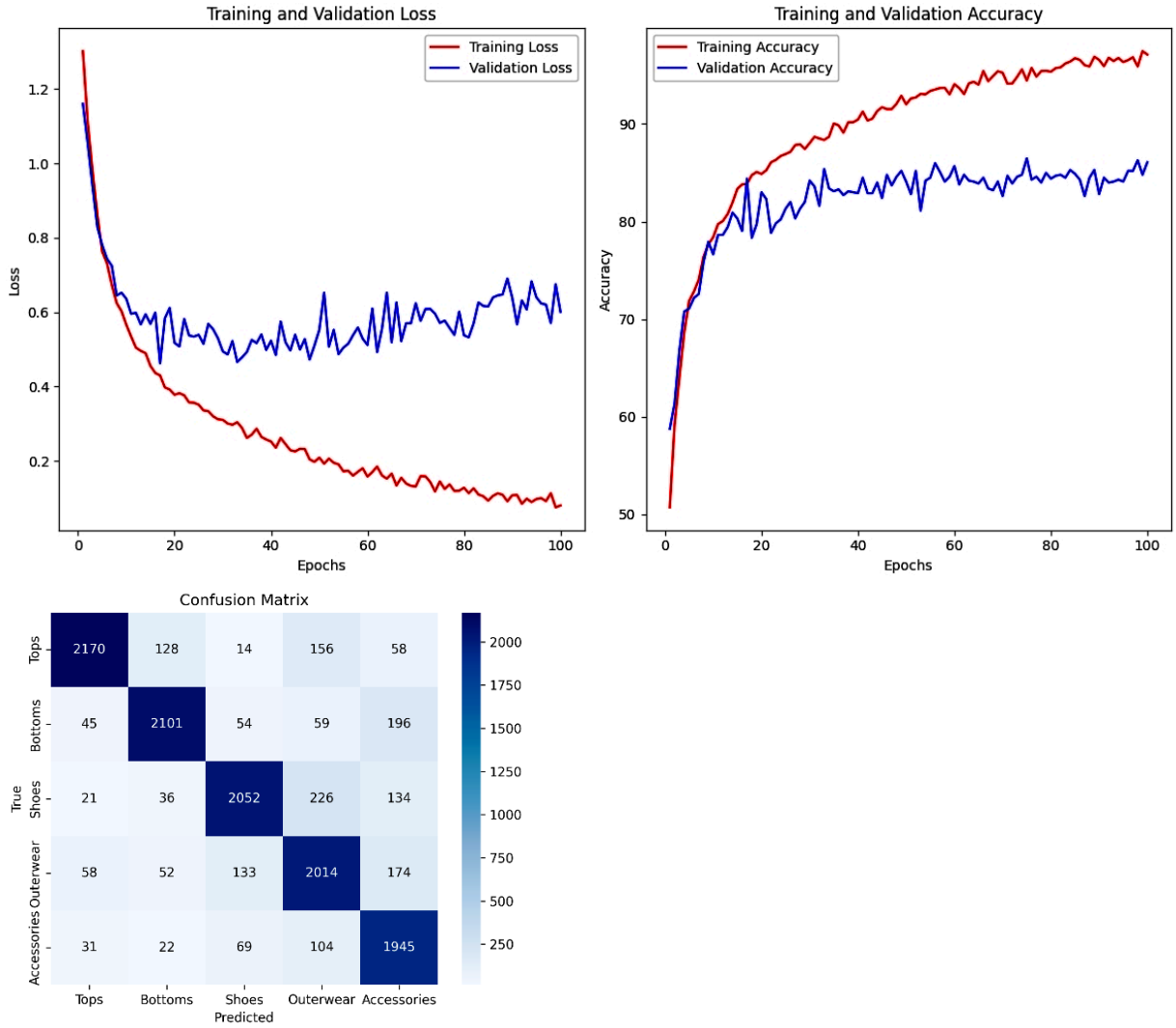


Figure 6. Performance of YOLO12n: Accuracy Curve and Confusion Matrix after Training for 100 Epochs

