International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(14s) 2026 pp. 76-88
https://doi.org/10.70917/ijcisim-2026-4197
Article

Feature-Decoupled CNN-Boosting Ensemble for Robust Handwritten Digit Recognition: Accuracy, Generalization and Deployment Trade-offs

Prashant Kumar[1], Ragini Shukla[2]

[1,2]Department of IT & CS, Dr. C. V. Raman University, Bilaspur, Chhattisgarh, India
Corresponding author: Prashant Kumar(prashantkulmitraiant@gmail.com)

Abstract: Handwritten digit recognition is also relevant to the digitization of documents, data capture in education, postal automation and financial record processing, but models that excel on normalized benchmarks might not be resilient to domain shift and computational constraints. The present study suggests a hybrid approach that combines a compact three-block CNN with exports of a 256-dimensional feature vector Dense1 to classifiers (AdaBoost, XGBoost and LightGBM) to obtain spatial representations from handwritten digit images. They make predictions using hard majority voting guided by the validation. The model was trained on the EMNIST Digits split which has 240,000 training images and 40,000 test images, using controlled augmentation with stratified validation and leakage-prevention rules. Evaluation was performed across classes, across the single classifiers, with component ablations, 10-fold cross validation, addition of synthetic noise, external-domain test sets (USPS and CEDAR), a perturbation test with the FGSM and computational profiling. The overall performance with the accuracy of 98.45% and macro precision, recall and F1 score of 0.984 is 1.60% better than the standalone CNN. The accuracy drops to 98.20%, 97.95% and 97.80% when removing AdaBoost, XGBoost or LightGBM, respectively. The accuracy was kept at 97.82% when noise was Gaussian and 97.53% when noise was salt-and-pepper, but dropped to 94.80% on USPS and 90.20% on CEDAR. The ensemble achieved 85.30% accuracy while the CNN achieved 82.10% accuracy for the FGSM with $\varepsilon = 0.1$. The gains were obtained with 45 minutes of training, 150 ms/image CPU inference and 3.5 GB of memory. The contribution is then a characterization, guided by evidence, of when the CNN-to-boosting decoupling gives an edge to recognition and when it is hindered by costs of generalization and deployment.

Keywords: Handwritten Digit Recognition; EMNIST; convolutional neural network; deep feature extraction; AdaBoost; XGBoost; LightGBM; ensemble learning; domain generalization

1. Introduction

Numerical handwritten data is still found in examination records, postal codes, financial records, administrative forms, registers of archives and other paper-based processes. The long history of this problem is handwriting varies in scale, slant, curvature, stroke order, thickness, closure and acquisition quality and is a pattern-recognition problem. The variations are particularly significant for classes 3 and 8, 4 and 9 and 6 and 8. A useful recognition system should thus be able to learn stable class evidence without being too much influenced by the appearance of the benchmarks.

Handwritten Digit Recognition was revolutionized by convolutional neural networks (CNNs), which used hierarchical feature learning instead of handcrafted descriptors. LeNet family showed that document recognition can be achieved with the effectiveness of convolution, subsampling and gradient based training (LeCun et al., 1998). Subsequent research extended optimizations, regularizations and depths to rectified activations, batch normalization, dropout and residual learning (He et al., 2016; Ioffe & Szegedy, 2015; Srivastava et al., 2014). However, an end-to-end CNN still features representation learning together with a neural classification head.

Copyright: © 2026 by the authors

The final SoftMax layer doesn't need to take full advantage of the learned feature space as the margin-based or tree-based classifiers do. A solution to this is hybrid recognition which employs a neural network as an automatic feature extractor and passes the learned representation to the second classifier. CNN-SVM studies proved that separation of feature learning and decision formation can be beneficial for handwritten digits (Ahlawat & Choudhary, 2020; Niu & Suen, 2012). More recent studies have been based on Deep Networks, Transfer Learning, Feature Fusion, Attention and Ensembles, which have led to better class-wise performance and coverage in multilingual applications (Agrawal et al., 2024; Azawi, 2023; Fateh et al., 2024; Rajpal & Garg, 2023). It is suggested that these studies provide encouraging evidence for complementary learners but numerical comparisons of these datasets can be challenging due to various datasets, partitions, preprocessing, augmentation, parameter budgets and hardware conditions.

