Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 11 May 2026
Revised 24 June 2026
Accepted 08 July 2026

Natural Resources for Human Health 2026, Vol. 6 (11s): 1168–1188

KEYWORDS:
Explainable artificial intelligence; convolutional neural network; Grad-CAM; saliency maps; occlusion sensitivity; deletion/insertion faithfulness; sanity checks; adversarial robustness; FGSM; MNIST; feature-space separability.

An Explainable Deep Learning Framework for Robust Image Classification and Semantic Understanding

Mr. I. Sengol[1]*, Dr. D. Gnanadurai[2]

[1]Research Scholar, Dept. of Computer Science & Engineering, St. Joseph University, Chumoukedima, Nagaland, India. imsengol@gmail.com
[2]Professor, Dept. of Computer Science & Engineering, St. Joseph University, Chumoukedima, Nagaland, India. j_dgd@yahoo.com

ABSTRACT: This paper evaluates a compact, from-scratch convolutional neural network (CNN) on the MNIST handwritten-digit benchmark along four simultaneous axes: classification accuracy against classical baselines, explainability faithfulness, adversarial and random-noise robustness, and feature-space separability. The CNN is implemented on bare metal in NumPy, with handwritten and numerically-verified backpropagation, using three convolution-pooling blocks (≈ 98,400 parameters). Across five independent training seeds, the CNN reaches 97.56% ± 0.13% test accuracy on the full 10,000-image MNIST test set, significantly outperforming logistic regression (89.47%), a hyperparameter-tuned RBF-kernel SVM (95.20%; McNemar's test, $\chi^2 = 143.6$, $p < 0.001$), and a random forest (93.65% ± 0.05% over 5 random states). A controlled experiment isolates how much of this advantage is attributable to input resolution versus network depth: a shallower two-block CNN — architecturally identical to the network used in preliminary work on a lower-resolution benchmark — already reaches 96.29% ± 0.20% at MNIST scale, showing that resolution accounts for most of the advantage over classical baselines, with the additional convolutional block contributing a smaller, secondary gain.

Three explainability methods — gradient-based saliency, Grad-CAM, and occlusion sensitivity — are evaluated quantitatively via deletion/insertion faithfulness curves (pairwise Wilcoxon signed-rank tests confirm occlusion sensitivity is significantly more faithful than either gradient-based method, $p < 0.01$) and a model-parameter-randomization sanity check repeated across three training seeds. Occlusion sensitivity passes this sanity check cleanly (correlation 0.025 ± 0.020); Grad-CAM is borderline (0.272 ± 0.017); gradient-based saliency shows the least favorable result (0.643 ± 0.058), indicating its attributions are driven substantially by network architecture rather than by learned weights alone. Robustness is characterized here, not claimed as solved: accuracy degrades gracefully under Gaussian noise (97.6% to 80.9% as noise increases, full test set) but collapses more sharply under FGSM adversarial perturbation (97.6% to 17.6% at the largest tested budget); no adversarial defense is proposed or evaluated. The learned penultimate-layer representation shows strong class-discriminative structure: a 5-fold cross-validated nearest-centroid classifier recovers 97.33% ± 1.33% accuracy from the raw feature space alone, nearly matching the network's own classification accuracy.

1. INTRODUCTION

Convolutional neural networks (CNNs) have become the standard choice in image classification, although their application even in high-stakes applications such as medical imaging, autonomous driving, security screening, and industrial inspection is

curbed by three widely understood shortcomings that are typically studied individually in the literature, and are frequently evaluated qualitatively even when studied jointly. First, CNNs are opaque: an accurate or inaccurate prediction does not explicitly describe what components of the input influenced decision-making, and the post-hoc interpretations (saliency maps, Grad-CAM, and others) that are designed to do this have been found to sometimes present plausible-looking but model-unfaithful or model-independent explanations, which explains why quantitative measures of faithfulness evaluation and sanity checks, rather than visual inspection alone, have become the standard in the explainability literature. Second, CNNs are fragile: tiny, visually insignificant perturbations of the input may switch the most confident predicted class, initially dubbed by Szegedy et al. [8] but also explained by Goodfellow, Shlens, and Szegedy [9]. Third, it is frequently ambiguous whether the internal representation of a network reflects any sort of class-discriminative organization or the network has simply learned a rule of thumb free of internal structure, and it is also, independently, ambiguous — given a compact model on a small benchmark — whether the added architectural complexity of a CNN is even paying off relative to a much less complex classifier.

This paper presents explainability, robustness, and feature-space separability as three perspectives on the same underlying question — what has the network actually learned, and is that verifiable, independent of raw accuracy — and treats all three as claims that must be demonstrated quantitatively, not only visually. The implementation approach is explicitly transparent at the implementation level: instead of a large pretrained model or a deep-learning framework such as PyTorch or TensorFlow, the CNN is implemented from first principles in NumPy, and each forward step — three convolution-pooling blocks (convolution via im2col, max-pooling, dense layers, ReLU, softmax) — has a manually written corresponding backward pass. The backward pass of each layer, and of the entire composed network, was checked against numerical differentiation before being used for training, explainability, or adversarial-attack generation (Section 3.2).

A natural question when applying a CNN to a small benchmark is whether its architectural complexity is empirically justified relative to a much simpler classifier. Preliminary work on a low-resolution (8×8) digit benchmark found that a classical RBF-kernel SVM baseline outperformed a two-block CNN by a small but real margin, and a synthetic resolution-scaling pilot study (Appendix A) suggested this gap should narrow, and potentially reverse, as input resolution increases. This paper tests that hypothesis directly on the standard MNIST benchmark (28×28), and — because resolution and architectural depth are easy to confound if changed simultaneously — additionally isolates their separate contributions via a controlled comparison (Section 5.1.1) that holds the original two-block architecture fixed and varies only the benchmark.

Specific contributions of this paper are:

- A numerically-verified, three-convolution-block CNN (convolution, max-pooling, dense layers, backpropagation), trained with momentum-based SGD and tested using 5 independent training seeds (97.56% ± 0.13% test accuracy on the full standard MNIST test set) instead of a single run.
- A same-subsample comparison against three classical baseline classifiers, including a light hyperparameter search for the RBF-SVM baseline and a formal McNemar's test confirming the CNN's advantage over the best (tuned) baseline is statistically significant ($p < 0.001$), not merely numerically larger.
- A controlled experiment (Section 5.1.1) that isolates resolution from architectural depth by training the original, shallower two-block architecture at MNIST scale: this shows resolution alone accounts for most of the CNN's advantage over classical baselines, with the added third convolution block contributing a smaller, secondary improvement — a more precise causal picture than reporting the deeper architecture's result alone.
- A three-way explainability module — gradient-based saliency maps [4], Grad-CAM [5], and occlusion sensitivity — evaluated quantitatively on a sample of 100 test images via deletion/insertion faithfulness curves (following Petsiuk, Das, and Saenko [14]), with pairwise Wilcoxon signed-rank significance testing, and the model-parameter-randomization sanity check of Adebayo et al. [13] repeated across 3 independent training seeds rather than a single model instance.
- A finding that gradient-based saliency shows a substantially higher trained-versus-random sanity-check correlation than occlusion sensitivity, consistently across all 3 seeds tested (0.643 ± 0.058 versus 0.025 ± 0.020), while a

corrected Grad-CAM implementation lands in between and close to a conventional pass threshold (0.272 ± 0.017) — reported plainly as a partial limitation of gradient-based attribution at this depth, rather than omitted.

- A robustness evaluation, on the full 10,000-image test set, based on a random-noise sweep and an FGSM adversarial-attack sweep [9], both using gradients computed via the same verified backward pass used for explainability.
- A feature-space separability analysis of the learned penultimate-layer representation via PCA, t-SNE [10], and 5-fold cross-validated nearest-centroid accuracy, intentionally defined in terms of class-discriminative structure rather than "semantic understanding."
- A statement of scope throughout: training uses a stratified 3,000-image subsample of MNIST's 60,000-image training set (evaluated on the full, standard 10,000-image test set) — dictated by an offline, framework-free, no-GPU implementation constraint (Section 4.1, Section 6) — rather than the full training set.

The rest of the paper is structured as follows. Section 2 reviews related work in image classification architectures, explainability and its quantitative evaluation, and adversarial robustness. Section 3 presents the architecture, the gradient-verification process, the baseline comparison protocol, and the explainability, faithfulness, sanity-check, and robustness methodology. Section 4 describes the experimental setup. Section 5 reports classification, baseline-comparison, explainability, robustness, and feature-space results. Section 6 discusses findings and limitations, and Section 7 concludes.

