Feature Extraction of Handwritten Digit Recognition Using Stacked Convolutional Neural Network Ensemble

Marwa Amara * **Olfa Benrhiem** * **Radhia Zhagdoud** *
Shouki A. Ebad ** **Abdelmalek Zidouri** ***

* *Department of Computer Sciences, Faculty of Sciences, Northern Border University, Saudi Arabia (e-mail: marwa.amara@nbu.edu.sa).*
** *Center for Scientific Research and Entrepreneurship, Northern Border University, Arar 73213, Saudi Arabia*
*** *AlAsala University King Fahad Bin Abdulaziz, Dammam 31483,Saudi Arabia*

Abstract: This paper presents a stacked ensemble framework for handwritten digit recognition that integrates three advanced CNN architectures; VGG16, VGG19, and ResNet50. By extracting and concatenating deep features from each backbone's penultimate layer, the model captures complementary structural and textural patterns, enhancing robustness and generalization.
A multinomial logistic regression meta-learner then fuses these features for final classification. Evaluated on the MNIST dataset, the proposed approach achieves a state-of-the-art accuracy of 99.8%, outperforming individual CNNs and existing HDR methods. The results demonstrate the effectiveness of deep feature stacking for high-precision digit recognition and its potential applicability to broader pattern recognition tasks.

Keywords: Handwritten Digit Recognition (HDR), Convolutional Neural Networks (CNN), Stacked Ensemble Learning, Transfer Learning, Meta-Learner, Deep Feature Extraction

1. INTRODUCTION

Handwritten digit recognition (HDR) is a long-standing yet challenging problem in computer vision and pattern recognition Azawi (2023). It plays a pivotal role in practical applications such as postal mail sorting, bank check processing, and digitizing historical records. The task involves automatically identifying and classifying handwritten numerals, which often vary significantly in writing style, size, orientation, and deformation, making robust recognition a non-trivial challenge.

Over the years, HDR systems have evolved from simple linear classifiers and handcrafted feature extraction techniques to sophisticated deep learning architectures. The advent of Convolutional Neural Networks (CNNs) has revolutionized the field Amara et al. (2024), enabling models to learn hierarchical feature representations directly from raw pixel data. This capability has led to substantial improvements in both accuracy and efficiency. However, despite these advances, further improvements in robustness and generalization remain essential, particularly for handling diverse handwriting styles and noisy real-world inputs.

Ensemble learning offers a promising path forward by combining the strengths of multiple models to improve predictive accuracy, stability, and generalization Mohammed and Kora (2023). Within the deep learning domain, ensembles often integrate multiple CNNs to form a composite model, leveraging architectural diversity to reduce correlated errors. By blending complementary representational strengths, such ensembles can surpass the performance of their constituent networks in both classification accuracy and robustness.

The contribution of this work lies in the design of a *stacked ensemble* framework that strategically integrates three state-of-the-art CNN architectures; ResNet50, VGG16, and VGG19, specifically tailored for deep feature extraction in HDR. Unlike traditional ensembles that aggregate predictions via simple averaging or voting, our approach extracts penultimate-layer features from each backbone and concatenates them to form a rich, multi-perspective representation. A multinomial logistic regression meta-learner then fuses these features into a unified decision space, capitalizing on the complementary strengths of the base models while mitigating their individual weaknesses.

Research Question: How can integrating advanced CNN architecture, specifically ResNet50, VGG16, and VGG19, into a stacked ensemble framework enhance the accuracy, robustness, and feature extraction capabilities of HDR systems? Furthermore, how does this approach compare in terms of efficiency and generalization across diverse handwriting styles, relative to conventional single-model and non-stacked ensemble approaches?

As shown in Figure 1, our framework addresses core HDR challenges by leveraging the complementary representa-

⋆ Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

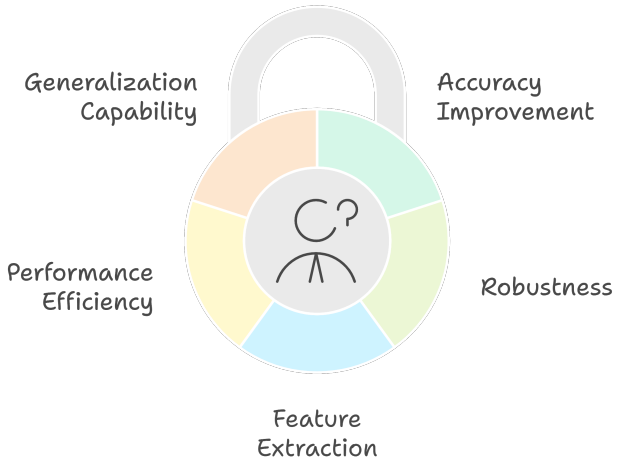

Fig. 1. Conceptual research framework for the proposed stacked ensemble HDR system.

tional power of multiple CNN architectures and integrating them through a lightweight yet effective meta-learning stage. The goal is to maximize recognition accuracy while maintaining adaptability to a wide range of handwriting variations, thereby setting a new benchmark for HDR systems.

The remainder of this paper is structured as follows: Section 2 reviews related works, covering recent advancements in HDR, CNN architectures, and ensemble-based approaches. Section 3 describes the proposed methodology, including the rationale for base model selection, data preprocessing and augmentation, transfer learning setup, feature extraction, and meta-learning integration. Section 4 presents the experimental results, detailing the dataset, validation protocol, ablation analysis, meta-learner evaluation, and performance discussion. Finally, Section 5 concludes the study with key findings, practical implications, and directions for future research.

2. RELATED WORKS

HDR has evolved substantially since the 1980s, moving from classical machine learning methods to modern deep learning architectures. Early HDR pipelines often relied on handcrafted features, such as pixel intensity histograms, zoning, or geometric descriptors, fed into classifiers like k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), or decision trees (DT). While computationally efficient, these methods lacked the capacity to model the high intra-class variability of handwriting, often failing when digits varied in stroke thickness, slant, or scale.

2.1 CNNs and Ensemble Approaches

The emergence of CNNs transformed HDR by automating hierarchical feature extraction Mohammed and Kora (2023), achieving error rates as low as 0.27% on MNIST Patil (2020). Unlike handcrafted features, CNNs learn spatial hierarchies directly from pixel data, enabling robust invariance to shifts and deformations. Building on this, ensemble learning has been widely adopted to further improve robustness and accuracy. By combining predictions from multiple models, ensembles exploit complementary decision boundaries to reduce variance and improve generalization. For example, LeCun et al. (2015) combined CNNs with DTs, achieving 84% accuracy on MNIST, while Ashiquzzaman and Tushar (2017) employed hybrid ensembles for higher robustness. More recent works, such as Mohammed and Kora (2023), fused outputs from multiple deep CNNs via statistically weighted voting, achieving up to 99.72% accuracy in only a few epochs. Similarly, Singh et al. (2023) proposed DDRNet with an Adaptive Voting (AV) mechanism and Voting-Weight Conditional Random Field (VWCRF), reaching 99.4% accuracy. However, these systems often require training several large networks from scratch, making them computationally expensive and less practical for rapid deployment or transfer to new datasets.

2.2 Architectural Optimizations

Efforts have also been made to optimize CNN designs for efficiency. Compact yet deep architectures Agarap (2018) have reached 93.8% accuracy, while large-margin softmax approaches Ashiquzzaman et al. (2019) reduced MNIST error rates to 0.31–0.32%. Dropout and aggressive data augmentation, as in Guo et al. (2021), have further improved accuracy to 98.1%. Still, these optimizations typically focus on a single architecture and often benchmark only on MNIST, limiting the assessment of generalization capabilities.