In the present study, we explore a different approach to hybridization, where a compact CNN generates a fixed representation of 256 dimensions and it is provided to AdaBoost, XGBoost and LightGBM. The algorithms are based on various error emphasis mechanisms, regularization of additive trees and partitioning of high-dimensional feature space (Chen & Guestrin, 2016; Freund & Schapire, 1997; Ke et al., 2017). Then they predict labels and they combine their predicted labels using hard majority voting. The study not only assesses the accuracy of clean tests, but also synthetic corruption, adversarial perturbation, memory and training cost, component contribution and class-wise consistency. The main point is that a recognition gain must be understood within the context of the resources and constraints needed to gain it. In this context, the paper queries the improvement of the implemented CNN baseline on a common protocol and the classifiers that are the most significant for the final result, the performance of the system outside of an in-domain test set, as well as whether the extra computational burden is acceptable in feasible document-processing scenarios.

2. Related Work

2.1 CNN-based handwritten digit recognition

CNN-based digit recognition is based on local receptive fields and shared weights that enforce the spatial structure and regulate the growth of parameters. The classic gradient-based document-recognition architecture demonstrated that the use of learned filters for document recognition could be used in place of explicit stroke rules (LeCun et al., 1998). Multi-column and multi-committee approaches were then shown to be effective for decreasing the recognition error (Cireşan et al., 2012), which is further decreased when using models that are trained independently. The key is that regularization and augmentation are at the heart of handwriting datasets as they allow for many plausible geometric variations. Co-adaptation and optimization stability was the influence which is addressed by dropout and batch normalization (Ioffe & Szegedy, 2015; Simard et al., 2003; Srivastava et al., 2014) and elastic deformation has been particularly influential in digit recognition.

There have been several directions in which the baseline CNN has been extended recently. For MNIST, Azawi (2023) exploited transfer learning combined with deep models selected using statistics to enhance the recognition performance of the target classes. To investigate the hybrid convolutional vision-transformer design, Agrawal et al. (2024) tested it on clean and noisy digit images, where global attention is increasingly used, as well as local convolutional features. These models can be very sensitive, but their deeper or multi-model versions can be more complex and can be more challenging to attribute the gain.

2.2 Feature-decoupled and hybrid classifiers

Feature-decoupled systems keep the power of representation learning of CNNs but instead of the native neural decision layer. Niu and Suen (2012) applied a CNN as a trainable feature extractor and an SVM as recognizer and Ahlawat and Choudhary (2020) reported one such CNN-SVM design for MNIST. The same observation was made by Bhatti et al. (2023) where CNN features extracted with SVM were found to be better for Urdu numeral recognition than those features extracted with SoftMax. These results corroborate the overall conclusion that the optimal feature extractor and the optimal decision function do not have to be of the same type.

Combinations have also been looked at outside of the Latin digits. Rajpal and Garg (2023) combined the characteristics of the pretrained CNNs and classified the optimized representation using SVM for handwritten Hindi numerals. Fateh et al. (2024) employed attention-driven transfer learning with 12 numeral systems. These studies also stress that some evaluation is script-specific: what works well in one script doesn't necessarily work well in another script and what works well in one benchmark doesn't necessarily work well in another.

2.3 Boosting and ensemble decision formation

AdaBoost builds a series of weak learners and boosts the weight of the misclassified samples (Freund & Schapire, 1997). XGBoost introduces new features such as regularized gradient boosting, sparse-aware processing and scalable tree construction (Chen & Guestrin, 2016). LightGBM implements a discretization method based on histogram and growth method of leaves for enhancing training efficiency of structured feature matrices (Ke et al., 2017). All three are boosting, but have slightly different training dynamics and decision partitions that create helpful error diversity.

The advantage of ensemble learning is that if the individual classifiers are competent in solving the problem, but they make imperfectly correlated errors (Dietterich, 2000), then the ensemble will work best. In hard voting, the voting is transparent and the reliability of each classifier is assumed to be the same. The present work applies hard voting to ensure interpretability of the combination rule and sets the priority of the tie-breaker on the validation data before the test evaluation. This design allows a direct examination of the value added by each of the classifiers.

2.4 Research gap and contributions

The literature establishes that CNNs learn effective image representations, that non-neural classifiers can exploit deep features and that ensembles may improve recognition. However, four gaps remain important. First, many studies report only the strongest accuracy without quantifying the marginal contribution of each ensemble member. Second, robustness is often inferred from augmentation rather than tested under synthetic corruption, external datasets and adversarial perturbation. Third, comparisons rarely integrate predictive performance with CPU latency and memory use. Fourth, the distinction between strong in-domain performance and trustworthy deployment is not always explicit.