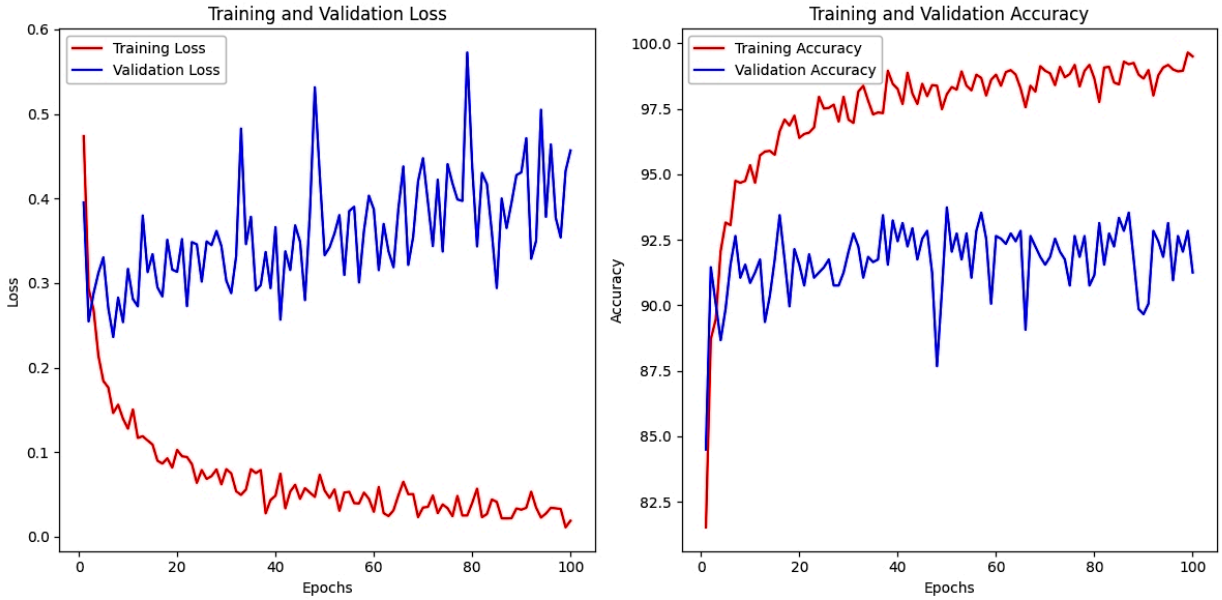
Table 6. Per-class Precision, Recall, and F1-score for YOLO12n

Class	Precision	Recall	F1-score	Support
Tops	0.92	0.86	0.89	2526
Bottoms	0.90	0.86	0.88	2455
Shoes	0.88	0.83	0.86	2469
Outerwears	0.81	0.83	0.82	2431
Accessories	0.80	0.90	0.85	2171
Average	**0.86**	**0.86**	**0.86**	**12052**

Fig.6 illustrates the training performance and classification results of YOLO12n after 100 epochs. The training and validation loss curves demonstrate smooth convergence, and the validation accuracy stabilizes around 86%, indicating minimal overfitting and strong generalization.

The confusion matrix shows that YOLO12n achieves high true positive counts across all categories, with particularly strong performance in dominant classes such as Tops (2,170 correct) and Bottoms (2,101 correct). Compared to MNv4-Conv-S and YOLO11n, YOLO12n significantly reduces misclassifications in challenging categories such as Outerwear and Accessories; for example, Accessories are correctly classified 1,945 times, with fewer confusions into Outerwear or Shoes.

These results confirm that YOLO12n not only improves classification accuracy and robustness across all classes, but also handles inter-class ambiguity better than earlier variants. Its balance of efficiency and accuracy makes it a more effective choice for multiclass clothing classification on resource-constrained devices.



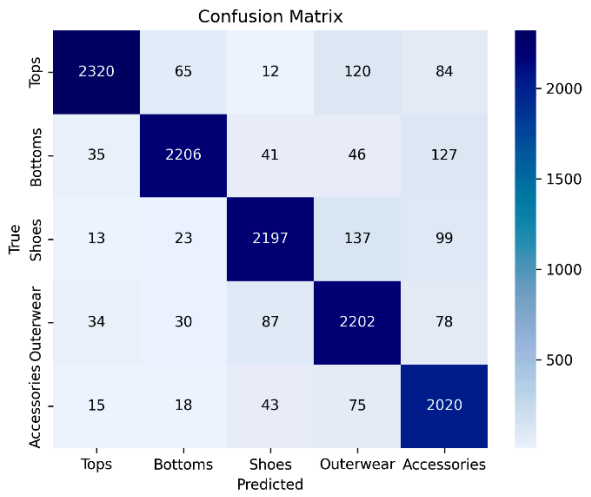


Figure 7. Performance of YOLO12n-LC: Accuracy Curve and Confusion Matrix After Training for 100 Epochs