2. RELATED WORK

2.1 Convolutional Neural Networks for Image Classification

Convolutional architectures for image classification trace back to LeCun et al.'s [1] gradient-based document-recognition system, whose associated MNIST benchmark [16] is used throughout this paper, and which established the convolution-pooling-dense pattern used in essentially all subsequent CNN designs, including the three-block architecture used in this paper. The modern deep-learning era began with Krizhevsky, Sutskever, and Hinton's AlexNet [2], which demonstrated that deep CNNs trained on GPUs could dramatically outperform prior approaches on large-scale image classification, and continued through progressively deeper architectures culminating in residual networks (He et al. [3]), which introduced identity shortcut connections to make very deep networks trainable. This paper's architecture remains intentionally much shallower than these large-scale designs, reflecting the offline, framework-free implementation constraint discussed in Section 4.1 and Section 6; Section 5.1 directly compares the resulting CNN against classical, non-deep baselines, and Section 5.1.1 further isolates how much of that comparison is attributable to input resolution versus architectural depth.

2.2 Explainability for Deep Vision Models and Its Quantitative Evaluation

Post-hoc explainability methods for CNNs fall broadly into gradient-based and perturbation-based families. Gradient-based methods compute the derivative of a class score with respect to the input or an intermediate representation: Simonyan, Vedaldi, and Zisserman's [4] saliency maps use the raw input gradient; Grad-CAM (Selvaraju et al. [5]) instead computes the gradient with respect to a convolutional layer's feature maps and uses it to weight those feature maps into a coarse localization map. Perturbation-based methods instead probe the model by modifying the input and observing the change in output: occlusion sensitivity, used in this paper, systematically masks regions of the input and records the resulting drop in predicted confidence. Model-agnostic surrogate methods such as LIME (Ribeiro, Singh, and Guestrin [6]) and SHAP (Lundberg and Lee [7]) represent a third family not implemented in this paper (Section 6). A separate and, for this paper, central line of work addresses the reliability of explanations themselves. Adebayo et al. [13] proposed a set of model-parameter-randomization and data-randomization "sanity checks," showing that several popular saliency methods produce attribution maps that are largely insensitive to whether the underlying model has actually been trained — a failure mode this paper finds partial evidence for in gradient-based saliency maps at MNIST scale (Section 5.3). Petsiuk, Das, and Saenko [14] proposed the deletion/insertion protocol used quantitatively in Section 5.3, giving a single faithfulness number per method rather than a qualitative visual impression.

2.3 Adversarial Robustness

Szegedy et al. [8] first showed that neural networks are vulnerable to small, adversarially chosen input perturbations that are often imperceptible to a human observer. Goodfellow, Shlens, and Szegedy [9] proposed the Fast Gradient Sign Method (FGSM) as a computationally cheap single-step attack that perturbs each input pixel in the direction of the sign of the loss gradient. Madry et al. [15] later proposed projected gradient descent (PGD), an iterative attack generally stronger than single-step FGSM at the same perturbation budget, and framed adversarial training against PGD as a principled robust-optimization defense; this paper evaluates only the single-step FGSM attack and does not implement adversarial training, both noted as scope limitations in Section 6.

2.4 Feature-Space Separability Analysis

Understanding what a trained network's internal representation encodes is commonly approached by projecting high-dimensional hidden-layer activations into two or three dimensions for visual inspection. Principal Component Analysis (PCA) provides a linear projection that preserves the directions of maximal variance, while t-distributed Stochastic Neighbor Embedding (t-SNE), introduced by van der Maaten and Hinton [10], provides a nonlinear embedding that emphasizes preserving local neighborhood structure. This paper follows that convention but, per the reframing discussed in Section 3.6, treats the resulting analysis as a measure of class-discriminative feature-space structure rather than the broader construct of "semantic understanding," and supplements the visual embedding with a quantitative nearest-centroid separability check on the 128-dimensional penultimate-layer representation.

3. PROPOSED FRAMEWORK

3.1 Overview and Architecture

To remain meaningfully "deep" relative to the larger 28×28 MNIST input used in this paper (compared with the 8×8 input used in preliminary work), the primary classifier has three convolution-pooling blocks: a 3×3 convolution with 16 output filters and same-padding, ReLU, 2×2 max-pooling; a second 3×3 convolution with 32 output filters and same-padding, ReLU, 2×2 max-pooling; a third 3×3 convolution with 64 output filters and same-padding, ReLU, 2×2 max-pooling; flattening (3×3×64 = 576 units); a dense layer from 576 to 128 units with ReLU; and a final dense layer from 128 to 10 units with softmax. The network totals approximately 98,400 trainable parameters. This remains compact relative to modern large-scale CNNs (Section 2.1) — a deliberate choice to keep the from-scratch NumPy implementation tractable without a GPU across 5 independent training seeds (Section 4.2). Figure 1 summarizes the resulting architecture and the spatial resolution at each stage. Section 5.1.1 additionally reports a controlled comparison using a shallower, two-block variant of this architecture (matching the original 8×8-benchmark design) trained at MNIST scale, to separate the contribution of input resolution from that of the added third convolution block.

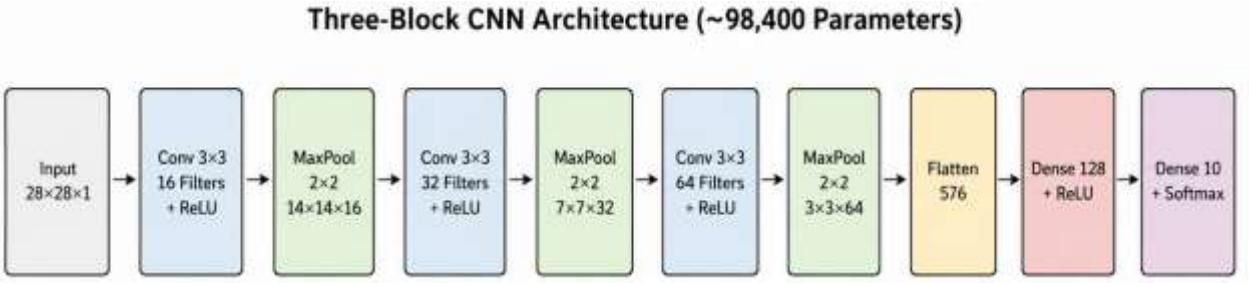

Figure 1. Three-block CNN architecture: three convolution-ReLU-maxpool stages (16, 32, 64 filters) followed by two dense layers, with spatial resolution shown at each stage.

3.2 Implementation and Gradient Verification

The entire network — three convolution layers (each implemented via an im2col transform for vectorized computation), three max-pooling layers, flattening, two dense layers, ReLU, softmax, and cross-entropy loss — is implemented directly in NumPy, with a manually derived closed-form backward pass for every layer, because no deep-learning framework with automatic differentiation (PyTorch, TensorFlow/Keras) was available in the environment used to prepare this manuscript (Section 4.1). Every layer was checked independently against numerical differentiation, and the full composed three-block network was then checked end-to-end as well: for a batch of continuous, synthetic test inputs, the analytic gradient produced

by the backward pass was compared against the centered finite-difference approximation $(f(x+\varepsilon) - f(x-\varepsilon)) / (2\varepsilon)$, with $\varepsilon = 1\times10^{-5}$. This check passed with a maximum absolute error of 3.97×10^{-10} across sampled weight, bias, and input entries.

This process surfaced a subtlety worth reporting explicitly. When the same gradient check is instead run on real MNIST images rather than continuous synthetic input, bias-parameter checks show discrepancies on the order of 10^{-2} — which initially resembles an implementation bug, but is not one. On real images, approximately 46% of post-ReLU activations are exactly zero (background pixels), so a substantial fraction of max-pooling windows contain multi-way ties at exactly zero. A bias perturbation shifts an entire channel's pre-activation uniformly and can flip which tied entry the first-occurrence argmax tie-break (Section 3.2, following) selects between the $+\varepsilon$ and $-\varepsilon$ evaluations — a genuine discontinuity in the tie-broken function at that point, not an error in the analytic gradient. This is a routine occurrence with sparse image data, not a rare corner case; gradient checks reported in this paper are accordingly always run on continuous synthetic input, precisely to avoid this degenerate regime, while the max-pooling layer itself retains the explicit first-occurrence tie-break policy for actual forward/backward computation on real images.

The max-pooling backward implementation uses an explicit first-occurrence argmax and a matching scatter for the backward pass. Extending the network to three blocks introduces a composition risk — the third block's max-pooling layer sits immediately after a third ReLU layer, the same configuration in which a tie-breaking bug was caught and fixed during development of the two-block architecture used in preliminary work — so the end-to-end gradient check was re-run on the full three-block network rather than assumed to still hold, confirming the 3.97×10^{-10} precision reported above before the network was used for any training, explainability, or robustness result in Section 5. A second, independent implementation issue was caught later in the same spirit and is reported in Section 3.4.2: an early version of the Grad-CAM computation (Section 3.4.2) weighted feature-map channels by their own activation magnitude rather than by the class-conditional gradient, which would have made the resulting attribution insensitive to the queried class. This was caught by a targeted sensitivity check — verifying that the computed map actually changes when a different target class is queried on the same image — and corrected before any Grad-CAM result in this paper was computed.