This study addresses these gaps through four contributions: (i) a compact feature-decoupled CNN-boosting architecture that exposes a 256-dimensional embedding to three heterogeneous boosting classifiers; (ii) controlled comparisons with the standalone CNN and each individual CNN-boosting model; (iii) ablation, cross-validation, synthetic-noise, external-domain and FGSM evaluations; and (iv) transparent reporting of the accuracy-latency-memory trade-off. The work does not claim universal superiority; its contribution is the integrated evidence describing when the proposed design is beneficial and where its limitations emerge.

3. Proposed Method

3.1 Data and partitioning

The research gap and contributions will be discussed next. The literature validates that CNNs extract efficient image representations, non-neural classifiers can take advantage of deep features and ensembles can enhance recognition. There are four significant gaps, however. First, many studies do not report on the accuracy of the ensemble and only accuracy of the best individual member. Second, robustness is typically assumed by adding more data and augmenting existing data, not tested with synthetic perturbations, external datasets or adversarial perturbation. Thirdly, comparisons lack a combination of predictive performance and CPU latency and memory consumption. The fourth is that the separation of good in-domain performance and reliable deployment is not clear.

In this work, these issues are tackled by (i) a compact feature-decoupled CNN-boosting architecture that feeds a 256-dimensional embedding into 3 different and heterogeneous boosting classifiers; (ii) controlled testing of the standalone CNN and each individual CNN-boosting model; (iii) ablation, cross-validation, synthetic-noise, external-domain and FGSM testing; and (iv) transparent reporting of the accuracy-latency-memory trade-off. The work is not meant to be superior in all aspects; it is intended to offer integrated evidence of the benefits of the proposed design and the points at which there are limitations.

Dataset / condition	Role	Images / format	Purpose
EMNIST Digits	Primary development and test benchmark	240,000 train; 40,000 test; 28 × 28 grayscale	CNN training, validation, in-domain comparison
USPS	External test	9,298 low-resolution postal digits; resized to 28 × 28	Moderate cross-domain transfer

CEDAR	External test	Bank-check / document digit images; standardized to 28 × 28	Stronger document-domain shift
Synthetic corruptions	Stress tests on EMNIST test images	Gaussian noise σ = 0.1; salt-and-pepper p = 0.05	Controlled robustness assessment
FGSM	Adversarial stress test	ε = 0.1	Relative vulnerability of CNN and ensemble

Table 1. Datasets and evaluation conditions.

3.2 Preprocessing and augmentation

Where necessary, all inputs were converted to grayscale, resized to 28x28 and normalized to [0, 1]. The augmentation during training time consisted of rotation, scaling, translation, shear, controlled noise and elastic deformation up to ±10°. The augmentation was done stochastically on the training samples and a deterministic process on validation and test samples. This rule ensures that transformed copies of evaluation images do not get into the model fitting. The augmentation policy was tested incrementally to see its impact on accuracy and the train-validation gap (Shorten & Khoshgoftaar, 2019; Simard et al., 2003).

3.3 CNN feature extractor

CNN has 3 convolutional layers having 32, 64 and 128 3 × 3 filters. The ReLU and batch normalization modules are used at each stage, with 2 x 2 max pooling after the first and second stages. The last 7x7x128 tensor is flattened to 6,272 values and compressed with a Dense1 layer with ReLU activation, a layer containing 256 neurons. A dropout rate of 0.25 is used while training CNN. A temporary ten-neuron SoftMax layer enables categorical cross-entropy optimization, but the hybrid path gets rid of this decision head after training and gives Dense1 activations as output.

$$z_i = f_{CNN}(x_i; \theta) \in \mathbb{R}^{256} \quad (1)$$

In this case, refers to the digit image that is fed to the CNN, θ are the trained parameters of the CNN and is a 256-dimensional fixed representation. Adam's optimized CNN learning rate and batch size were 0.001 and 128 respectively. Restored best checkpoint and monitored validation loss with early stopping. The feature extractor was trained with the training behaviour disabled, meaning dropout behaviour was not active and batch-normalization moving statistics was not active. xizi

3.4 Boosting classifiers and decision fusion

All boosting classifiers received the same training, validation and test feature matrices. 50 estimators, learning rate 1.0, were used for AdaBoost. The XGBoost model was fitted with 100 estimators, maximum depth 3 and learning rate 0.1. LightGBM was trained with 100 estimators, 31 leaves and learning rate 0.05. Constrained validation searches were used to select parameters to minimize unnecessary search cost.