Table 7. Per-class Precision, Recall, and F1-score for YOLO12n-LC

Class	**Precision**	**Recall**	**F1-score**	**Support**
Tops	0.94	0.89	0.91	2601
Bottoms	0.92	0.85	0.88	2502
Shoes	0.90	0.89	0.90	2469
Outerwears	0.86	0.89	0.87	2473
Accessories	0.86	0.92	0.89	2171
Average	**0.90**	**0.89**	**0.89**	**12216**

Fig.7 illustrates the training dynamics and classification results of the optimized YOLO12n-LC model after 100 epochs. The training and validation curves demonstrate rapid convergence, with the validation accuracy consistently stabilizing above 92%, indicating improved generalization. Compared to the original YOLO12n, the optimized version achieves higher accuracy and lower validation loss, suggesting that the removal of redundant detection components improves adaptation to single-label classification tasks. The confusion matrix further confirms the effectiveness of the design, showing clearer class boundaries and significantly improved predictions for previously challenging categories such as Outerwear and Accessories. These results highlight the suitability of YOLO12n-LC for deployment in resource-constrained environments, where high accuracy and low computational overhead are essential.

We present a qualitative comparison of the prediction results on five sample garment images using four different models. Each row corresponds to a model, and each column represents the same test image across models. The predicted labels are shown at the top of each image, with the correct predictions highlighted in green and the incorrect predictions in red. The ground truth labels for the five images, from left to right, are Accessories, Accessories, Accessories, Shoes, and Shoes.

The MNv4-Conv-S and YOLO11n models consistently misclassify the accessory items, often confusing them with visually similar classes such as Bottoms or Outerwears, and fail to correctly classify any of the five images. YOLO12n achieves moderate success, correctly identifying the last two shoe items, but continues to struggle with ambiguous classes such as Accessories and Tops. In contrast, YOLO12n-LC shows the best performance, correctly predicting 4 out of 5 samples and significantly improving the

classification of Accessories, a category that typically suffers from low precision due to overlapping visual features with other garments.

This qualitative analysis aligns with the quantitative metrics, further supporting the conclusion that YOLO12n-LC exhibits superior generalization and robustness. Its consistent accuracy in visually challenging classes, such as Accessories and Shoes, highlights the effectiveness of the lightweight refinement and class clustering strategy.

To further analyze the inference performance and accuracy of the models, we compared MNv4-Conv-S, YOLO11n, YOLO12n, and YOLO12n-LC on both server-side devices (with GPU) and local devices (with only CPU, Raspberry Pi 5). The specific configuration is shown in **Table 8**. As shown in **Table 9** and **Table 10**, YOLO12n-LC outperformed MNv4-Conv-S, YOLO11n, and YOLO12n in both inference time and precision, regardless of whether a GPU was available or the models were deployed on edge devices.

Figure 8. Visual Comparison of Predictions Made by Different Models on Five Sample Garment Images

Table 8. Raspberry Pi 5 Local Device Configuration

Configuration Item	Description
Processor (CPU)	ARM Cortex-A76 Quad-Core, 2.0GHz
Memory (RAM)	8GB LPDDR4
Storage	64GB MicroSD Card
Operating System (OS)	Raspberry Pi OS 64-bit

Table 9. Model Comparison on Server Testing Dataset

Model Name	Accuracy	Avg Time (s/img)	F1	mAP	Throughput (img/s)	Energy (J/img)	Memory (MB)	CPU
MNv4-Conv-S	83.33%	0.04	0.83	0.81	25.00	3.20	3680	19.90%
YOLO11n	83.92%	0.34	0.84	0.85	2.94	27.20	3725	20.10%
YOLO12n	85.47%	0.30	0.86	0.89	3.33	24.00	3756	20.80%
YOLO12n-LC	**90.33%**	**0.16**	**0.92**	**0.92**	**6.25**	**12.80**	**3580**	**15.70%**

Table 10. Model Comparison on Raspberry Pi 5 Testing Dataset

Model Name	Accuracy	Avg Time (s/img)	F1	mAP	Throughput (img/s)	Energy (J/img)	Memory (MB)	CPU
MNv4-Conv-S	84.58%	0.12	0.85	0.82	8.33	0.60	2688	65.57%
YOLO11n	84.96%	1.93	0.85	0.86	0.52	9.65	2746	82.33%
YOLO12n	86.76%	2.21	0.87	0.89	0.45	11.05	2763	86.19%
YOLO12n-LC	**91.76%**	**1.25**	**0.91**	**0.93**	**0.80**	**6.25**	**2463**	**64.36%**