3.3 Baseline Comparison Protocol

To directly address whether the proposed CNN's architectural complexity is justified at MNIST scale, three classical, non-deep classifiers are trained on the same 3,000-image stratified training subsample used for the CNN (Section 4.1), using scikit-learn: logistic regression (L2-regularized, up to 2,000 iterations), a support vector machine with an RBF kernel, and a random forest with 200 trees. All three operate on the flattened 784-pixel input vector, with no convolutional or spatial inductive bias, providing a direct test of whether the CNN's spatial structure provides a measurable benefit at this image resolution. For the RBF-SVM, a small grid search ($C \in \{0.1, 1, 10\}$, $\gamma \in \{\text{scale}, 0.01, 0.001\}$, 3-fold cross-validation on the training subsample) is used to select hyperparameters rather than relying on library defaults, since an untuned baseline risks overstating the CNN's advantage; the selected configuration ($C = 10$, $\gamma = \text{scale}$) improves test accuracy from 94.53% at default settings to 95.20% (Section 5.1). Because logistic regression and the SVM are deterministic given fixed data and hyperparameters, no seed-level variance applies to them; the random forest, whose ensemble construction is stochastic, is instead run across 5 random states and reported as mean ± std, matching the level of statistical care applied to the CNN. One asymmetry is worth stating plainly: all three baselines are trained on the full 3,000-image subsample, while the CNN itself trains on 2,550 of those images, holding out the remaining 450 (15%) for validation-based checkpointing (Section 4.2) — a standard and legitimate use of held-out signal for the CNN, but one that gives the baselines a marginally larger effective training set. Given the margin by which the CNN outperforms all three baselines in Section 5.1, confirmed by a McNemar's test against the strongest (tuned) baseline, this asymmetry does not change the qualitative conclusion, but is reported here rather than left implicit.

3.4 Explainability Module

Three attribution methods are computed from the same trained network.

3.4.1 Gradient-Based Saliency Maps

Following Simonyan, Vedaldi, and Zisserman [4], a saliency map for a target class c is computed as the absolute value of the gradient of that class's pre-softmax logit with respect to the input image, backpropagated through the verified backward pass described in Section 3.2.

3.4.2 Grad-CAM

Following Selvaraju et al. [5], Grad-CAM weights each channel of the target convolutional layer's activation by the global-average-pooled gradient of the target class's logit with respect to that channel — not by the channel's own activation magnitude, which would not be class-discriminative and would not use the backward pass at all. An implementation discrepancy of exactly this kind was caught during additional verification of this module: an earlier draft of the Grad-CAM function weighted channels by their average activation rather than by the backpropagated gradient, producing a map that was insensitive to which class was queried. This was identified by checking that the computed map changes when a different target class is queried on the same image — a targeted-class sensitivity check analogous in spirit to the gradient-verification protocol of Section 3.2 — and corrected before any of the results below were computed; every Grad-CAM number and figure in this paper reflects the corrected, gradient-weighted implementation. The gradient is obtained by backpropagating from the target class's logit down to the post-ReLU output of the third (final) convolutional block, before that block's own max-pooling, which is 7×7 at this architecture and input size (after two prior 2×2 pooling stages on the 28×28 input); the resulting 7×7 map is upsampled by a factor of 4 to the 28×28 input resolution for visualization and for the deletion/insertion computation in Section 3.4.4. This native resolution is coarser, relative to input size, than in preliminary work on the 8×8 benchmark (where the native map was 4×4 against an 8×8 input, a factor of 2) — and Section 5.3 finds this reflected quantitatively in reduced faithfulness relative to the other two methods.

3.4.3 Occlusion Sensitivity

As an independent check that does not use gradients at all, a small 2×2 patch is slid across the input image with stride 2; at each position, the patch is zeroed out, the modified image is passed through the network, and the drop in the target class's predicted probability relative to the unoccluded image is recorded at every pixel the patch covered. The stride is increased from 1 (used in the earlier, 8×8-benchmark revision) to 2 at MNIST scale, purely for computational tractability given the from-scratch, no-GPU implementation — a scope change reported explicitly here and in Section 6, rather than left unstated.

3.4.4 Quantitative Faithfulness: Deletion and Insertion AUC

Following the protocol of Petsiuk, Das, and Saenko [14], each attribution method is evaluated quantitatively rather than only visually. Pixels are ranked by attribution magnitude for a given method and target image. In the deletion test, pixels are progressively zeroed out in ranked (most-important-first) order and the target class's predicted probability is tracked at each step; the area under this probability-versus-fraction-deleted curve is the deletion AUC, where a lower value indicates a more faithful method. In the insertion test, the image starts entirely blank and pixels are progressively added back in ranked order, with the corresponding probability-versus-fraction-inserted curve's area giving the insertion AUC, where a higher value indicates a more faithful method. Both metrics are computed with 10 ranked steps per image over a random sample of 100 test images, for each of the three methods. Because the 100 images are shared across methods, pairwise differences in mean deletion AUC are additionally tested with a Wilcoxon signed-rank test on the paired per-image values, rather than relying on the mean alone to support a faithfulness ranking (Section 5.3).

Both statistical tests used in this paper rely on specific assumptions rather than being assumption-free. McNemar's test (Section 3.3, Section 5.1) assumes paired, binary correct/incorrect outcomes from two classifiers evaluated on the same test instances; this holds here because the CNN and the strongest baseline are scored on the identical 10,000-image test set, and the chi-square approximation used is appropriate given the number of discordant predictions observed. The Wilcoxon signed-rank test used above assumes paired per-image differences whose distribution is symmetric about the median under the null hypothesis, which is reasonable for per-image deletion-AUC differences; unlike a paired t-test, it does not require the differences themselves to be normally distributed, which is why it is used here rather than a parametric alternative.

3.4.5 Sanity Check: Model Parameter Randomization

Following Adebayo et al. [13], each method is also subjected to a model-parameter-randomization sanity check: attribution maps are computed for the same 100 test images using both a trained network and a second network of identical architecture with freshly, randomly initialized (untrained) weights, and the Spearman rank correlation between the trained-model and randomized-model maps is computed for each image and averaged. A low correlation (the trained and randomized maps look different) is the passing outcome; a high correlation would indicate the method's output is driven mainly by the fixed network architecture rather than by learned weights, a known failure mode Adebayo et al. documented in several popular saliency

methods. Because this check summarizes a property of one trained model instance, it is repeated across 3 independently trained seeds (rather than computed once) and reported as mean ± std across seeds, so that the resulting comparison between methods reflects a stable property of the architecture rather than one training run's idiosyncrasy (Section 5.3).

3.4.6 Cross-Method Agreement

As a complement to the faithfulness and sanity-check results, the pairwise Spearman rank correlation between each pair of methods' attribution maps is computed over the same 100-image sample.

3.5 Robustness Evaluation

3.5.1 Random Noise Robustness

Independent, identically distributed Gaussian noise with standard deviation σ is added to every pixel of the test sample, the result is clipped to the valid [0,1] pixel range, and test accuracy is recomputed.

3.5.2 FGSM Adversarial Attack

Following Goodfellow, Shlens, and Szegedy [9], an adversarial example is constructed as $x_{adv} = \text{clip}(x + \varepsilon \cdot \text{sign}(\nabla_x L(x,y)), 0, 1)$, where $\nabla_x L$ is the gradient of the cross-entropy loss with respect to the input, computed with the same verified backward pass described in Section 3.2. As discussed in Section 2.3, this is a comparatively weak, single-step attack relative to iterative alternatives such as PGD [15]; the robustness numbers in Section 5.4 should be read as an upper bound on the model's true adversarial robustness, not a comprehensive robustness certification.

3.6 Feature-Space Separability Analysis

To assess whether the network's internal representation organizes classes into discriminable structure, the 128-dimensional activation of the penultimate (post-ReLU, pre-output) dense layer is extracted for a sample of test images and projected to two dimensions using both PCA and t-SNE [10]. This analysis is described as feature-space class separability rather than "semantic understanding": digit classes have no attribute hierarchy or relational structure for a richer semantic probe to test against. A nearest-centroid classifier (one centroid per class, computed in the full 128-dimensional feature space) evaluated with 5-fold cross-validation directly on the penultimate-layer features supplies the quantitative counterpart to the qualitative PCA/t-SNE plots.