The three classifiers get labels $h_A(z_i), h_X(z_i)$ *and* $h_L(z_i)$. for an unseen feature vector z_i. When there is at least a pair of classifiers that agree, then the modal class is the final. If they do not match, then the prediction of the highest-ranking validator macro-F1 of the classifiers is returned, with the ranking being frozen before final testing.

$$\hat{y}_i = mode\{h_A(z_i), h_X(z_i), h_L(z_i)\} \quad (2)$$

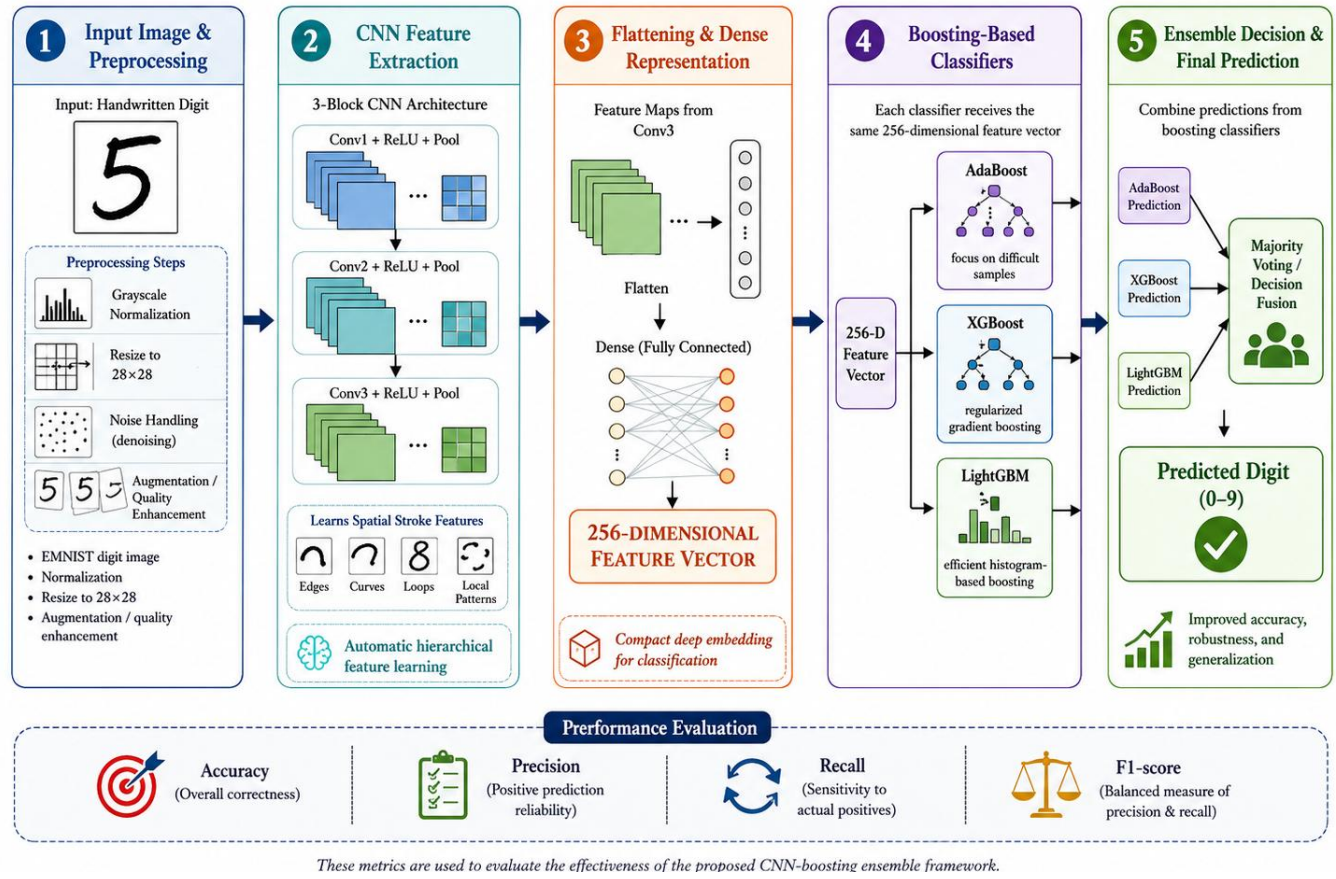

Figure 1. Feature-decoupled CNN-boosting ensemble architecture.