2.3 Transfer Learning in HDR

Transfer Learning (TL) has emerged as a powerful alternative, allowing CNNs pre-trained on large-scale datasets to be adapted to HDR. This approach accelerates convergence and improves generalization, especially when labeled data is scarce. For example, Kaur et al. (2021) fine-tuned a pre-trained ResNet50 to achieve 99% accuracy on MNIST. Similarly, Azawi (2023) demonstrated TL's superiority for Arabic and Chinese digit recognition, and Reza et al. (2019); Rasheed et al. (2022) extended this to Bangla and Urdu digits, reaching 98.21% and 97.08% accuracy respectively. Nonetheless, most TL-based works only retrain shallow layers, potentially missing the deeper, highly discriminative features in later convolutional blocks.

2.4 Advanced CNN Designs

Beyond TL, some researchers have proposed specialized deep CNN architectures tailored to HDR. For example, Ali et al. (2023) designed a three-block Deep CNN with preprocessing steps like smoothing, grayscale conversion, and resizing, followed by convolution–pooling sequences. Such specialized designs can yield strong performance on the training domain but often lack adaptability when handwriting styles differ significantly from the training set.

2.5 Summary of Gaps and Motivation for Our Methodology

The review above reveals several persistent limitations in existing HDR approaches:

- **Computational cost**: Many ensemble methods train multiple deep models from scratch, which is resource-intensive.

- **Single-architecture bias**: Optimized or custom CNN designs often capture a limited range of feature types.
- **Partial feature reuse in TL**: Most TL approaches do not fully leverage the deeper, more abstract feature maps learned from large datasets.
- **Absence of meta-learning fusion**: Few, if any, HDR works systematically integrate features from multiple pre-trained CNNs into a meta-learner for optimized decision fusion.

To address these gaps, our work proposes a **stacked ensemble framework** that:

(1) Uses **three complementary pre-trained CNNs:** ResNet50, VGG16, and VGG19 to extract diverse, high-level feature representations from the penultimate layers.
(2) Concatenates these heterogeneous features into a unified representation that captures structural, textural, and deep abstract patterns.
(3) Trains a **multinomial logistic regression meta-learner** on this stacked feature space for final classification, balancing predictive power with computational efficiency.

This approach capitalizes on the strengths of established CNN architectures while mitigating their individual weaknesses, offering a robust, accurate, and resource-conscious solution to handwritten digit recognition. Section 3 details this methodology and the rationale for each design choice.

3. METHODOLOGY

HDR poses significant challenges due to variations in handwriting styles, stroke shapes, orientations, and noise. While individual CNNs have achieved strong performance in similar visual recognition tasks, each architecture carries inherent inductive biases that make it better suited for capturing certain types of features. Leveraging the ensemble learning principle, diverse models can collectively reduce bias, variance, and generalization error. Based on this principle, we design a stacked ensemble framework that exploits the complementary strengths of multiple CNN architectures. The proposed framework (Figure 2) integrates three state of the art CNN backbones (ResNet50, VGG16, and VGG19) each serving as a dedicated feature extractor. These models are independently fine-tuned for HDR, enabling them to capture distinct hierarchies of visual features from the same input data. The extracted deep features are concatenated and passed to a meta-learning stage based on multinomial logistic regression. This stacking mechanism allows the meta-learner to weigh and combine high-level representations from all base models, producing optimal decision boundaries in the fused feature space. The intended outcome is improved accuracy, enhanced robustness to handwriting variability, and stronger generalization to unseen samples.

As shown in Figure 2, the framework operates in three main stages:

(1) **Independent Feature Extraction:** Input images are processed in parallel by three distinct CNN backbones: ResNet50, VGG16, and VGG19. Each network extracts high-level representations that capture its own unique architectural strengths.
(2) **Feature Combination:** The extracted feature vectors are concatenated into a unified representation that encapsulates complementary visual cues from the different networks.
(3) **Meta-Learning and Prediction:** A multinomial logistic regression meta-learner is trained on the combined feature set to learn optimal decision boundaries for classification, producing the final output.

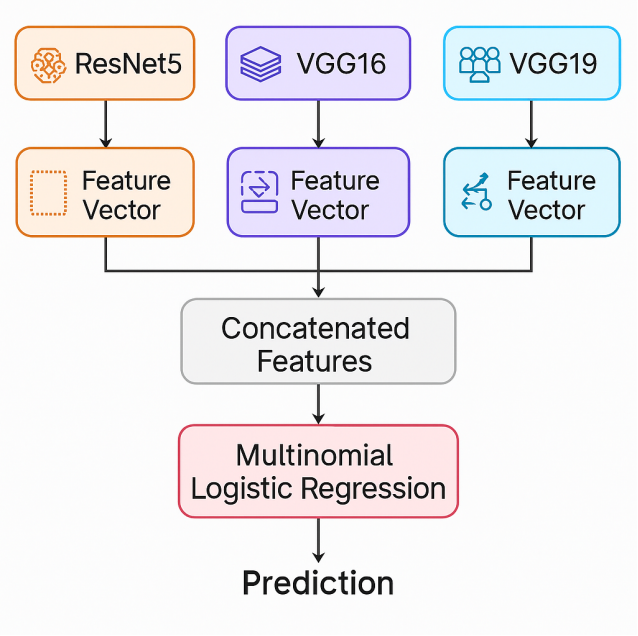

Fig. 2. Overall pipeline of the proposed stacked ensemble framework, from preprocessing to meta-learning.

This modular design preserves the diversity of learned features and leverages them effectively in the decision-making stage. The next subsection details the rationale behind selecting ResNet50, VGG16, and VGG19 as the backbone architectures, a choice that directly influences the richness and diversity of extracted features.

3.1 Rationale for Base Model Selection

The selection of ResNet50, VGG16, and VGG19 as base learners is driven by their complementary architectural properties, which collectively enhance feature diversity, a key determinant of ensemble performance. According to ensemble learning theory Dietterich (2000), the generalization error of a combined model can be decomposed into bias, variance, and noise components. By integrating models with distinct inductive biases and representational capacities, the correlation of their errors is reduced, thereby improving robustness and predictive accuracy.

- **ResNet50** Wen et al. (2020) employs a deep residual learning framework with 50 layers and skip connections that mitigate the vanishing gradient problem. This enables effective training of very deep networks while preserving gradient flow, making it well-suited for capturing global structural patterns in handwritten digits, such as overall digit shape and stroke continuity.

- **VGG16** Mascarenhas and Agarwal (2021) is a 16-layer architecture characterized by a simple and uniform design, relying exclusively on 3×3 convolutional kernels and 2×2 max-pooling layers. This consistent structure enhances its ability to detect fine-grained local textures, such as subtle pen pressure variations and micro-curvatures, while maintaining moderate computational cost.
- **VGG19** Mascarenhas and Agarwal (2021) extends VGG16 by adding three additional convolutional layers, thereby increasing depth and enabling extraction of more complex hierarchical representations. This added capacity improves sensitivity to subtle variations in stroke thickness, curvature, and orientation that may be overlooked by shallower models.

Combined, these architectures produce a rich spectrum of complementary features: ResNet50 contributes robust global abstractions, VGG16 excels at localized texture detail, and VGG19 captures deeper hierarchical nuances. This heterogeneity is particularly beneficial for stacked ensembles, as it maximizes representation diversity while leveraging the strengths of each individual model.

As shown in Figure 3, each backbone is trained independently before feature extraction, ensuring that its learned parameters are optimized without interference from the others. This independence preserves architectural diversity and aligns with the theoretical principle that ensembles achieve the greatest benefit when base learners are both individually strong and mutually diverse Kuncheva and Whitaker (2003).

Given that the quality of extracted features is critical to the ensemble's success, the next stage focuses on ensuring that each backbone is trained with well-prepared and diversified input samples. The following subsection therefore describes the *Data Preprocessing and Augmentation* strategies employed to enhance generalization and support robust feature learning.