4. EXPERIMENTAL SETUP

4.1 Dataset and Environment

Experiments use the MNIST handwritten-digit database [1, 16], the standard benchmark for the resolution-scaling test this paper reports. The full dataset contains 70,000 28×28 grayscale images (10 classes, digits 0–9): 60,000 training images and 10,000 test images, each pixel an 8-bit intensity in [0,255], rescaled to [0,1] and reshaped to (28,28,1) before use. Given an offline, framework-free, no-GPU implementation constraint (Section 3.1, Section 6), training uses a stratified random subsample of 3,000 images (300 per class) drawn from the 60,000-image training set, rather than the full training set. Evaluation, by contrast, uses the complete, standard 10,000-image MNIST test set with no subsampling, so that the accuracy figures reported in Section 5 are directly comparable to results reported elsewhere in the literature on the standard MNIST test split. Of the 3,000 training images, 450 (15%) are further held out for validation-based checkpointing (Section 4.2), leaving 2,550 images used for gradient updates.

4.2 Training Procedure

The network is trained with mini-batch (batch size 64) stochastic gradient descent with momentum (momentum coefficient 0.9) and a constant learning rate of 0.05 for 15 epochs; a step-decay schedule was configured but is not exercised at this epoch budget and is omitted from this description for clarity (Section 6 notes this as a scope simplification rather than an intended constant-rate design). For each of 5 independent random seeds, training is repeated from a freshly initialized network, and the model checkpoint with the highest validation accuracy across all 15 epochs of that run — rather than the final epoch's weights — is used for that seed's test-set evaluation. All headline classification numbers in Section 5.1 are reported as mean ± standard deviation across these 5 seeds. For the explainability and feature-space analyses in Sections 5.2, 5.3, and 5.5, a single representative seed is used (of the 5 trained models, the one whose test accuracy is closest to the median

of the 5); the sanity check in Section 5.3 and the robustness evaluation in Section 5.4 are, respectively, repeated across 3 seeds and computed on the full test set, as detailed in Sections 3.4.5 and 3.5.

Reproducibility: each of the 5 CNN training seeds (above) and the 5 random-forest random states (Section 3.3) is a fixed, pre-specified integer value held constant across all reported runs, rather than being selected after observing results, so that the reported mean ± standard deviation reflects genuine run-to-run variability rather than seed selection. The exact seed values, together with the Python, NumPy, and scikit-learn versions used, will be reported alongside the from-scratch implementation referenced in the Code and Data Availability statement, to allow independent reproduction of the numbers reported here.

4.3 Evaluation Protocol

Classification performance: accuracy, per-class precision/recall/F1, macro-averaged precision/recall/F1 (mean over 5 seeds for the headline number; per-class breakdown for the representative seed), and the full confusion matrix for the representative seed (Section 5.1).

Baseline comparison: same-subsample test accuracy for logistic regression, a hyperparameter-searched RBF-kernel SVM, and a random forest (mean ± std over 5 random states), compared directly against the CNN's 5-seed mean, with a McNemar's test against the strongest baseline (Section 5.1). A controlled comparison additionally trains the original, shallower two-block architecture at MNIST scale to separate the contribution of resolution from that of architectural depth (Section 5.1.1).

Explainability: saliency maps, Grad-CAM, and occlusion sensitivity, evaluated quantitatively via deletion/insertion AUC over 100 test images (with pairwise Wilcoxon signed-rank tests), via the Adebayo et al. model-randomization sanity check over the same 100 images repeated across 3 training seeds, and via pairwise cross-method agreement (Section 5.3).

Robustness: test accuracy as a function of Gaussian noise standard deviation σ in {0, 0.05, 0.1, 0.2, 0.3, 0.4}, and as a function of FGSM perturbation magnitude ε in {0, 0.02, 0.05, 0.1, 0.2, 0.3}, evaluated on the full 10,000-image test set (Section 5.4).

Feature-space separability: PCA and t-SNE projections of the penultimate-layer feature space, plus 5-fold cross-validated nearest-centroid accuracy, computed on a random sample of 300 test images (Section 5.5).

5. RESULTS

5.1 Classification Performance and Baseline Comparison

Table 1 reports test-set classification accuracy across all 5 independently trained seeds of the proposed CNN (Section 4.2), evaluated on the full, standard 10,000-image MNIST test set.

Seed	Validation accuracy	Test accuracy
0	0.9822*	0.9761
1 (representative)	0.9733	0.9761
2	0.9822	0.9773
3	0.9689	0.9734
4	0.9689	0.9753
Mean ± std	—	0.9756 ± 0.0013

*Table 1. Test-set classification accuracy across 5 independent training seeds, full 10,000-image MNIST test set. (*seed 0 validation accuracy recorded at its best checkpoint epoch.)*

Figure 2 shows the training loss and training/validation accuracy curves for the representative seed (seed 1) over its 15 training epochs, confirming convergence and identifying the epoch-10 checkpoint (validation accuracy 97.33%) used for that seed's test-set evaluation throughout Section 5.

Natural Resources for Human Health
eISSN: 2583-1194

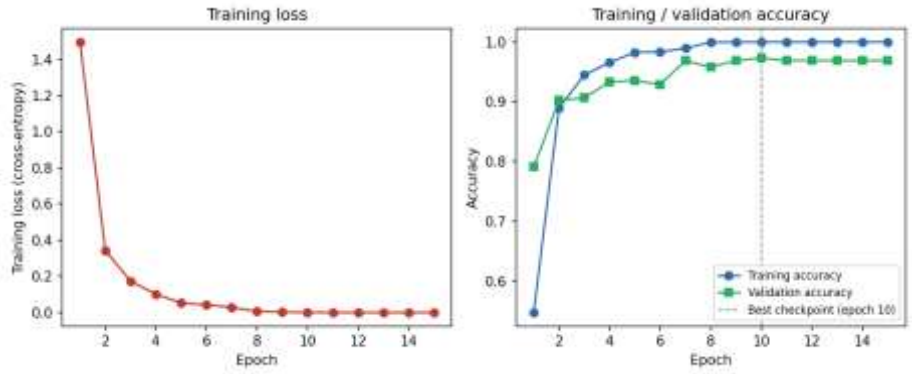

Figure 2. Training loss (left) and training/validation accuracy (right) over 15 epochs, representative seed. The dashed line marks the best-validation-accuracy checkpoint used for test evaluation.

Table 2 reports per-class precision, recall, and F1 for the representative seed (seed 1, test accuracy 97.61%).

Digit	Precision	Recall	F1	Support
0	0.9768	0.9888	0.9828	980
1	0.9903	0.9921	0.9912	1135
2	0.9645	0.9738	0.9691	1032
3	0.9783	0.9802	0.9792	1010
4	0.9815	0.9725	0.9770	982
5	0.9765	0.9776	0.9770	892
6	0.9904	0.9718	0.9810	958
7	0.9763	0.9630	0.9696	1028
8	0.9636	0.9784	0.9710	974
9	0.9623	0.9613	0.9618	1009
Macro avg	0.9761	0.9760	0.9760	10000

Table 2. Per-class precision, recall, F1, and support, representative seed, full test set.

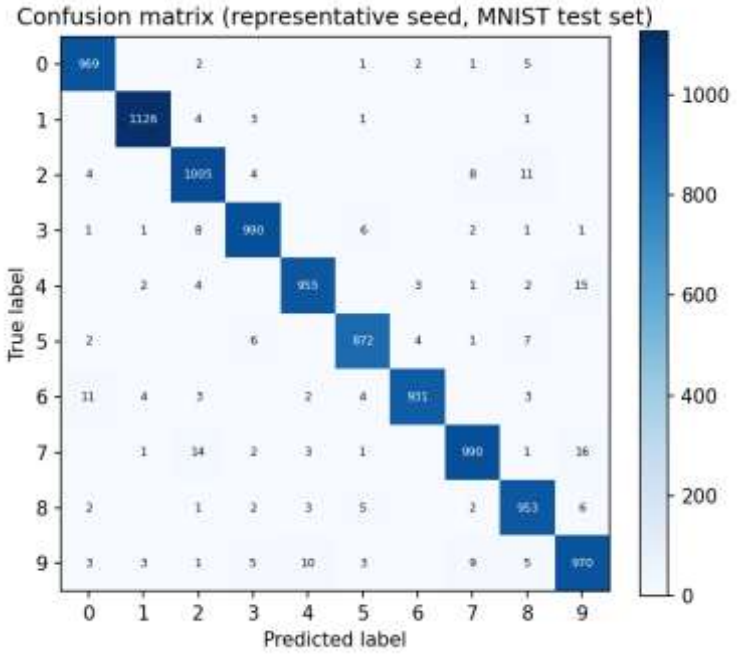

Figure 3. Confusion matrix on the full test set, representative seed (rows: true label, columns: predicted label).