Table 9 and Table 10 summarize the performance of four lightweight models—MNv4-Conv-S, YOLO11n, YOLO12n, and YOLO12n-LC—on both the server platform and the Raspberry Pi 5 platform. On the server side, YOLO12n-LC achieves the highest classification accuracy (90.33%) and F1 score (0.92), while maintaining a moderate inference time (0.16 s/img) and the lowest CPU usage (15.70%). YOLO12n also performs well, reaching 85.47% accuracy and an F1 score of 0.86, though with slightly higher latency (0.30 s/img) and memory usage (3,756 MB). YOLO11n achieves a precision comparable to that of MNv4-Conv-S (83.92% vs. 83.33%) but suffers from a much longer inference time (0.34 s/img), suggesting lower efficiency. MNv4-Conv-S stands out with the fastest inference time (0.04 s/img) and the lowest resource consumption, at the cost of slightly reduced accuracy and F1 score.

On the Raspberry Pi 5, YOLO12n-LC again demonstrates the best performance, with the highest accuracy (91.76%) and F1 score (0.91), while also consuming the least memory (2,463 MB) and

maintaining lower CPU usage (64.36%) than other YOLO variants. YOLO12n delivers competitive accuracy (86.76%) but has the longest inference time (2.21 s/img) and the highest CPU usage (86.19%), indicating significant computational overhead. YOLO11n shows similar drawbacks, with an inference time of 1.93 s/img and CPU usage of 82.33%, while only achieving 84.96% accuracy. MNv4-Conv-S maintains fast inference (0.12 s/img), lower CPU usage (65.57%), and decent accuracy (84.58%), making it the most efficient baseline model for real-time edge deployment.

These results confirm that YOLO12n-LC offers the best trade-off between accuracy and resource efficiency across platforms, while MNv4-Conv-S remains the preferred choice for applications requiring ultra-low latency and limited hardware resources.

The observed performance differences are mainly attributed to architectural design and task-specific optimization. YOLO12n-LC removes the detection head and redundant layers from YOLO12n, replacing them with a lightweight classification head. This modification reduces computational overhead while improving classification accuracy for single-label tasks. In contrast, YOLO11n and YOLO12n retain detection-oriented modules, which introduce unnecessary complexity and slow down inference, particularly on CPU-bound edge devices. MNv4-Conv-S, built on depthwise separable convolutions and enhanced with Squeeze-and-Excitation (SE) attention modules, balances speed and accuracy. Its streamlined architecture enables faster execution and lower memory usage, explaining its efficiency in resource-constrained environments. However, it provides relatively limited feature extraction capability compared to YOLO12n-LC, resulting in slightly lower classification accuracy.

In addition, we assess the impact of transfer learning and Squeeze-and-Excitation (SE) attention modules. The results show that transfer learning significantly improves training efficiency and classification accuracy. Models initialized with pre-trained weights achieve approximately 10% higher accuracy than those trained from scratch. In addition, pre-trained models converge more quickly, with validation accuracy stabilizing within 20–30 epochs, compared to 50–60 epochs for non-pre-trained models. The SE attention mechanism further improves classification performance by improving the model's focus on informative features. When combined with pre-trained backbones, SE modules contribute an additional gain in accuracy of 2–4.5%, particularly in identifying visually similar or minority classes. However, the most substantial improvement in this study results from architectural simplification and task alignment in YOLO12n-LC, enabling efficient and accurate single-label classification on resource-limited devices.

To evaluate deployment suitability on low-power devices, we estimate energy consumption per inference using power profiling tools. As shown in Table 9 and Table 10, YOLO12n-LC consumes only 6.25 joules per inference on Raspberry Pi 5, which is significantly lower than other YOLO-based models (e.g., 11.05 J for YOLO12n and 9.65 J for YOLO11n). This further validates its suitability for energy-constrained embedded applications.

In summary, the experimental findings demonstrate that YOLO12n-LC achieves the most favorable balance between precision and efficiency among all evaluated models. By simplifying the original YOLO12n architecture—removing detection heads and optimizing for single-label classification—YOLO12n-LC consistently delivers the highest classification accuracy (up to 91.76%) and F1 score (0.91), while minimizing memory and CPU usage in both server and edge environments.