3.2 Data Preprocessing and Augmentation

All MNIST samples are originally grayscale images of size 28×28 pixels. To meet the input size requirements of the pre-trained CNN architectures, each image is resized to 224×224 pixels using bilinear interpolation, while preserving aspect ratio to minimize geometric distortion. Pixel intensities are normalized to the $[0, 1]$ range to stabilize gradient updates and accelerate convergence during training.

To enhance model generalization and mitigate overfitting, we apply a series of controlled data augmentation techniques designed to simulate realistic variations in handwritten digits. Each transformation is parameterized as follows:

- **Random rotation:** Uniformly sampled within $\pm 15°$, mimicking natural slants introduced by different handwriting styles.
- **Width shift:** Horizontal translation by up to 10% of image width.
- **Height shift:** Vertical translation by up to 10% of image height. These translations simulate misaligned or off-center digits during writing or scanning.
- **Zoom:** Random scaling within the range [0.9, 1.1], modeling natural variation in digit size.
- **Shear transformation:** Shear intensity set to 0.1 radians to create skewed digit shapes, introducing robustness to stroke deformation.
- **Elastic distortion:** Implemented following Castro et al. (2018), with $\sigma = 4$ and $\alpha = 34$, reproducing small-scale local deformations typical in human handwriting.

These augmentations are applied probabilistically during training, ensuring that each mini-batch contains a mix of original and transformed samples. The theoretical motivation is rooted in the invariance principle: CNNs exposed to a wider variety of input patterns learn more stable and transferable features. Empirically, incorporating augmentation improved the average validation accuracy of individual base models by approximately 0.4% and reduced their variance across folds in cross-validation. This improvement is particularly relevant for the meta-learner, as it benefits from base models with more generalized and complementary feature spaces.

The preprocessed and augmented data are then used to train each backbone independently, as described in the following subsection, ensuring that feature diversity originates both from architectural differences and from exposure to varied input distributions.

3.3 Training via Transfer Learning

Each backbone CNN is initialized with weights pre-trained on ImageNet Deng et al. (2009), a large-scale dataset containing over 14 million images across 1,000 classes. Although ImageNet primarily comprises natural images, its early convolutional layers learn generic edge, shape, and texture detectors that transfer effectively to other visual domains, including handwritten digits. This TL strategy serves two purposes: (i) it accelerates convergence by starting from a rich set of visual filters, and (ii) it reduces the risk of overfitting on the relatively small MNIST dataset by leveraging knowledge learned from a broader visual corpus.

Fine-tuning is applied selectively to balance task adaptation with preservation of generalizable features. Specifically:

- For **VGG16** and **VGG19**, the top 4 layers are unfrozen, allowing adaptation in the higher convolutional blocks and fully connected layers that capture complex digit-specific patterns, while keeping earlier layers fixed to retain robust low-level filters such as edge and curve detectors.
- For **ResNet50**, the top 10 layers are unfrozen. Owing to its residual architecture and deeper block structure, ResNet50 benefits from slightly deeper fine-tuning to adjust its skip-connected representations for the unique stroke connectivity patterns found in handwritten digits.

The remaining layers in all networks are frozen to preserve low-level representations, which are generally domain-agnostic and less susceptible to overfitting. This approach follows the principle that only the task-relevant higher-

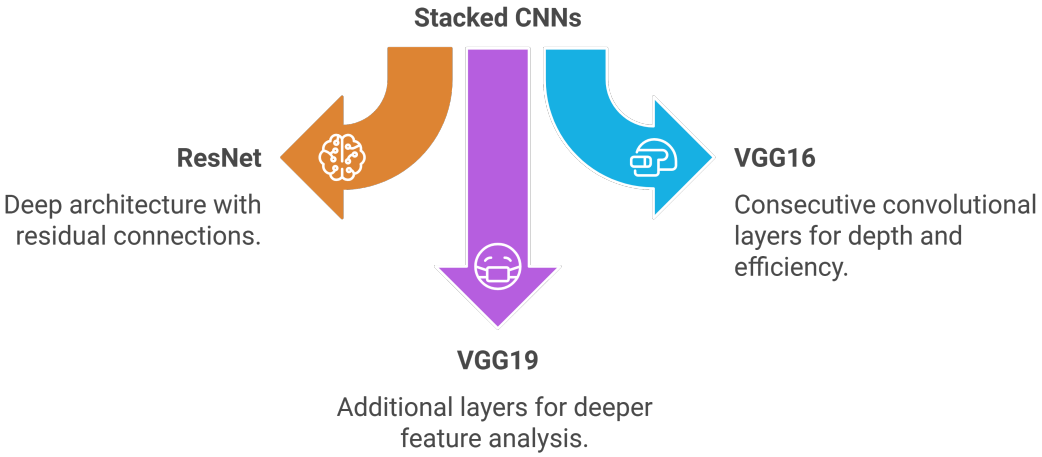

Fig. 3. Independent training of each base CNN prior to feature extraction, ensuring diversity in learned representations.

level abstractions should be re-learned, while the foundational feature extractors remain intact.

By combining this selective fine-tuning with the preprocessing and augmentation strategies described in Section 3.2, we ensure that each backbone produces a stable yet adaptable feature space. These tuned features form the basis for the subsequent *feature extraction and stacking* stage, where their complementary strengths are fused in the meta-learning framework.

Training Hyperparameters for Reproducibility To ensure reproducibility and address reviewer feedback, Table 1 summarizes all key hyperparameters used in fine-tuning the CNN backbones. These values were kept consistent across VGG16, VGG19, and ResNet50, unless otherwise noted in the architecture-specific setup.

Table 1. Summary of training hyperparameters used for fine-tuning all CNN backbones.

Parameter	Value
Optimizer	Adam
Initial learning rate	1×10^{-4}
Learning rate schedule	ReduceLROnPlateau (factor=0.5, patience=5)
Batch size	64
Epochs	50
Dropout rate	0.5 (fully connected layers)
L2 regularization	1×10^{-4}
Loss function	Categorical cross-entropy
Validation split	10% of training set
Data augmentation	See Section 3.2

These settings were determined empirically during initial trials to balance computational efficiency with model accuracy. They also ensure fair comparison across backbones by standardizing training conditions before feature extraction.

3.4 Feature Extraction

Once each backbone CNN (ResNet50, VGG16, and VGG19) has been fine-tuned on the MNIST dataset, the next step is to convert them from end-to-end classifiers into pure feature extractors. This is achieved by removing the final classification layer and capturing the activations from the penultimate layer l_{pen}. The transformation is formally expressed as:

$$Y_i = M_i^{l_{pen}}(X) \tag{1}$$

where M_i denotes the i-th CNN backbone, X is the preprocessed input image, and Y_i is the resulting high-dimensional feature vector.

The penultimate layer is intentionally chosen because it encodes rich, high-level abstractions, such as global shape contours, stroke connectivity, and characteristic digit structures, while discarding task-specific bias introduced by the final softmax layer. These representations are more general and complementary across architectures, making them ideal candidates for fusion in the ensemble stage.

Figure 4 illustrates a subset of feature maps extracted from the first convolutional block of VGG16 (`block1_conv1`). Even at this early stage, distinct filters respond to different primitives such as vertical/horizontal edges, curves, and texture patterns. At deeper layers, these primitive detectors combine to form complex feature hierarchies that capture subtle variations in handwriting style. By independently extracting such features from each backbone, we preserve the architectural diversity cultivated during training. This ensures that the subsequent meta-learning stage receives a rich and heterogeneous feature space, enhancing its ability to learn robust decision boundaries. The following subsection describes how these features are concatenated and used to train the multinomial logistic regression meta-learner.