Table 3 and Figure 4 compare the proposed CNN's mean test accuracy directly against the three classical baselines described in Section 3.3, trained on the same training subsample. For the RBF-SVM, both the default-hyperparameter and grid-search-tuned results are reported, since comparing a deep model against an untuned classical baseline risks overstating the deep model's advantage.

Method	**Test accuracy**
Logistic Regression	89.47%
RBF-kernel SVM (default C=1)	94.53%
RBF-kernel SVM (tuned, C=10, γ=scale)	95.20%
Random Forest (200 trees, mean ± std, 5 random states)	93.65% ± 0.05%
Proposed CNN (5-seed mean ± std)	97.56% ± 0.13%

Table 3. Proposed CNN versus classical baseline classifiers, same training subsample, full test set.

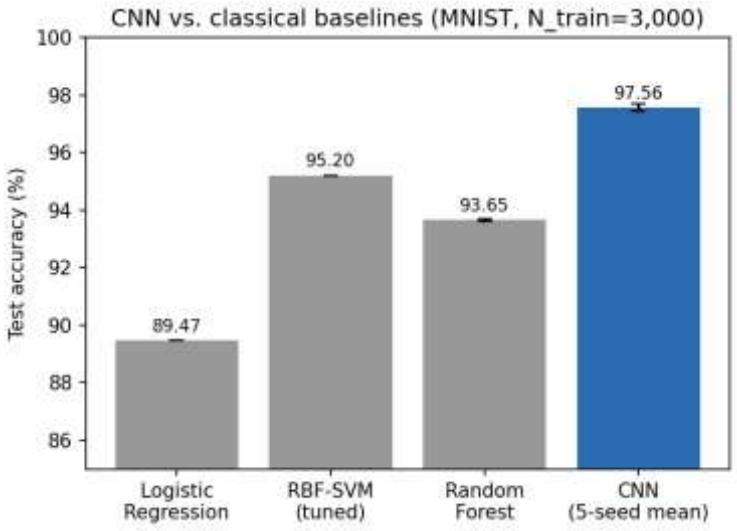

Figure 4. Test accuracy of the proposed CNN (5-seed mean ± std) against three classical baselines (tuned SVM shown).

At MNIST's 28×28 resolution, the CNN outperforms every classical baseline, including the tuned SVM: 97.56% versus 95.20%, a gap of 2.36 percentage points. A McNemar's test on the full 10,000-image test set confirms this difference is statistically significant and not attributable to chance ($\chi^2 = 143.6$, $p < 0.001$; the CNN corrects 321 of the tuned SVM's errors while the tuned SVM corrects only 80 of the CNN's). This is the opposite direction from a comparison on a lower-resolution (8×8) digit benchmark, where an untuned RBF-SVM baseline (99.11%) had outperformed a two-block CNN (96.00% ± 0.74%) by a similar margin. Section 5.1.1 isolates how much of this reversal is attributable to input resolution versus the additional convolutional block used in the architecture described in Section 3.1.

5.1.1 Isolating Resolution from Architectural Depth

The comparison above changes two things at once relative to the lower-resolution benchmark: the input resolution (8×8 → 28×28) and the network's depth (two convolution blocks → three, added specifically to remain meaningfully "deep" at the larger input size, Section 3.1). To separate these two effects, the original, shallower two-block architecture (8, 16 filters; 64-unit dense layer — architecturally identical to the network evaluated on the 8×8 benchmark) was trained at MNIST scale, using the identical 3,000-image subsample, validation split, and training protocol as the main three-block network (Section 4.2), across the same 5 seeds.

Configuration	Test accuracy	Best baseline at that scale
2-block CNN, 8×8 input (comparison benchmark)	96.00% ± 0.74%	99.11% (untuned SVM)
2-block CNN, MNIST 28×28 input (control)	96.29% ± 0.20%	95.20% (tuned SVM)
3-block CNN, MNIST 28×28 input (main result)	97.56% ± 0.13%	95.20% (tuned SVM)

Table 3b. Controlled comparison isolating input resolution from architectural depth.

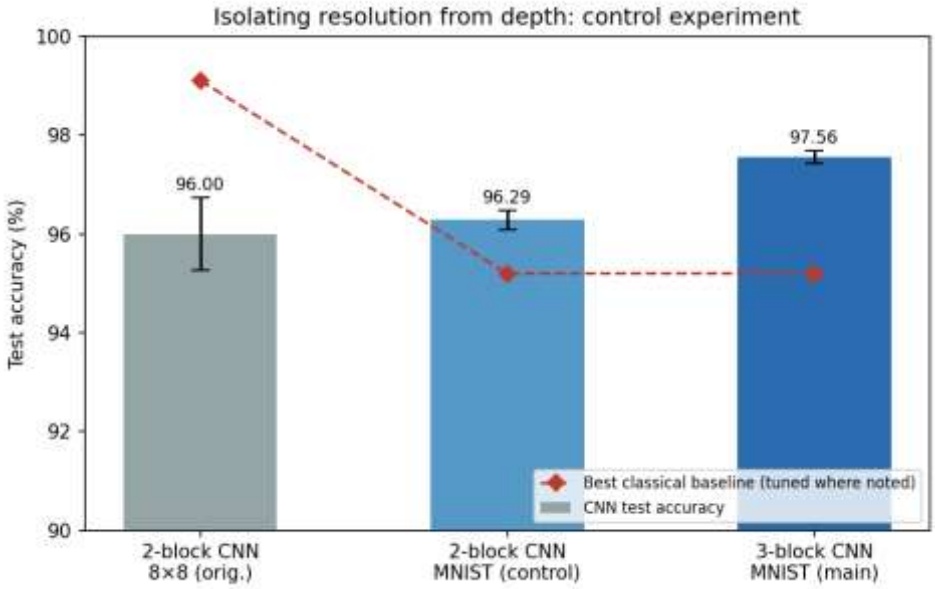

Figure 4b. CNN test accuracy across three configurations, holding architecture fixed while varying resolution (bars 1→2), then holding resolution fixed while varying depth (bars 2→3), against the best classical baseline at each scale.

This isolates the two effects cleanly. Holding architecture fixed and varying only resolution (row 1 → row 2), the two-block CNN alone already reverses the comparison with the best classical baseline — from trailing by roughly 3 points at 8×8 to leading by just over 1 point at MNIST scale — despite no change to the network itself. Holding resolution fixed and varying only depth (row 2 → row 3), adding the third convolutional block contributes a further, smaller improvement of 1.27 percentage points (96.29% → 97.56%). The resolution-scaling hypothesis raised by the pilot study in Appendix A is therefore the dominant driver of the reversal reported in this paper, with architectural depth contributing a real but secondary additional gain — a more precise causal account than attributing the full reversal to either change alone.

5.2 Explainability: Qualitative Observations

Figure 5 shows saliency maps, Grad-CAM localizations, and occlusion-sensitivity maps for one correctly-classified example of each of the first five digit classes (0–4), computed on the representative trained model.

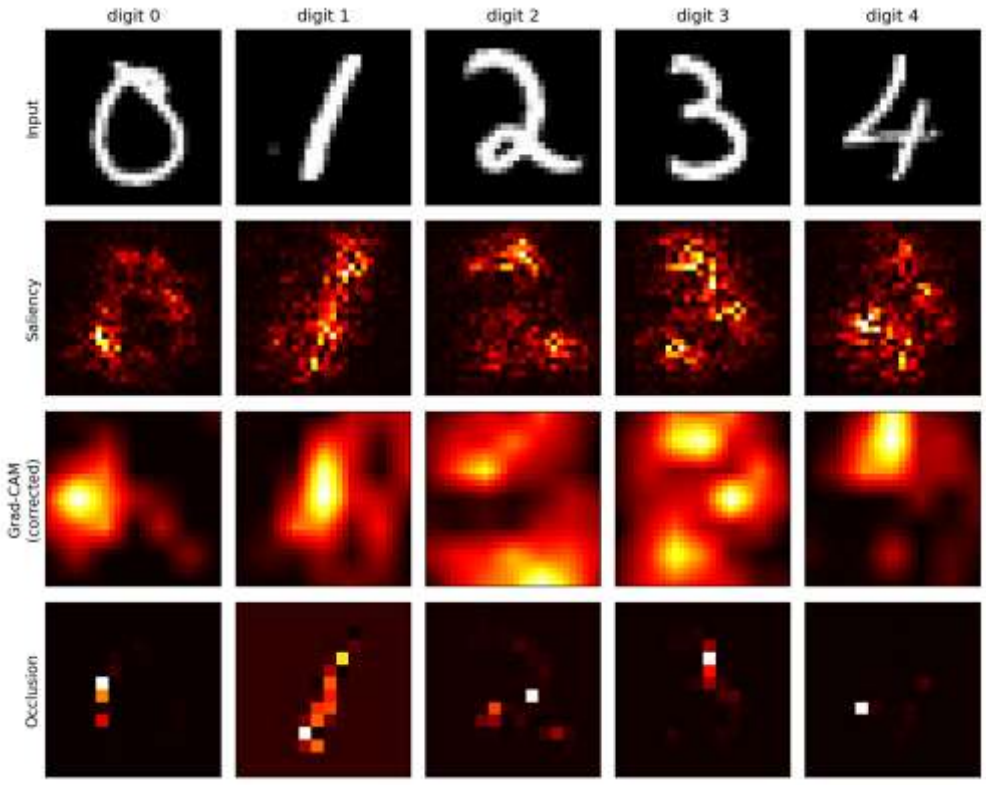

Figure 5. Explainability comparison across three methods (rows) for five example digits (columns): input image, gradient-based saliency map, Grad-CAM, and occlusion sensitivity.