Although MNv4-Conv-S exhibits slightly lower accuracy, it provides exceptional speed and low resource usage, making it highly suitable for real-time applications on constrained platforms. In contrast, YOLO11n, despite its lightweight design, does not surpass MNv4-Conv-S in accuracy and incurs significantly higher inference latency, reducing its practical utility. YOLO12n improves classification performance but is hindered by higher computational demands. These results highlight the importance of task-specific design, transfer learning, and hardware-aware optimization, as exemplified by YOLO12n-LC, for robust and efficient clothing classification in real-world scenarios.

Ablation Study

To better evaluate the impact of the proposed architectural modifications in YOLO12n-LC, we performed an ablation study isolating the effects of its two main design elements:

- Replacing standard convolutional layers with LightConv blocks for improved computational efficiency
- Introducing a lightweight classification head tailored for single-label prediction tasks

Each variant was trained on the same dataset and evaluated on Raspberry Pi 5 under identical testing conditions.

Table 11. Ablation Study Results of YOLO12n-LC

Model Variant	Accuracy	F1	mAP	Time (s/img)	Energy (J/img)
YOLO12n (baseline)	86.76%	0.87	0.89	2.21	11.05
+ LightConv only	88.10%	0.89	0.91	1.65	8.25
+ Lightweight Head only	88.76%	0.90	0.91	1.54	7.40
YOLO12n-LC (full)	**91.76%**	**0.91**	**0.93**	**1.25**	**6.25**

As shown in Table 11, both LightConv and Lightweight Head contributed to improving classification performance while reducing inference cost. When combined in YOLO12n-LC, these components achieved the best overall trade-off, confirming their complementary benefits and suitability for efficient edge deployment.

DISCUSSION

This study presents a comprehensive comparison and technical dissection of four lightweight neural network architectures: MNv4-Conv-S, YOLO11n, YOLO12n, and the proposed YOLO12n-LC are applied to the classification of the image of multiclass clothing.

MNv4-Conv-S is based on the MobileNetV4 backbone and employs depthwise separable convolutions to significantly reduce the parameter count and computational overhead. It integrates the Squeeze-and-Excitation (SE) attention module (Shubathra et al., 2020; Lyu et al., 2024) to re-calibrate channel-wise features, enhancing the model's focus on informative signals. Empirical results show that MNv4-Conv-S achieved the fastest inference time on Raspberry Pi 5 (0.12 sec/img), with low CPU usage (65.57%) and moderate classification accuracy (84.58%). However, due to limited semantic depth, it showed frequent misclassifications between visually similar categories such as outerwear and tops in the confusion matrix, indicating insufficient feature expressiveness for fine-grained discrimination.

YOLO11n: Lightweight in size, but not in efficiency. YOLO11n utilizes a simplified CSPDarkNet backbone and retains anchor-based detection heads and three-scale output. Although effective in object detection, such components are redundant in single-label classification. The model achieved an accuracy of 84.96%, comparable to MNv4-Conv-S, but with significantly higher inference latency (1.93 s/img) and CPU usage (82.33%). This discrepancy demonstrates that a lightweight model in terms of parameters does not guarantee runtime efficiency when redundant paths are retained. The confusion matrix analysis also revealed weak differentiation between Accessories and Shoes, underscoring its suboptimal adaptation to classification tasks.

YOLO12n: YOLO12n introduces transformer-inspired attention modules, SPPF (spatial pyramid pooling), and multiscale feature aggregation (PANet). These components enrich spatial semantics and context modeling in detection tasks. In the classification setting, YOLO12n achieved improved accuracy (86.76%) and an F1 score (0.87), outperforming YOLO11n and MNv4-Conv-S. However, its inference time reached 2.21 s/img and memory usage increased to 2763 MB, highlighting the computational overhead caused by detection-specific components. These results indicate that although its semantic capacity is stronger, its structural complexity limits deployment feasibility on non-GPU edge platforms.

YOLO12n-LC: YOLO12n-LC is a task-specific variant of YOLO12n. It removes the detection head, regression modules, and multiscale fusion layers, retaining only the backbone and attention modules. A lightweight classification head composed of global average pooling (GAP) and a fully connected layer is included. This redesign dramatically improves deployment metrics. The model achieves the highest

classification accuracy (91.76%) and the F1 score (0.91) on the Raspberry Pi 5, with the inference time reduced to 0.87 s/img and the memory usage lowered to 2463 MB. The confusion matrix further confirms its improved recognition of outerwear and accessories, demonstrating the benefit of removing detection-oriented redundancy and focusing on class-level abstraction.