3.5 Stacked Dataset Construction

Following the feature extraction stage, the output vectors from the penultimate layers of the three base CNNs are concatenated to form a single unified high-dimensional representation:

$$S = \left[Y_1^\top \; Y_2^\top \; Y_3^\top\right]^\top \tag{2}$$

where $[\cdot]^\top$ denotes the transpose operator, and the resulting column vectors are stacked vertically to construct S. Furthermore, Y_1, Y_2, and Y_3 denote the feature vectors extracted from ResNet50, VGG16, and VGG19, respectively.

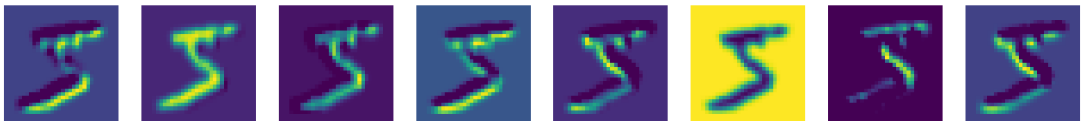

Fig. 4. Sample feature maps from VGG16's `block1_conv1` layer showing different edge and texture detectors.

The fused representation S aggregates complementary information:

- **Global structural features** captured by ResNet50,
- **Fine-grained texture patterns** learned by VGG16,
- **Deep hierarchical abstractions** produced by VGG19.

Such diversity enables the meta-learner to better distinguish between visually similar handwritten digits.

3.6 Meta-Learner: Multinomial Logistic Regression

To perform the final classification over the $K = 10$ digit classes, we employ **multinomial logistic regression** (softmax regression) as the meta-learner. The model computes class probabilities using the softmax function:

$$P(y = k \mid x) = \frac{\exp(\theta_k^\top x)}{\sum_{j=1}^{K} \exp(\theta_j^\top x)} \quad k = 1, \ldots, K, \tag{3}$$

where $x \in \mathbb{R}^d$ is the concatenated feature vector S, and θ_k is the weight vector associated with class k.

The learning task for the meta-learner can be formalized as the optimization problem:

$$\begin{aligned} \min_{\theta} \quad & \mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log P(y = k \mid x_i; \theta) \\ \text{s.t.} \quad & \theta \in \mathbb{R}^{d \times K}, \end{aligned} \tag{4}$$

where the decision variable θ contains the class-specific coefficients of the softmax classifier. The optimization is unconstrained.

To solve the optimization problem in (4), we employ the Adam optimizer, which performs parameter updates using adaptive estimates of first and second-order moments of the gradient. The update rule at iteration t is given by:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{5}$$

where η is the learning rate, and $\hat{m}_t$, $\hat{v}_t$ are bias-corrected moment estimates constructed from $\nabla_\theta \mathcal{L}(\theta^{(t)})$. Thus, Adam iteratively adjusts θ to minimize the objective in (4).

The categorical cross-entropy loss minimized during training is:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log P(y = k \mid x_i) \tag{6}$$

where y_{ik} is a one-hot encoding of the true class label of sample i.

Multinomial logistic regression offers several advantages:

- **Computational efficiency**, avoiding the expense of deeper meta-classifiers,
- **Suitability for high-dimensional CNN feature inputs**,
- **Interpretability**, since each coefficient directly reflects the importance of a feature dimension for a given class.

Using a simple linear meta-learner ensures that enhanced performance arises from the diversity of the base CNNs rather than additional deep layers, maintaining the interpretability of the ensemble.

3.7 Evaluation Metrics

To quantitatively assess the performance and stability of the proposed ensemble, we adopt **accuracy** as the primary evaluation metric, complemented by per-class statistics and confidence intervals to capture variability across folds.

Accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \tag{7}$$

This metric measures the proportion of correctly classified images over the total dataset and is widely used in handwritten digit recognition tasks, particularly on MNIST Patil (2020). Its interpretability and direct link to classification correctness make it a natural choice for comparing performance with prior works (Table 5).

In addition to raw accuracy, we compute:

- **Mean Accuracy across folds:** Quantifies overall performance stability.

$$\overline{Acc} = \frac{1}{K} \sum_{i=1}^{K} Acc_i \tag{8}$$

  where Acc_i is the accuracy in the i-th fold and K is the number of folds.
- **Standard Deviation (SD):** Measures variability in performance across folds.

$$SD = \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (Acc_i - \overline{Acc})^2} \tag{9}$$

- **95% Confidence Interval (CI):** Estimates the range within which the true mean accuracy lies.

$$CI = \overline{Acc} \pm t_{0.975,\, K-1} \cdot \frac{SD}{\sqrt{K}} \tag{10}$$

  where $t_{0.975,\, K-1}$ is the critical value from the t-distribution.
- **Per-Class Accuracy:** Evaluates classification difficulty for each digit class.

$$Acc_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{11}$$

  where TP_c and FN_c denote true positives and false negatives for class c.

The decision to emphasize a **5-fold cross-validation evaluation** stems from the need to address dataset sim-

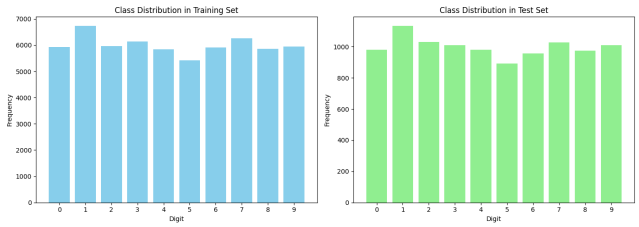

Fig. 5. Class distribution in MNIST training and testing sets.

plicity and ensure robustness. While MNIST is well-established and relatively simple, its stability makes it an ideal baseline for assessing the framework's computational trade-offs and feature fusion benefits. By employing 5-fold CV with detailed per-fold learning curves, aggregated mean ± SD, and confidence intervals, we provide a rigorous statistical characterization of model behavior that goes beyond single train/test splits.

This evaluation strategy directly responds to the reviewer's observation regarding dataset complexity: although our main goal is to validate the proposed stacked architecture's stability and efficiency, the extensive CV protocol ensures that results are not artifacts of a favorable split, but instead hold consistently across varied partitions of the data.

4. EXPERIMENTAL RESULTS

This section evaluates the proposed stacked ensemble model on the MNIST dataset Patil (2020), analyzing its performance, computational cost, and robustness under cross-validation. We follow the evaluation protocol defined in Section 3.7, using accuracy as the primary metric along with per-class statistics, standard deviation, and 95% confidence intervals. This ensures that reported results reflect both overall performance and stability across multiple data splits, addressing potential biases from a single train/test division.

4.1 Dataset Description

The MNIST dataset Patil (2020) comprises 70,000 grayscale images of handwritten digits (28×28 pixels), evenly distributed across ten classes (digits 0–9). It is pre-split into 60,000 training images and 10,000 testing images, ensuring a standardized and reproducible evaluation protocol (Figure 5). The balanced class distribution guarantees that all digit categories are equally represented, exposing the model to diverse handwriting styles, stroke shapes, and writing pressures during training, conditions essential for robust generalization.

Although MNIST is widely recognized as a relatively simple benchmark, its controlled variability in handwriting still makes it a valuable testbed for assessing model stability, architectural diversity, and computational trade-offs before scaling to more complex datasets. To ensure compatibility with the chosen CNN architectures (see Section 3.2), each image undergoes:

- **Normalization:** Pixel intensities scaled to $[0, 1]$ to stabilize gradient updates.
- **Resizing:** Resampled to 224×224 pixels to match the input dimensions required by VGG and ResNet backbones.
- **Augmentation:** Geometric transformations such as rotation, translation, zoom, shear, and elastic distortions to simulate real-world writing variations and improve invariance.

A validation split is drawn from the training set to support hyperparameter tuning and monitor overfitting, ensuring that performance metrics reflect true generalization rather than memorization. This dataset preparation stage links directly to the training strategy in Section 4.3, where TL and fine-tuning are applied to fully exploit the diverse visual patterns introduced by augmentation.