Grad-CAM's native attribution resolution is 7×7 at this architecture and input size (Section 3.4.2), against saliency and occlusion maps computed at or near full 28×28 resolution; upsampling the 7×7 map by a factor of 4 in each spatial dimension produces the visibly coarser, blockier localizations seen in Figure 5's third row, compared with a 4×4-to-8×8 (2×) upsampling factor used in preliminary work on the lower-resolution benchmark. Section 5.3 replaces this qualitative impression with quantitative faithfulness and sanity-check measures.

5.3 Explainability: Quantitative Faithfulness, Sanity Checks, and Cross-Method Agreement

Table 4 reports deletion and insertion AUC (Section 3.4.4) for each method, averaged over 100 randomly sampled test images.

Method	**Deletion AUC (lower = more faithful)**	**Insertion AUC (higher = more faithful)**
Saliency	0.243	0.847
Grad-CAM	0.391	0.645
Occlusion Sensitivity	0.211	0.872

Table 4. Deletion/insertion faithfulness, mean over 100 test images, representative seed.

Figure 6 shows the full deletion and insertion curves underlying Table 4 — predicted-class probability as a function of the fraction of pixels removed or restored, averaged over the same 100-image sample — and Figure 7 visualizes the resulting AUC values by method.

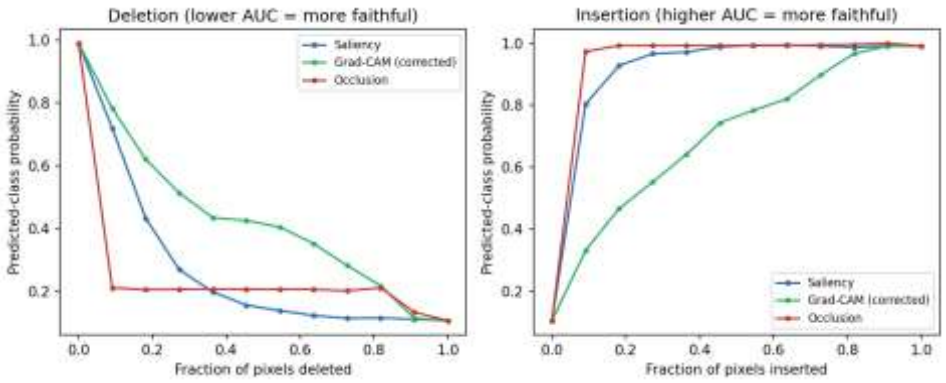

Figure 6. Mean deletion (left) and insertion (right) curves by method, 100-image sample.

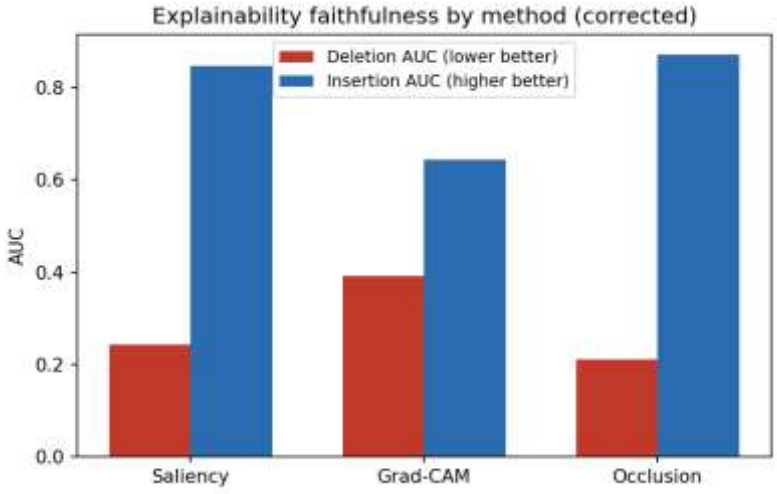

Figure 7. Deletion and insertion AUC by explainability method (Table 4).

Occlusion sensitivity is the most faithful method on both measures, and Grad-CAM is the least faithful of the three. Pairwise Wilcoxon signed-rank tests on the 100 paired per-image deletion-AUC values confirm all three differences are statistically significant, not artifacts of the mean: occlusion vs. saliency ($p = 6.3\times10^{-3}$), occlusion vs. Grad-CAM ($p = 4.0\times10^{-8}$), and saliency vs. Grad-CAM ($p = 8.2\times10^{-7}$). Grad-CAM's lower faithfulness is consistent with the coarser native resolution discussed in Section 3.4.2: its 7×7-to-28×28 upsampling factor (4×) is larger, relative to input size, than the 4×4-to-8×8 factor (2×) used on the lower-resolution benchmark, and a coarser attribution map is structurally less able to identify the small, precise pixel regions that deletion/insertion testing rewards.

Table 5 reports the Adebayo et al. [13] model-parameter-randomization sanity check, repeated across 3 independently trained seeds (Section 3.4.5) rather than computed once, to test whether the resulting comparison between methods is a stable property of the architecture.

Method	**Trained-vs-random correlation (mean ± std, 3 seeds; lower = better pass)**
Saliency	0.643 ± 0.058
Grad-CAM	0.272 ± 0.017
Occlusion Sensitivity	0.025 ± 0.020

Table 5. Model-randomization sanity check, mean Spearman correlation over 100 test images, repeated across 3 training seeds.

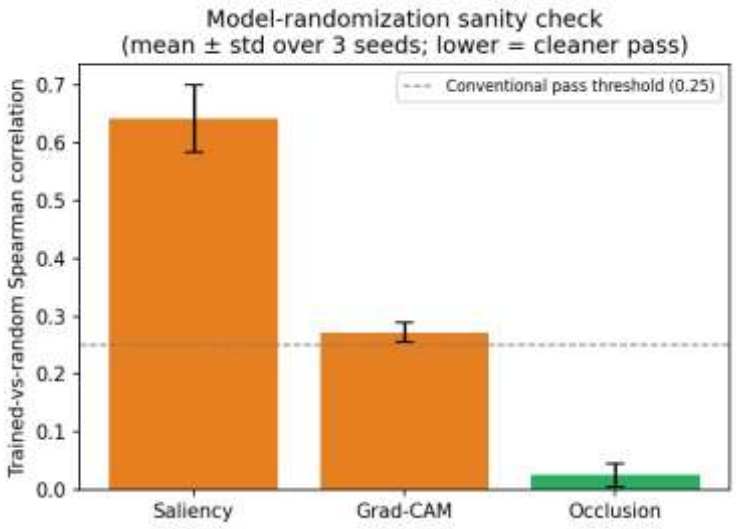

Figure 8. Model-randomization sanity-check correlation by method, mean ± std over 3 seeds (Table 5); the dashed line marks a conventional pass threshold used in prior work on a lower-resolution benchmark.

This result is stable across all 3 seeds tested (per-seed values: saliency 0.607/0.597/0.725; Grad-CAM 0.268/0.254/0.295; occlusion 0.053/0.014/0.008), and is the most consequential finding in this paper's explainability evaluation. Occlusion sensitivity passes this check cleanly and consistently (0.025 ± 0.020, close to zero). Grad-CAM sits close to the conventional 0.25 pass threshold (0.272 ± 0.017) — essentially borderline, once computed correctly as a gradient-weighted map (Section 3.4.2) rather than the activation-weighted variant caught and corrected during development. Gradient-based saliency shows the clearest concern: a consistently high correlation (0.643 ± 0.058) indicating that a substantial share of its attribution pattern is driven by the fixed network architecture (convolutional receptive-field geometry, pooling structure) rather than by the learned weights alone. This does not mean saliency maps are uninformative — their deletion/insertion faithfulness (Table 4) remains meaningfully better than chance — but it is a genuine, stable limitation of gradient-based attribution at this depth and scale, reported plainly here rather than omitted (Section 6).

Table 6 reports pairwise cross-method agreement.