To better understand the effectiveness of the YOLO12n-LC architecture, we conducted an ablation study focusing on its two key structural innovations: replacement of standard convolutions with LightConv blocks and the introduction of a lightweight classification head.As shown in Table 11, each modification led to consistent improvements in precision, mAP, inference latency, and energy efficiency. Their combination in YOLO12n-LC achieved the most balanced performance, demonstrating that architectural simplification tailored to classification tasks is highly effective for edge deployment.

Although this experiment primarily highlights structural improvements, we also briefly explored the effects of transfer learning and SE attention modules during training. Both contributed positively: Transfer learning accelerated convergence and improved generalization, while SE attention improved recognition of fine-grained classes. Due to space constraints, detailed results on these factors are omitted, but confirmed their complementary value to structural optimization.

CONCLUSION

The experimental results validate that YOLO12n-LC achieves the best overall performance on all the platforms tested. Its specific architectural simplification for classification, replacing detection heads with a lightweight classification module, effectively reduces redundancy and improves both accuracy (91.76%) and efficiency on edge devices. Although MNv4-Conv-S offers the fastest inference and the lowest resource usage, its limited expressiveness of features leads to lower precision in visually similar categories. In contrast, YOLO11n and YOLO12n incur higher computational costs due to their detection-oriented structures, highlighting the importance of aligning model architecture with classification tasks. Although techniques such as transfer learning and SE attention modules contribute to performance, their improvements are secondary compared to the structural advantages of YOLO12n-LC.

Despite the promising results, we still have challenges remaining for future work (Zhang, Nguyen, Yan, 2025; Zhang, 2026). First, further optimization for extremely low-power devices is critical. Although MNv4-Conv-S performs well, techniques such as model pruning, quantization, and light attention can further reduce computational load while maintaining performance, enabling deployment on power-constrained platforms (Liu et al., 2024). Second, addressing class imbalance remains essential. Although data augmentation helped partially, minority class performance still lags behind. Future work will explore lightweight GAN-based data generation to augment underrepresented categories and enhance model generalization, while keeping inference feasible on edge devices. Third, real-world deployment and iterative model adaptation will be central to the next phase. We plan to integrate the proposed model into real-world scenarios such as on-line retail and smart factory terminals (Vijayaraj et al., 2022), leveraging continuously labeled production data for incremental fine-tuning and performance stabilization over time.

This study confirms the value of lightweight models in constrained environments and provides practical insight into balancing accuracy and efficiency. YOLO12n-LC demonstrates a compelling trade-off between accuracy, inference speed, and resource usage, which makes it well suited for deployment in smart retail, personalized recommendation systems, and edge-based manufacturing management.

By refining model design, training strategies, and deployment pipelines, future work will further advance the real-world readiness of lightweight clothing classification systems in both industrial and consumer applications.

REFERENCES

Abbas, W., Zhang, Z., Asim, M., Chen, J., & Ahmad, S. (2024). AI-driven precision clothing classification: Revolutionizing online fashion retailing with hybrid two-objective learning. Information, 15(4), 196.

Cao, X. and Yan, W. (2022) Pose estimation for swimmers in video surveillance. Multimedia Tools and Applications, Springer.

Chen, Z., Yan, W. (2024) Real-time pose recognition for billiard player using deep learning. Deep Learning, Reinforcement Learning and the Rise of Intelligent Systems, pp.188-208, Chapter 10, IGI Global.

Cong, X., & Zhang, W. (2024). The application of hierarchical perception technology based on deep learning in 3D fashion design. Heliyon, 10(9), e29983. https://doi.org/10.1016/j.heliyon.2024.e29983

Cychnerski, J., Brzeski, A., Boguszewski, A., Marmolowski, M., & Trojanowicz, M. (2017). Clothes detection and classification using convolutional neural networks. In Proceedings of the International Conference on Emerging Technologies and Factory Automation (pp. 1–8). IEEE. https://doi.org/10.1109/ETFA.2017.8247638

Di, W. (2020). A comparative research on clothing images classification based on neural network models. In Proceedings of the IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT) (pp. 495–499).