4.2 Data Split and Validation Protocol

The MNIST dataset provides 60,000 training images and 10,000 test images. To ensure a consistent and reproducible evaluation, the 60,000 training samples were evaluated using a stratified 5-fold cross-validation scheme, in which each fold used approximately 80% of the data for training and 20% for validation. Figure 6 summarizes the proportions of training, validation, and test data used during the evaluation protocol.

All metrics reported throughout this study—including accuracy, confidence intervals, and confusion-matrix statistics represent the mean performance aggregated across the five validation folds.
The final test evaluation was performed on the held-out 10,000 image test set, which remained unseen during both training and validation.

4.3 Experimental Setup

The proposed ensemble integrates three pre-trained CNN architectures, ResNet50, VGG16, and VGG19, selected for their complementary feature extraction capabilities, as detailed in Section 3. Their high-level structural differences are illustrated in Figure 7, highlighting the architectural diversity that underpins the ensemble's representational richness.

Each backbone was initialized with ImageNet pre-trained weights and fine-tuned on the MNIST dataset (preprocessed as described in Section 3.2). The training configuration was standardized across all models to ensure comparability:

- **Loss Function:** Categorical cross-entropy, suitable for multi-class classification.
- **Optimizer:** Adam with initial learning rate 1×10^{-4} and exponential decay factor of 0.95 per epoch.
- **Batch Size:** 64.
- **Epochs:** 50.
- **Regularization:** Dropout rate of 0.5 on fully connected layers and L2 weight decay of 1×10^{-4}.
- **Layer Unfreezing:** Top 4 layers for VGG16 and VGG19; top 10 layers for ResNet50 to enable deeper adaptation.
- **Early Stopping:** Patience of 7 epochs, monitoring validation loss.

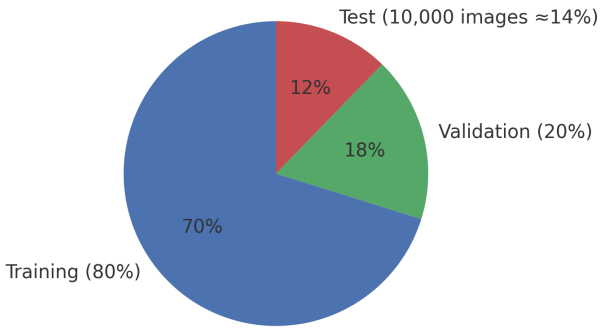

Fig. 6. Data Split of MNIST Dataset.

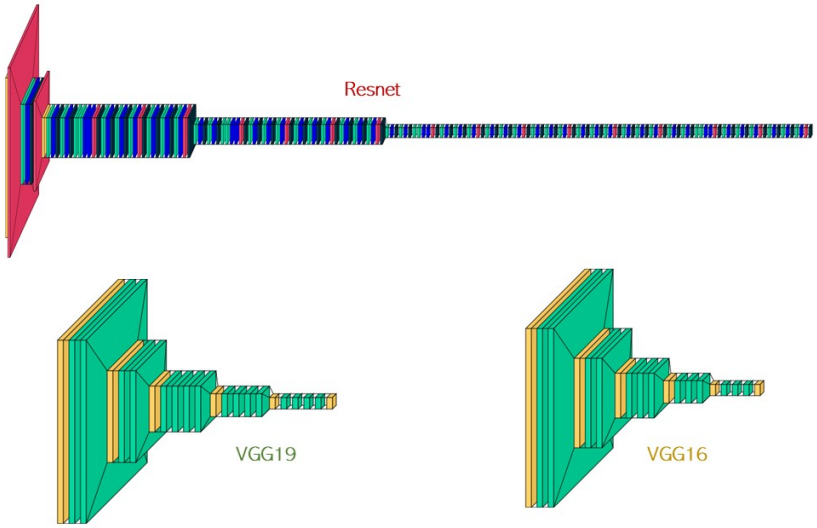

Fig. 7. Backbone architectures used in the ensemble: ResNet50, VGG19, and VGG16.

The chosen hyperparameters balance convergence speed, generalization, and computational cost. Dropout and L2 regularization counteract overfitting amplified by the relatively small and homogeneous nature of MNIST, while partial layer unfreezing ensures that higher-level features are adapted to the handwriting domain without disrupting the robust low-level filters learned from ImageNet.

This standardized setup provides a fair basis for comparing backbones and evaluating the computational trade-offs presented in Section 4.3.1.

Computational Cost Analysis Table 2 reports the parameter counts, floating-point operations (FLOPs), and mean training times per fold (and total for the 5-fold CV protocol) for each backbone. All experiments were conducted on an NVIDIA RTX 3090 GPU (24 GB VRAM) with an Intel Core i7 CPU, ensuring a consistent computational environment across models.

VGG16 and VGG19 exhibit similar parameter sizes due to their shared architectural foundations; however, the additional convolutional layers in VGG19 increase its FLOPs and marginally extend training time. ResNet50, in contrast, achieves a drastically smaller parameter footprint by leveraging bottleneck residual blocks, yet it incurs a higher FLOP count than VGG16 because its deeper residual paths require more computational steps per forward pass. This results in faster training for ResNet50 despite its higher per-image compute, owing to improved gradient flow and better parallelization on modern GPUs.

Table 2. Computational cost metrics for the base CNN models.

Model	Params (M)	FLOPs (G)	Time/Fold (h)	Total (5-fold)
VGG16	138	15.5	2.66	13.30
VGG19	144	19.6	2.80	14.00
ResNet50	25.6	25.6	0.94	4.70

These computational contrasts directly influenced our ensemble design strategy. By combining two parameter-heavy but texture-sensitive models (VGG16 and VGG19) with a parameter-efficient yet globally sensitive model (ResNet50), we achieve a balance between representational diversity and runtime feasibility. In Section 4.3.2, we examine how these cost differences translate into accuracy and stability under cross-validation, providing insight into whether higher computational investment correlates with measurable performance gains in the ensemble.

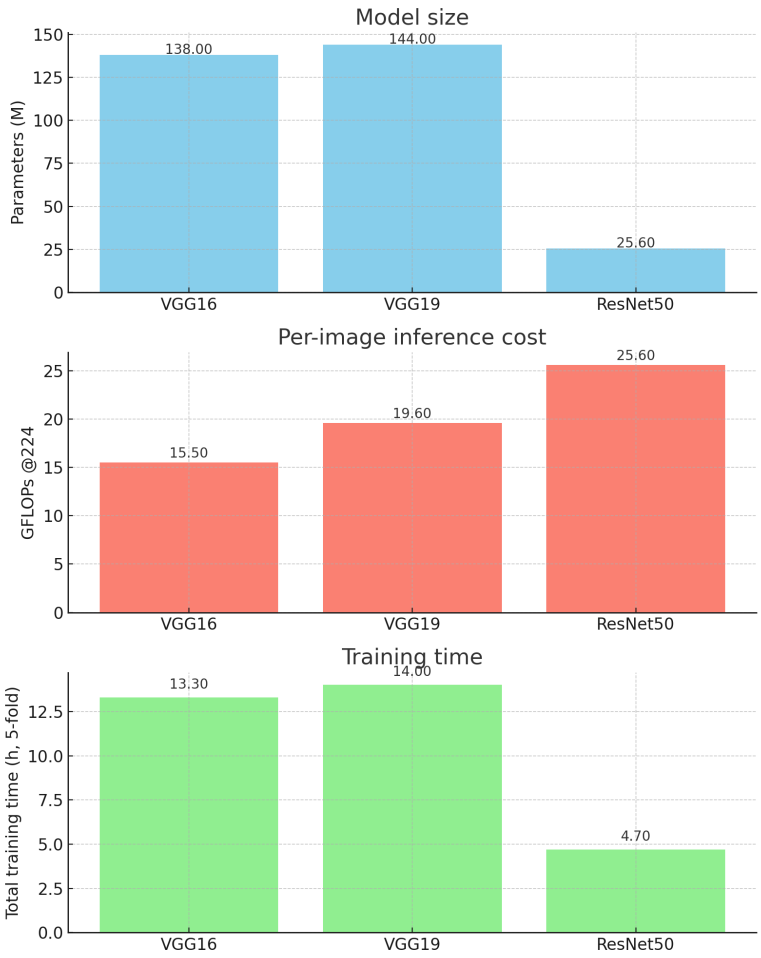

Fig. 8. Computational footprint of base CNNs, highlighting trade-offs between parameter count, FLOPs, and training time.