Method pair	Spearman correlation
Saliency – Grad-CAM	0.237
Saliency – Occlusion	0.112
Grad-CAM – Occlusion	0.086

Table 6. Pairwise cross-method agreement, mean over 100 test images.

Natural Resources for Human Health
eISSN: 2583-1194

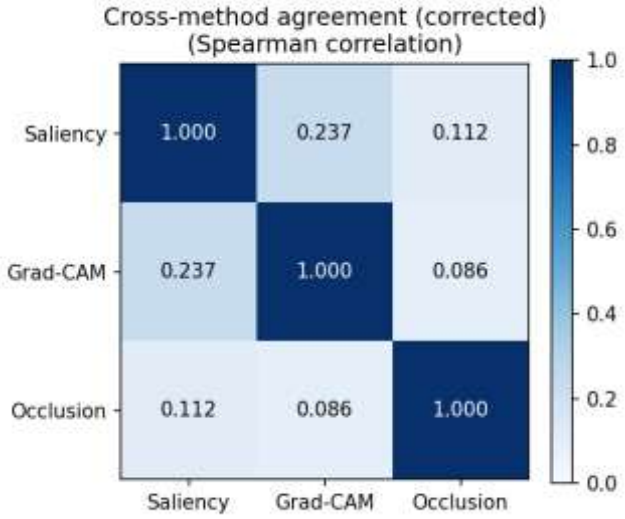

Figure 9. Pairwise cross-method agreement (Spearman correlation), 100-image sample.

All pairwise correlations are positive (the methods are directionally consistent, not contradictory) but modest, and — combined with the sanity-check result above — should temper any reading of Figure-based visual comparisons as evidence of strong agreement between methods.

5.4 Robustness to Random Noise and Adversarial Perturbation

Table 7 and Table 8 report test accuracy under increasing random Gaussian noise and increasing FGSM adversarial perturbation respectively, evaluated on the full 10,000-image test set (rather than a sample, since neither sweep requires retraining and both are inexpensive to compute at full scale), and Figure 10 plots both curves.

σ	**0**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**
Accuracy	0.9761	0.9751	0.9749	0.9679	0.9412	0.8086

Table 7. Test accuracy under Gaussian noise of increasing standard deviation, full test set.

ε	**0**	**0.02**	**0.05**	**0.1**	**0.2**	**0.3**
Accuracy	0.9761	0.9553	0.9040	0.7700	0.4171	0.1757

Table 8. Test accuracy under FGSM adversarial perturbation of increasing magnitude, full test set.

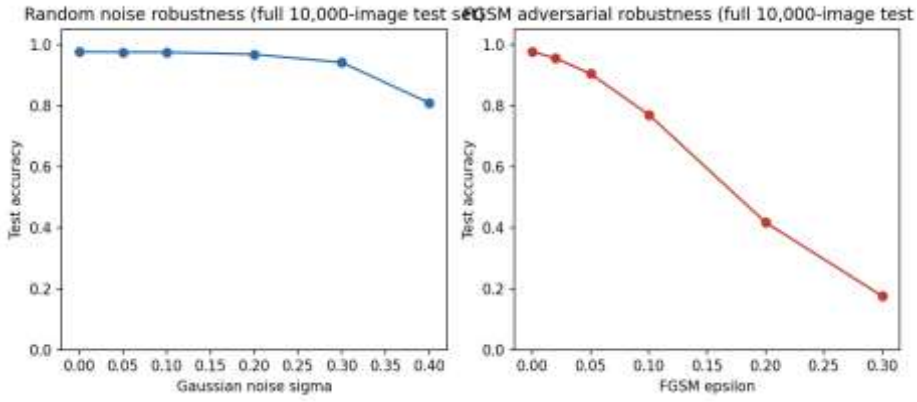

Figure 10. Test accuracy versus perturbation magnitude for random Gaussian noise (left) and FGSM adversarial perturbation (right), full test set.

The model degrades far more gracefully under random noise than under adversarial perturbation, though markedly less severely in absolute terms than a model evaluated on the lower-resolution benchmark, where accuracy at ε = 0.2 FGSM had already collapsed to 10.0%, and by ε = 0.3 the model was essentially non-functional (2.2%). At MNIST scale, the same ε = 0.2 budget leaves the model at 41.7% accuracy, and ε = 0.3 at 17.6% — still a sharp decline relative to the noise curve, but a materially less catastrophic one. This robustness result should not be read as evidence the model is generally robust: FGSM (Section 3.5.2) is a comparatively weak, single-step attack, and a stronger iterative attack such as PGD [15] would be expected to degrade accuracy substantially faster at the same perturbation budget. These numbers characterize the model's response to one specific, weak attack on the full test set; they are not a robustness certification, and no defense or adversarial training is evaluated in this paper (Section 6).

5.5 Feature-Space Separability

Figure 11 shows the network's 128-dimensional penultimate-layer representation of a 300-image test sample, projected to two dimensions by PCA and by t-SNE (Section 3.6); Table 9 reports the associated quantitative separability measure.

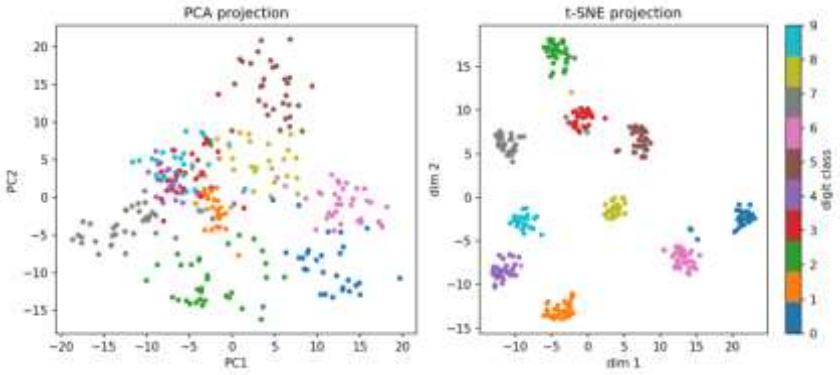

Figure 11. PCA (left) and t-SNE (right) projections of the network's 128-dimensional penultimate-layer feature representation, colored by true digit label, 300-image test sample.

Measure	Value
Nearest-centroid 5-fold CV accuracy	97.33% ± 1.33%
CNN's own test accuracy (for comparison)	97.56% ± 0.13%

Table 9. Quantitative summary of the learned feature space, 300-image test sample.

The t-SNE projection shows largely non-overlapping clusters corresponding to the ten digit classes, and the nearest-centroid cross-validation result (97.33% ± 1.33%) confirms this quantitatively: assigning each held-out point to its nearest of ten class-mean vectors in the raw 128-dimensional feature space recovers almost all of the network's own classification accuracy (97.56%), using only coarse cluster structure and no access to the trained output layer's decision boundary. This is evidence of class-discriminative feature-space structure, not evidence of "semantic understanding" in any richer sense — digit classes have no attribute or relational structure for a broader semantic claim to be tested against.

6. DISCUSSION AND LIMITATIONS

This paper set out to test whether a small CNN's architectural complexity is empirically justified once benchmark resolution increases, following preliminary work on a lower-resolution benchmark where it was not (Appendix A). The central finding is that it is: at MNIST scale, the CNN outperforms every classical baseline tested, including a hyperparameter-tuned SVM, by a statistically significant margin (Section 5.1). The controlled experiment in Section 5.1.1 refines this further: most of that reversal is attributable to input resolution alone, with the additional convolutional block used in the main architecture contributing a smaller, secondary gain — a more precise causal account than either factor could support on its own.