Donati, L., Iotti, E., Mordonini, G., & Prati, A. (2019). Fashion product classification through deep learning and computer vision. Applied Sciences, 9(7), 1385.

Elleuch, M., Mezghani, A., Khemakhem, M., & Kherallah, M. (2019). Clothing classification using deep CNN architecture based on transfer learning. In International Conference on Hybrid Intelligent Systems (pp. 240–248). Springer.

Eshwar, S. G., Rishikesh, A. V., Charan, N. A., & Umadevi, V. (2016). Apparel classification using convolutional neural networks. In Proceedings of the International Conference on ICT in Business Industry & Government (pp. 1–5). IEEE. https://doi.org/10.1109/ICTBIG.2016.7892641

Fan, W., Zhao, Q., Liu, Q., & Yin, B. (2016). Attribute-based approach for clothing recognition. In T. Tan et al. (Eds.), Pattern Recognition (pp. 364–378). Springer. https://doi.org/10.1007/978-981-10-3005-5_30

Gao, X., Nguyen, M., Yan, W. (2024) HFM-YOLO: A novel lightweight and high-speed object detection model. Optimization, Machine Learning, and Fuzzy Logic: Theory, Algorithms, and Applications (Chapter 16). IGI Global.

Gu, M., Hua, W., & Liu, J. (2023). Clothing attribute recognition algorithm based on improved YOLOv4-Tiny. Signal, Image and Video Processing, 17(7), 3555–3563. https://doi.org/10.1007/s11760-023-02580-5

Huang, T., Huang, L., You, S., Wang, F., Qian, C., & Xu, C. (2022). LightViT: Towards lightweight convolution-free vision transformers. arXiv preprint arXiv:2207.05557. https://arxiv.org/abs/2207.05557

Islam, T., Miron, A., Liu, X., & Li, Y. (2024). Deep learning in virtual try-on: A comprehensive survey. IEEE Access, 12, 29475–29502.

Kaur, N., & Pandey, S. (2023). Predicting clothing attributes with CNN and SURF-based classification model. Multimedia Tools and Applications, 82(7), 10681–10701. https://doi.org/10.1007/s11042-022-13714-1

Kayed, M., Anter, A., & Mohamed, H. (2020). Classification of garments from Fashion-MNIST dataset using CNN LeNet-5 architecture. In Proceedings of the International Conference

on Innovative Trends in Communication and Computer Engineering (ITCE) (pp. 238–243). IEEE.
Liang, J., Cui, Y., Wang, Q., Geng, T., Wang, W., & Liu, D. (2023). ClusterFormer: Clustering as a universal visual learner. Advances in Neural Information Processing Systems, 36, 64029–64042.
Liang, J., Zhou, T., Liu, D., & Wang, W. (2023). ClustSeg: Clustering for universal segmentation. arXiv preprint arXiv:2305.02187. https://arxiv.org/abs/2305.02187
Liu, R., Joseph, A. A., Xin, M., Zang, H., Wang, W., & Zhang, S. (2024). Personalized clothing prediction algorithm based on multi-modal feature fusion. International Journal of Engineering & Technology Innovation, 14(2).
Liu, Y., Zhao, M., Zhang, Z., Liu, Y., & Yan, S. (2024). Arbitrary virtual try-on network: Characteristics preservation and trade-off between body and clothing. ACM Transactions on Multimedia Computing, Communications and Applications, 20(5), 1–23.
Liu, Z., Zhao, Q., Liu, Q., & Yin, B. (2016). DeepFashion: Powering robust clothes recognition and retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
Liu, Z. (2018). A deep learning method for suit detection in images. In IEEE International Conference on Signal Processing (ICSP) (pp. 439–444).
Lyu, X., Li, X., Zhang, Y., & Lu, W. (2024). Two-stage method for clothing feature detection. Big Data and Cognitive Computing, 8(4), 35. https://doi.org/10.3390/bdcc8040035
Nodari, A., Ghiringhelli, M., Zamberletti, A., Vanetti, M., Albertini, S., & Gallo, I. (2012). A mobile visual search application for content-based image retrieval in the fashion domain. In Proceedings of the International Workshop on Content-Based Multimedia Indexing (pp. 1–6). IEEE. https://doi.org/10.1109/CBMI.2012.6269838
Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., & El-Nouby, A. (2023). DINOv2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193. https://arxiv.org/abs/2304.07193
Pan, C., Yan, W. (2018) A learning-based positive feedback in salient object detection. International Conference on Image and Vision Computing New Zealand.
Pan, C., Yan, W. (2020) Object detection based on saturation of visual perception. Multimedia Tools and Applications, 79 (27-28), 19925-19944.
Pan, C., Liu, J., Yan, W., Zhou, Y. (2021) Salient object detection based on visual perceptual saturation and two-stream hybrid networks. IEEE Transactions on Image Processing.
Qin, D., Leichner, C., Delakis, M., Fornoni, M., Luo, S., Yang, F., Wang, W., Banbury, C., Ye, C., & Akin, B. (2025). MobileNetV4: Universal models for the mobile ecosystem. In Proceedings of the European Conference on Computer Vision (pp. 78–96). Springer.
Vijayaraj, A., Raj, P. T. V., Jebakumar, R., Senthilvel, P. G., Kumar, N., Suresh Kumar, R., & Dhanagopal, R. (2022). Deep learning image classification for fashion design. Wireless Communications and Mobile Computing, 2022, Article 7549397.
Sapkota, R., Meng, Z., Churuvija, M., Du, X., Ma, Z., & Karkee, M. (2024). Comprehensive performance evaluation of YOLOv12, YOLO11, YOLO10, YOLO9 and YOLO8 on detecting and counting fruitlet in complex orchard environments. arXiv preprint arXiv:2407.12040. https://arxiv.org/abs/2407.12040
Shen, J., Yan, W., Miller, P., Zhou, H. (2010) Human localization in a cluttered space using multiple cameras. IEEE International Conference on Advanced Video and Signal Based Surveillance.