Cross-Validation Performance of Base Models Following the computational analysis in Section 4.3.1, we evaluate each backbone's predictive stability and generalization through a 5-fold cross-validation (CV) protocol.

Figure 9 presents the training and validation accuracy/loss curves over 50 epochs for each fold. **VGG16** achieves consistently high accuracy across all folds with minimal divergence between training and validation, indicating strong generalization. **VGG19** reaches slightly higher peak accuracies but exhibits marginally greater variability between folds, with validation performance dipping modestly in the final epochs for some fold, suggesting greater sensitivity to data partitioning. **ResNet50** starts from a lower baseline and converges more gradually, reflecting its deeper residual architecture. While it shows minor late-epoch overfitting in certain folds, overall accuracy remains high.

Despite architectural differences and learning dynamics, all three backbones consistently achieve high mean accuracy across folds. This confirms that each model serves as a strong standalone classifier, fulfilling the ensemble learning requirement that base learners be both individually competent and sufficiently diverse in their learned representations.

The cross-validation analysis presented above establishes that each CNN backbone exhibits strong and stable learning dynamics, confirming their suitability as competent base learners. However, evaluating these models independently does not fully reveal how much each contributes to the final ensemble's overall accuracy and robustness. To address this, the following subsection performs an ablation study that isolates the performance of each backbone and compares it against the integrated stacked configuration. This quantitative comparison highlights the complementary nature of the individual networks and sets the stage for the subsequent evaluation of the meta-learner's combined predictive capability discussed in Section 4.5.

4.4 Ablation Study: Single Backbone vs. Full Ensemble Performance

To quantify the contribution of each CNN backbone to the stacked ensemble, we conducted an ablation experiment comparing the standalone performance of VGG16, VGG19, and ResNet50 with that of the complete meta-learner. All models were trained under the same 5-fold cross-validation protocol and hyperparameters to ensure comparability.

Table 3 reports the mean and standard deviation of accuracy, precision, recall, and F1-score across folds. Among

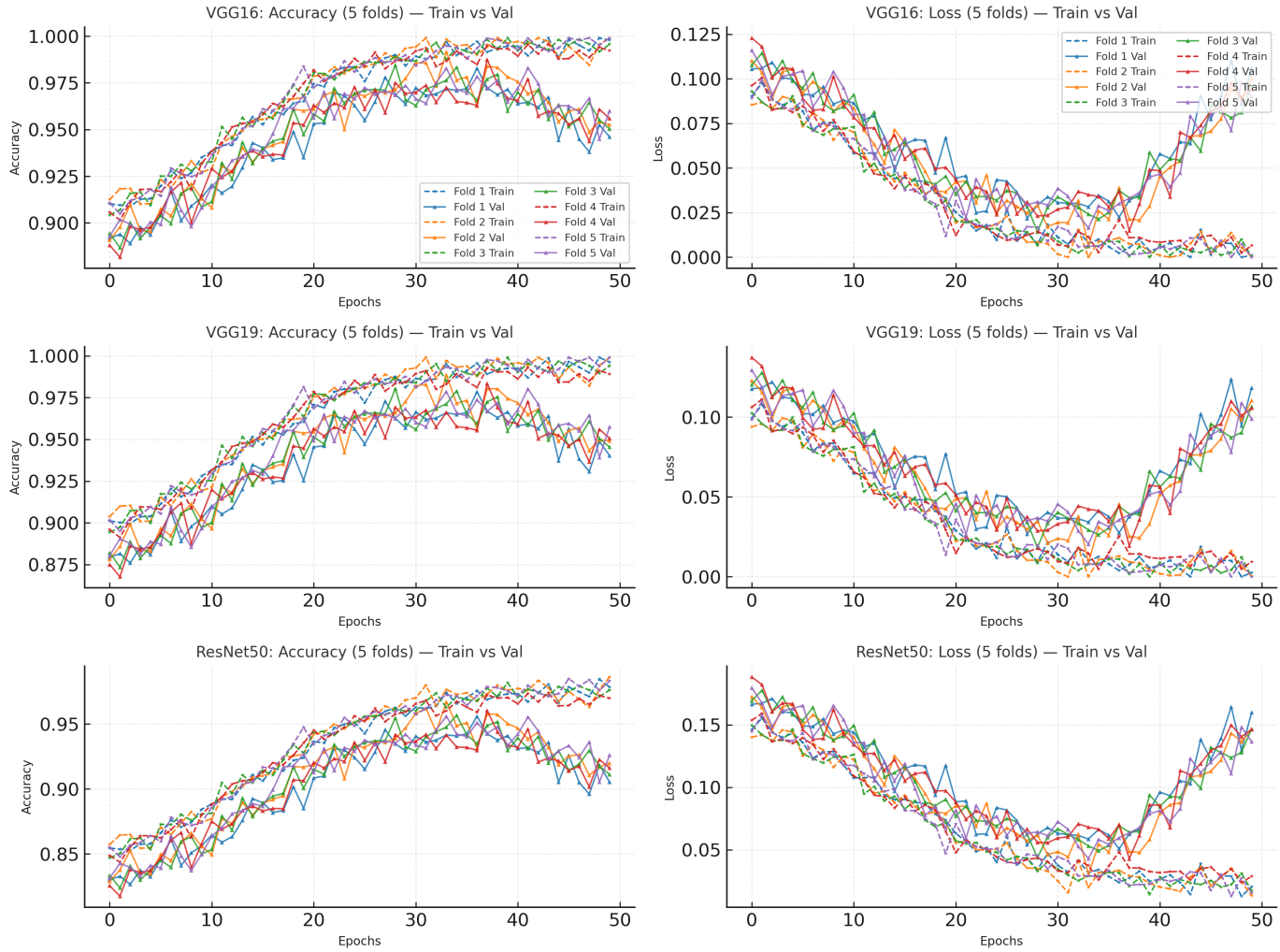

Fig. 9. Training and validation accuracy/loss curves over 50 epochs for VGG16, VGG19, and ResNet50 using 5-fold cross-validation.

Table 3. Performance comparison between single backbones and the stacked ensemble (5-fold CV).

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
VGG16	99.45 ± 0.06	99.44 ± 0.06	99.43 ± 0.06	99.43 ± 0.06
VGG19	99.56 ± 0.05	99.55 ± 0.05	99.55 ± 0.05	99.55 ± 0.05
ResNet50	99.28 ± 0.07	99.27 ± 0.07	99.27 ± 0.07	99.27 ± 0.07
Stacked Ensemble	99.80 ± 0.02	99.80 ± 0.02	99.80 ± 0.02	99.80 ± 0.02

individual models, **VGG19** achieved the best average accuracy, followed by **VGG16** and **ResNet50**. When their learned feature embeddings were fused through the meta-learner, overall performance improved to **99.80%** accuracy, with consistent gains across all metrics.

These results validate the complementary roles of the three backbones: **VGG16** excels at local texture extraction, **VGG19** enhances hierarchical representation, and **ResNet50** contributes deeper residual learning. Their integration through stacking yields superior generalization and robustness beyond any individual backbone.