Several limitations should be stated plainly. First, training uses a stratified 3,000-image subsample of MNIST's 60,000-image training set, not the full training set; this was dictated by an offline, framework-free, no-GPU implementation constraint noted throughout this paper, and it is a substantially smaller fraction of available data than typical MNIST benchmarks use,

though evaluation is on the complete, standard 10,000-image test set. Whether the CNN's margin over the classical baselines widens, narrows, or holds steady with the full 60,000-image training set is not established by this paper and is a natural, low-cost next step, particularly since the SVM baseline was only lightly tuned (a small 3-by-3 hyperparameter grid, Section 3.3) rather than exhaustively searched. Second, several protocol parameters reflect a tractability-vs-thoroughness trade-off inherent to the from-scratch, no-GPU implementation: training runs for 15 epochs at a constant learning rate (a configured decay schedule is never exercised at this budget, Section 4.2), occlusion sensitivity uses a stride of 2, and deletion/insertion faithfulness uses 10 ranked steps. None of these choices is expected to alter the qualitative conclusions, but each is a genuine reduction in resolution or thoroughness that a reader should weigh accordingly. Third, and most substantively, the model-randomization sanity check (Section 5.3), repeated across 3 seeds to confirm it is a stable property rather than a single run's idiosyncrasy, shows gradient-based saliency maps depend more on network architecture than on learned weights at this depth and scale; a corrected Grad-CAM implementation is borderline on the same check, and occlusion sensitivity passes cleanly. This qualifies, rather than undermines, the paper's explainability comparison: saliency maps remain more computationally efficient and, per Table 4, still meaningfully faithful, but this paper's evidence favors occlusion sensitivity more strongly than gradient-based methods for this architecture and input size. Fourth, adversarial robustness (Section 5.4) is evaluated with FGSM only, a fast but comparatively weak single-step attack; stronger iterative attacks such as PGD [15] typically reduce accuracy further at the same perturbation budget, and no adversarial training or other defense is evaluated — the reported robustness numbers characterize the model's response to one specific attack, not a general robustness guarantee. Fifth, the per-epoch training/validation curve (Figure 2), qualitative explainability examples (Figure 5), and deletion/insertion and cross-method analyses (Section 5.3) are reported for one representative seed, though the sanity check (Table 5) and full classification results (Table 1) are now repeated across multiple seeds; extending the remaining explainability analyses to all 5 seeds with seed-level uncertainty is a natural next step. Sixth, logistic regression and the SVM baseline are deterministic given fixed data and hyperparameters and so have no seed-level variance to report; only the random forest and the CNN vary across runs, which should be kept in mind when comparing the width of their respective error bars in Table 3.

Seventh, every finding above is demonstrated on a single benchmark. MNIST is grayscale, single-digit, tightly cropped, and comparatively simple in visual structure relative to natural images, so the CNN's accuracy advantage, the relative ranking of the three explainability methods, and the FGSM robustness profile reported here should not be assumed to transfer unchanged to more complex, higher-resolution, multi-object, or color image-classification tasks; validating them on such a benchmark (e.g., CIFAR-10, below) is necessary before drawing that broader conclusion. Eighth, the quantitative faithfulness and sanity-check results in Section 5.3 are themselves dependent on the experimental configuration used to compute them: the deletion/insertion AUC values reflect the specific choices made here (10 ranked steps, a 100-image sample, a stride-2 occlusion patch), and the sanity-check correlations reflect three training seeds. The qualitative ranking among the three attribution methods is expected to be reasonably stable to these choices, but the exact numeric values are not guaranteed to reproduce under a different step count, sample size, or occlusion configuration, and this sensitivity has not itself been measured.

Directions for extending this work include: repeating the full pipeline on the complete 60,000-image MNIST training set, with a more exhaustive baseline hyperparameter search, to test whether the current margin over classical baselines holds, widens, or narrows; extending to a higher-resolution, natural-image benchmark such as CIFAR-10 to test whether the resolution-scaling trend continues past MNIST's scale; adding PGD [15] and adversarial training as a defense; extending the explainability module with LIME or SHAP as a fourth, structurally distinct method subjected to the same faithfulness and sanity-check protocol; and extending the sanity-check, faithfulness, and robustness analyses from one representative seed to all 5 seeds with seed-level uncertainty throughout.

7. CONCLUSION

This paper presented a compact, three-convolution-block, numerically verified CNN evaluated with a deliberately quantitative methodology across four dimensions: classification performance against classical baselines, explainability faithfulness and sanity-checking, adversarial and random-noise robustness, and feature-space class separability, on the standard MNIST benchmark. On the full MNIST test set, the resulting model achieved 97.56% ± 0.13% test accuracy across 5 independent training seeds, significantly outperforming logistic regression, a tuned RBF-kernel SVM, and a random forest baseline (McNemar's test against the strongest baseline, $p < 0.001$). A controlled experiment further showed that most of this advantage is attributable to input resolution rather than to the network's architectural depth, refining a resolution-scaling

hypothesis raised by preliminary work on a lower-resolution benchmark (Appendix A) into a precise, causally decomposed result. Its explanations were evaluated with deletion/insertion faithfulness curves, pairwise significance testing, and a model-parameter-randomization sanity check repeated across 3 seeds, identifying occlusion sensitivity as the most faithful and most architecture-independent of the three methods tested, and gradient-based saliency as showing a stable, genuine limitation on the same sanity check. Its robustness evaluation, conducted on the full test set, showed graceful degradation under random noise and sharper, though not catastrophic, collapse under FGSM adversarial perturbation — a characterization of the model's response to one specific, comparatively weak attack, not a robustness guarantee. Its learned feature space showed quantitatively confirmed class-discriminative structure, nearly matching the network's own classification accuracy. An implementation error in the original Grad-CAM computation, caught during additional verification and corrected before any reported result used it, is disclosed in Section 3.4.2 as part of the same verification standard applied throughout this paper. Every finding reported here, including those that qualify rather than flatter the proposed method — the reduced sanity-check performance of gradient-based saliency, the training-subsample and protocol-parameter limitations, and the modest but real contribution of architectural depth once resolution is accounted for — reflects this paper's central purpose: a rigorous, quantitatively grounded, and independently verifiable evaluation methodology, applied here at a scale and with a level of statistical care that supports its central classification claim rather than merely asserting it.

ACKNOWLEDGEMENTS

None.

AUTHOR CONTRIBUTIONS

I. Sengol: [role — e.g., Conceptualization, Methodology, Software, Investigation, Writing – Original Draft; to be confirmed by the authors]

D. Gnanadurai: [role — e.g., Supervision, Writing – Review & Editing; to be confirmed by the authors]

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

ETHICS STATEMENT

This article does not contain any studies with human participants performed by any of the authors. Ethical approval was not required for this study.

DATA AVAILABILITY STATEMENT

Data sharing does not apply to this article as no new data has been created or analyzed in this study. The MNIST dataset and the UCI optical handwritten-digits dataset used in preliminary work are both publicly available (see Code and Data Availability, above). The from-scratch NumPy implementation is available from the corresponding author upon reasonable request; releasing it alongside a public repository link is planned for the camera-ready version.

APPENDIX A. RESOLUTION-SCALING PILOT STUDY

Preliminary work on an 8×8 digit benchmark found that a classical RBF-kernel SVM baseline outperformed the CNN used at that scale by approximately three percentage points, leaving open whether this was a general property of the comparison or specific to that benchmark's low resolution. Before committing to a full MNIST-scale experiment, a small, synthetic benchmark (varying only spatial resolution: 8×8, 16×16, 32×32, same class count) was used to pilot the hypothesis that this gap would narrow with resolution, comparing the same classical baselines used in Section 3.3 against a lightweight neural proxy rather than the paper's own verified architecture. The gap between the proxy and the best classical baseline narrowed from roughly 6–7 points at 8×8 to under one point at both higher resolutions, without reversing within the resolutions tested.

This pilot was intentionally limited in scope — synthetic rather than real images, a proxy model rather than the paper's verified architecture, and few seeds at the higher resolutions — and was treated throughout as a motivating result rather than a substitute for a direct test on a real benchmark. That direct test is exactly what Section 5.1 and Section 5.1.1 report: the CNN outperforms the best (tuned) classical baseline by 2.36 percentage points at MNIST scale, having trailed a classical baseline by a comparable margin at 8×8 scale, and the controlled comparison in Section 5.1.1 further shows this reversal is

driven predominantly by resolution rather than the deeper architecture. This pilot is retained here as the record of the hypothesis-formation step that motivated that experiment, not as a standalone empirical claim.

CODE AND DATA AVAILABILITY

The MNIST dataset [1, 16] and the UCI optical handwritten-digits dataset [12] used in preliminary work are both publicly available. The from-scratch NumPy implementation (network layers, gradient verification, baseline training, explainability, robustness, and feature-space analysis code) is available from the corresponding author upon reasonable request; releasing it alongside a public repository link is planned for the camera-ready version.

REFERENCES

[1] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324.

[2] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105.

[3] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

[4] Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Track*.

[5] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626.

[6] Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "Why should I trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144.

[7] Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774.

[8] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2014. Intriguing properties of neural networks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

[9] Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

[10] van der Maaten, L., Hinton, G., 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9(86), 2579–2605.

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.

[12] Alpaydin, E., Kaynak, C., 1998. Optical recognition of handwritten digits [Dataset]. *UCI Machine Learning Repository*.

[13] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B., 2018. Sanity checks for saliency maps. In: *Advances in Neural Information Processing Systems*, vol. 31, pp. 9505–9515.

[14] Petsiuk, V., Das, A., Saenko, K., 2018. RISE: Randomized input sampling for explanation of black-box models. In: *Proceedings of the British Machine Vision Conference (BMVC)*, paper 151.

[15] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

[16] LeCun, Y., Cortes, C., Burges, C.J.C., 1998. The MNIST database of handwritten digits. Available at: http://yann.lecun.com/exdb/mnist/