Shin, S.-Y., Jo, G., & Wang, G. (2023). A novel method for fashion clothing image classification based on deep learning. Journal of Information and Communication Technology, 22(1), 128–145. https://doi.org/10.32890/jict2023.22.1.6

Shubathra, S., Kalaivaani, P. C. D., & Santhoshkumar, S. (2020). Clothing image recognition based on multiple features using deep neural networks. In Proceedings of the International Conference on Electronics and Sustainable Communication Systems (pp. 166–172). IEEE. https://doi.org/10.1109/ICESC48915.2020.9155959

Xiang, Z., Zhu, C., Qian, M., Shen, Y., Shao, Y., & Yizhou, S. (2024). FashionSegNet: A model for high-precision semantic segmentation of clothing images. The Visual Computer, 40(3), 1711–1727. https://doi.org/10.1007/s00371-023-02881-3

Xu, J., Wei, Y., Wang, A., Zhao, H., & Lefloch, D. (2022). Analysis of clothing image classification models: A comparison study between traditional machine learning and deep learning models. Fibres & Textiles in Eastern Europe, 30(5).

Yan, W. (2019) Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer Nature.

Yan, W. (2023) Computational Methods for Deep Learning: Theoretic, Practice and Applications. Springer Nature.

Yan, W. (2026) Robotic Vision: From Deep Learning to Autonomous System. Springer.

Yang, X., Zhao, W., Wang, Y., Yan, W., Li, Y. (2024) Lightweight and efficient deep learning models for fruit detection in orchards. Scientific Reports 14, 26086

Wang, J. (2023). Classification and identification of garment images based on deep learning. Journal of Intelligent & Fuzzy Systems, 44(3), 4223–4232. https://doi.org/10.3233/JIFS-220109

Wang, W., Han, C., Zhou, T., & Liu, D. (2022). Visual recognition with deep nearest centroids. arXiv preprint arXiv:2209.07383. https://arxiv.org/abs/2209.07383

Wen, M., Xia, T., Liao, B., & Tian, Y. (2023). Few-shot relation classification using clustering-based prototype modification. Knowledge-Based Systems, 268, 110477. https://doi.org/10.1016/j.knosys.2023.110477

Zhang, Y., Nguyen, M., Yan, W. (2025) Diffusion-based virtual try-on system. IEEE IVCNZ.

Zhang, Y. (2026) ChatClothes: An AI-Powered Virtual Try-On System. Master's Thesis, Auckland University of Technology, New Zealand.

Zhou, Z., Deng, W., Wang, Y., & Zhu, Z. (2022). Classification of clothing images based on a parallel convolutional neural network and random vector functional link optimized by the grasshopper optimization algorithm. Textile Research Journal, 92(9–10), 1416–1427. https://doi.org/10.1177/00405175211059207