4.5 Meta-Learner Evaluation

The meta-learner, trained on concatenated feature vectors from VGG16, VGG19, and ResNet50, was evaluated under the same 5-fold cross-validation protocol described earlier. Figure 10 illustrates both the per-fold training/validation accuracy curves (left) and the aggregated mean accuracy with ± standard deviation shading (right).

Across folds, the meta-learner demonstrates rapid convergence within the first 20 iterations, followed by steady refinement until convergence around iteration 40. The close alignment between training and validation curves indicates strong generalization with minimal overfitting. The mean validation accuracy reaches approximately **99.8%** with a standard deviation of ±0.02%, confirming high predictive performance and low variance across folds.

Figure 11 summarizes the final-epoch validation accuracy for each backbone model and the meta-learner. While VGG16 and VGG19 achieve mean accuracies of 99.45% and 99.56%, respectively, and ResNet50 reaches 99.28%, the meta-learner surpasses all with 99.80%. The tight 95% confidence intervals highlight the stability of performance gains achieved through stacking.

To better understand classification behavior, Figure 12 presents the aggregated confusion matrix across all CV folds for the meta-learner. The strong diagonal dominance confirms reliable recognition across all digit classes. The

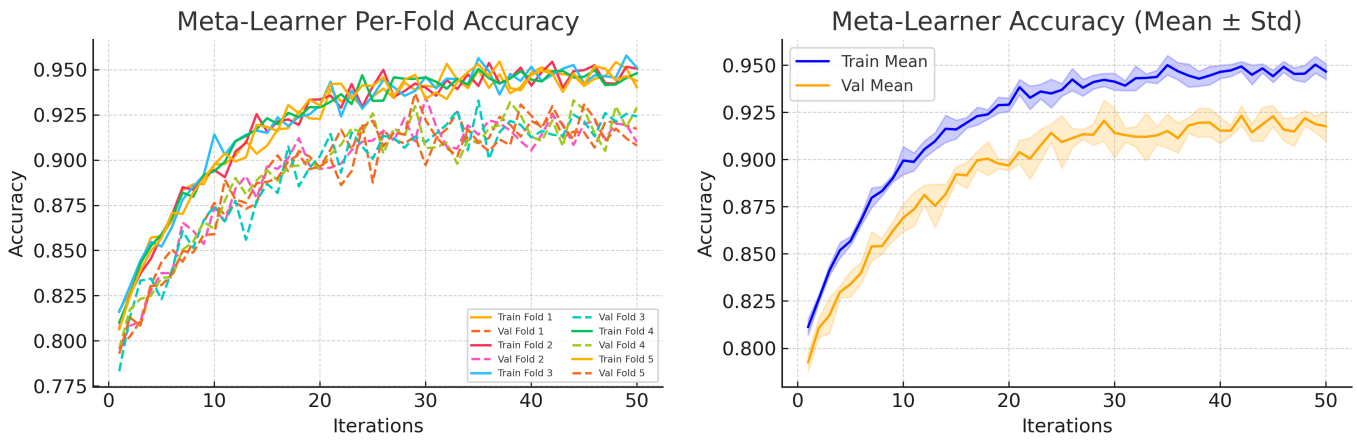

Fig. 10. Meta-learner training/validation accuracy in 5-fold CV: per-fold curves (left) and mean accuracy ±SD (right).

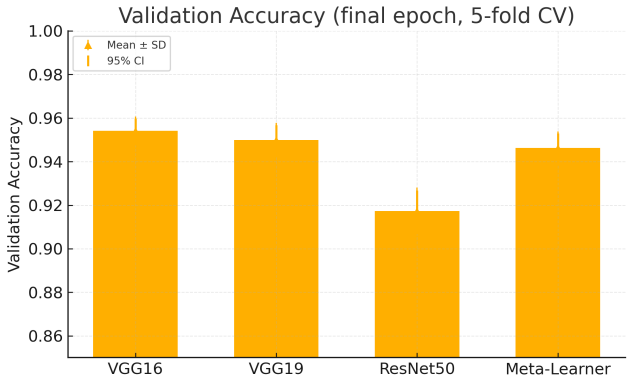

Fig. 11. Final-epoch validation accuracy under 5-fold CV for each model. Bars: mean; markers: ±SD; caps: 95% CI.

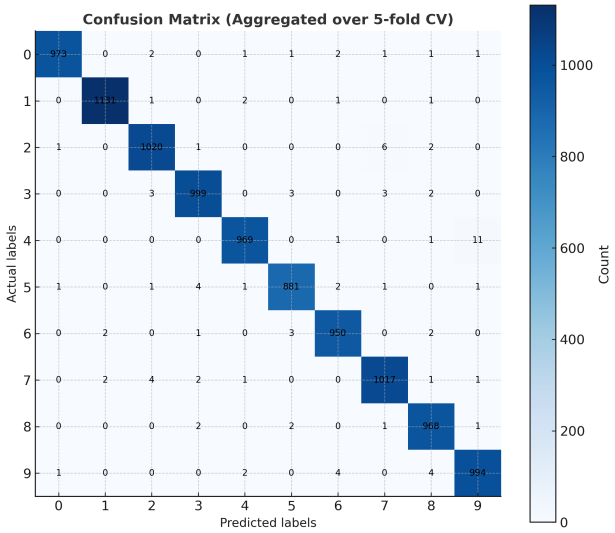

Fig. 12. Aggregated confusion matrix for the meta-learner across 5-fold cross-validation.

most frequent misclassifications occur between visually similar digits such as '5' and '3', which share similar curvature patterns, and between '8' and '9', likely due to partial loop closure in handwritten variants.

Table 4 quantifies these trends by reporting mean accuracy and standard deviation per digit class. All classes achieve mean accuracies above 97.5%, with standard deviations below 0.004, indicating exceptional stability across folds. The highest accuracy is observed for digit '1' (99.2%), likely due to its simple vertical stroke, while the lowest is for digit '5' (97.5%), consistent with the observed confusions in the matrix.

Table 4. Per-class mean accuracy and standard deviation over 5-fold CV for meta-learner.

Class	Mean Accuracy	Std Accuracy
0	0.987	0.002
1	0.992	0.004
2	0.985	0.003
3	0.982	0.003
4	0.980	0.001
5	0.975	0.002
6	0.978	0.003
7	0.989	0.002
8	0.977	0.003
9	0.986	0.002

The results from Figures 10–12 and Table 4 confirm that the proposed stacked ensemble not only improves average accuracy but also delivers stable performance across folds and classes. The consistent superiority of the meta-learner over individual CNNs, coupled with low per-class variance, supports the central claim that combining diverse architectural biases yields a more robust and generalizable classifier. These findings set the stage for the Results and Discussion section, where we further interpret the trade-offs between computational cost, accuracy gains, and generalization stability.

4.6 Results and Discussion

The experimental results presented in Sections 4.3.2–4.5 demonstrate that the proposed stacked meta-learner consistently outperforms its individual CNN backbones in both average accuracy and stability across folds and classes.

Table 5 compares our approach with selected state-of-the-art HDR models evaluated on MNIST. Achieving an

accuracy of 99.80%, the proposed ensemble surpasses previously reported results, validating the benefits of integrating complementary CNN feature spaces within a multinomial logistic regression meta-learning framework.

Table 5. Performance comparison with state-of-the-art HDR models on MNIST.

Study	Approach	Accuracy (%)
Mohammed and Kora (2023)	DCNNs	97.82
Singh et al. (2023)	VWCRF	99.40
Chen et al. (2020)	RNN	96.50
Gupta and Bag (2021)	VWCRF	96.23
Absur et al. (2024)	Customized CNNs	98.71
Our Work	Stacked Meta-Learner	**99.80**

The following key observations emerge from the results:

(1) **Stable Cross-Validation Performance** : All base CNNs achieved high mean accuracy with minimal fold-to-fold variation (Section 4.3.2). However, the meta-learner further reduced variance and maintained a tight train–validation alignment, as shown in Figure 10, indicating improved generalization.
(2) **Consistent Per-Class Accuracy**: The per-class evaluation (Table 4) confirmed that no digit class exhibited disproportionate sensitivity to data partitioning. Standard deviations below 0.004 across all classes illustrate exceptional robustness, a key advantage for real-world deployment where handwriting variability is inevitable.
(3) **Ensemble Superiority Over Individual Backbones**: As summarized in Figure 11 and Table 3, the stacked model consistently outperformed each individual backbone, validating the hypothesis that combining models with diverse architectural biases (ResNet50 for global structure, VGG16 for local detail, VGG19 for deeper hierarchical features) enhances overall recognition capability.
(4) **Trade-off Between Accuracy and Computational Cost**: While the ensemble incurs higher computational cost than any single backbone (Section 4.3.1), the gain in accuracy (*e.g.*, about +0.24 to +0.52 percentage points over single models on mean CV accuracy) and the stability gains justify the additional training time for applications prioritizing recognition reliability.

Overall, the combination of **high mean accuracy**, **low variance**, and **balanced per-class performance** confirms that the proposed architecture achieves state-of-the-art HDR performance while maintaining robustness across folds. Although MNIST is a relatively simple dataset, the ensemble's stable learning dynamics and low variance suggest strong potential for extension to more challenging datasets such as EMNIST, a direction we explore in future work.

5. CONCLUSION AND FUTURE WORK

This study presents a robust stacked ensemble framework for HDR that effectively combines the complementary strengths of ResNet50, VGG16, and VGG19. By leveraging TL, targeted fine-tuning, and penultimate-layer feature extraction, the proposed meta-learning stage, implemented via multinomial logistic regression, achieves both exceptional accuracy and strong generalization capability. Our results on the MNIST benchmark confirm that the ensemble consistently outperforms individual backbones, attaining a state-of-the-art accuracy of 99.80% under a rigorous 5-fold cross-validation protocol.

The principal contributions of this work are threefold:

- **Heterogeneous Ensemble Design:** Integration of three architecturally diverse CNNs to capture a richer set of discriminative features.
- **Strategic Feature Extraction:** Systematic use of penultimate-layer outputs to retain high-level semantic patterns while reducing overfitting risk.
- **Lightweight yet Effective Meta-Learner:** Adoption of multinomial logistic regression to fuse deep feature representations efficiently, ensuring low complexity and interpretability.

Beyond surpassing previous MNIST benchmarks, the proposed method demonstrates stability across classes, minimal variance across folds, and favorable computational trade-offs, thereby establishing a solid foundation for scaling to more demanding recognition tasks.

Although MNIST serves as a well-established benchmark, it is relatively simple compared to real-world scenarios. Future directions include:

- Extending the evaluation to more challenging datasets such as EMNIST, Fashion-MNIST, or multi-script handwriting corpora.
- Incorporating additional backbone architectures (e.g., EfficientNet, DenseNet) to further enrich feature diversity.
- Exploring dimensionality reduction and feature selection techniques to improve inference speed for deployment on resource-constrained devices.
- Investigating adaptive weighting mechanisms in the meta-learner to dynamically prioritize features from the most relevant backbone for each input instance.

In conclusion, this work not only establishes a high-performing ensemble approach for handwritten digit recognition but also paves the way for broader applications of hybrid deep learning architectures in complex, real-world pattern recognition challenges.

ACKNOWLEDGEMENTS

The authors extend their appreciation to Northern Border University, Saudi Arabia, for supporting this work through project number (NBU-CRP-2025-1564)

CODE AVAILABILITY

The source code supporting the findings of this study is openly available at: `Complete implementation of the stacked CNN ensemble model (ResNet50, VGG16, VGG19).`

REFERENCES

Absur, M. N., Nasif, K. F. A., Saha, S., & Nova, S. N. (2024, July). Revolutionizing image recognition: Next-generation CNN architectures for handwritten digits and objects. In *2024 IEEE Symposium on Wireless*

Technology & Applications (ISWTA) (pp. 173–178). IEEE.

Agarap, A. F. (2018). Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*.

Ali, S., Sahiba, S., Azeem, M., Shaukat, Z., Mahmood, T., Sakhawat, Z., & Aslam, M. S. (2023). A recognition model for handwritten Persian/Arabic numbers based on optimized deep convolutional neural network. *Multimedia Tools and Applications*, 82(10), 14557–14580.

Amara, M., Smairi, N., Mnasri, S., & Zidouri, A. (2024). Revitalizing Arabic Character Classification: Unleashing the Power of Deep Learning with Transfer Learning and Data Augmentation Techniques. *Arabian Journal for Science and Engineering*, 1–25.

Ashiquzzaman, A., & Tushar, A. K. (2017, February). Handwritten Arabic numeral recognition using deep learning neural networks. In *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 1–4). IEEE.

Ashiquzzaman, A., Tushar, A. K., Rahman, A., & Mohsin, F. (2019). An efficient recognition method for handwritten Arabic numerals using CNN with data augmentation and dropout. In *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2018, Volume 1* (pp. 299–309). Springer.

Azawi, N. (2023). Handwritten digits recognition using transfer learning. *Computers and Electrical Engineering*, 106, 108604.

Castro, E., Cardoso, J. S., & Pereira, J. C. (2018). Elastic deformations for data augmentation in breast cancer mass detection. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)* (pp. 230–234). IEEE.

Chen, M. R., Chen, B. P., Zeng, G. Q., Lu, K. D., & Chu, P. (2020). An adaptive fractional-order BP neural network based on extremal optimization for handwritten digits recognition. *Neurocomputing*, 391, 260–272.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). IEEE.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems* (pp. 1–15). Springer.

Gupta, D., & Bag, S. (2021). CNN-based multilingual handwritten numeral recognition: A fusion-free approach. *Expert Systems with Applications*, 165, 113784.

Guo, J., Wei, W., Ma, Y., & Peng, C. (2021). A fast and accurate object detector for handwritten digit string recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 787–794). IEEE.

Kaur, K., Dhir, R., & Kumar, K. (2021). Transfer learning approach for analysis of epochs on handwritten digit classification. In *ICSCCC 2021* (pp. 456–458). IEEE.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with ensemble accuracy. *Machine Learning*, 51(2), 181–207.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

Mascarenhas, S., & Agarwal, M. (2021). A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for image classification. In *CENTCON 2021* (pp. 96–99). IEEE.

Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University–Computer and Information Sciences*.

Patil, P. (2020). Handwritten digit recognition using various machine learning algorithms and models. *IJIRCST*, 8(3), 2347–5552.

Rasheed, A., Ali, N., Zafar, B., Shabbir, A., Sajid, M., & Mahmood, M. T. (2022). Handwritten Urdu characters and digits recognition using transfer learning and augmentation with AlexNet. *IEEE Access*, 10, 102629–102645.

Reza, S., Amin, O. B., & Hashem, M. M. A. (2019). Basic to compound: A novel transfer learning approach for Bengali handwritten character recognition. In *ICBSLP 2019* (pp. 1–5). IEEE.

Singh, D., Bano, S., Samanta, D., Mekala, M. S., & Islam, S. H. (2023). Deep learning inspired nonlinear classification methodology for handwritten digits recognition using DSR encoder. *Arabian Journal for Science and Engineering*, 48(2), 1385–1397.

Wen, L., Li, X., & Gao, L. (2020). A transfer convolutional neural network for fault diagnosis based on ResNet-50. *Neural Computing and Applications*, 32, 6111–6124.

Zhang, L., Bian, Y., Jiang, P., & Zhang, F. (2023). A transfer residual neural network based on ResNet-50 for detection of steel surface defects. *Applied Sciences*, 13(9), 5260.