3.5 Evaluation protocol

The accuracy and the mean of precision, recall and F1-score were used as measures of performance. Macro averaging assigns equal weight to each class and thus helps identify if a high overall score hides bad classes (Powers, 2011; Sokolova & Lapalme, 2009). The results and confusion pattern of the classes were also checked.

The variability was estimated using a stratified 10-fold cross-validation methodology and the learning of the representation and the training of the classifiers were repeated within each fold to prevent leakage (Kohavi, 1995). Strict independence was not assumed to be strong by performing pair-wise fold-level significance tests, though such significance tests are reported. Robustness assessment involved Gaussian noise, salt-and-pepper noise and an FGSM attack with $\varepsilon = 0.1$ (Goodfellow et al., 2015). The computational profiling results were obtained by measuring training time on an NVIDIA RTX 3060, inference latency and memory consumption on CPUs.

4. Experimental Environment

The implementation included TensorFlow 2.12 to build the CNN, Scikit-learn 1.2 to train the AdaBoost and to evaluate the results, XGBoost 1.7 and LightGBM 3.3. Training was carried out on an Intel Core i7-11700 processor workstation equipped with an NVIDIA RTX 3060 with 12 GB of VRAM. For each run a random seed, index of parameter values, parameters, checkpoints and software versions were saved. MATLAB R2023b was only used to visualize auxiliary features and to verify baseline and was not part of the operational prediction path.

Component	Configuration
CNN input	28 × 28 × 1
Convolutional blocks	32, 64, 128 filters; 3 × 3; ReLU; batch normalization

Pooling	2 × 2 after blocks 1 and 2
Embedding	Dense1, 256 neurons, ReLU, dropout 0.25
CNN optimization	Adam, learning rate 0.001, batch size 128, early stopping
AdaBoost	50 estimators; learning rate 1.0
XGBoost	100 estimators; max depth 3; learning rate 0.1
LightGBM	100 estimators; 31 leaves; learning rate 0.05
Decision fusion	Hard voting; validation-ranked tie rule

Table 2. Final model configuration.

5. Results

5.1 Overall and class-wise performance

The proposed ensemble was evaluated on the EMNIST Digits test set and the accuracy was found to be 98.45%. The overall measure was macro precision, recall and F1-score, all of which were 0.984, which means that none of the minority classes were sacrificed in the overall result. The F1-scores were in the range of 0.980 (for digit 8) to 0.990 (for digit 1), respectively. The smaller performance gap is consistent with the sentiment that the ensemble was more consistent with the balanced dataset, which indicates that the ensemble made the decisions between the ten classes more consistent, though loop-rich and structurally ambiguous digits were harder.

Digit	Precision	Recall	F1-score
0	0.987	0.987	0.987
1	0.990	0.990	0.990
2	0.982	0.982	0.982
3	0.982	0.982	0.982
4	0.983	0.983	0.983
5	0.981	0.981	0.981
6	0.985	0.985	0.985
7	0.986	0.986	0.986
8	0.980	0.980	0.980
9	0.981	0.981	0.981

Table 3. Class-wise performance of the proposed ensemble on EMNIST Digits.

5.2 Baseline and single-classifier comparison

The standalone CNN achieved an accuracy of 96.85% and a macro F1 of 0.968. The accuracy was further boosted in all cases when the SoftMax decision was replaced with a single boosting classifier: CNN + AdaBoost achieved 97.43%, CNN + XGBoost 98.02% and CNN + LightGBM 98.12%. The full ensemble achieved the best result at 98.45% which is 1.60% higher than the CNN that was implemented. This difference was found to be significant at the $p = 0.002$ level, when using a paired comparison method, at the fold level. The evidence suggests that the 256-dimensional representation has at least some decision structure that the boosting models have exploited more effectively than the selected neural head to conclude a study-specific conclusion.

Model	Accuracy (%)	Macro precision	Macro recall	Macro F1
Standalone CNN	96.85	0.968	0.969	0.968
CNN + AdaBoost	97.43	0.974	0.974	0.974
CNN + XGBoost	98.02	0.980	0.980	0.980
CNN + LightGBM	98.12	0.981	0.981	0.981
Proposed ensemble	98.45	0.984	0.984	0.984

Table 4. Comparison with the implemented baselines.

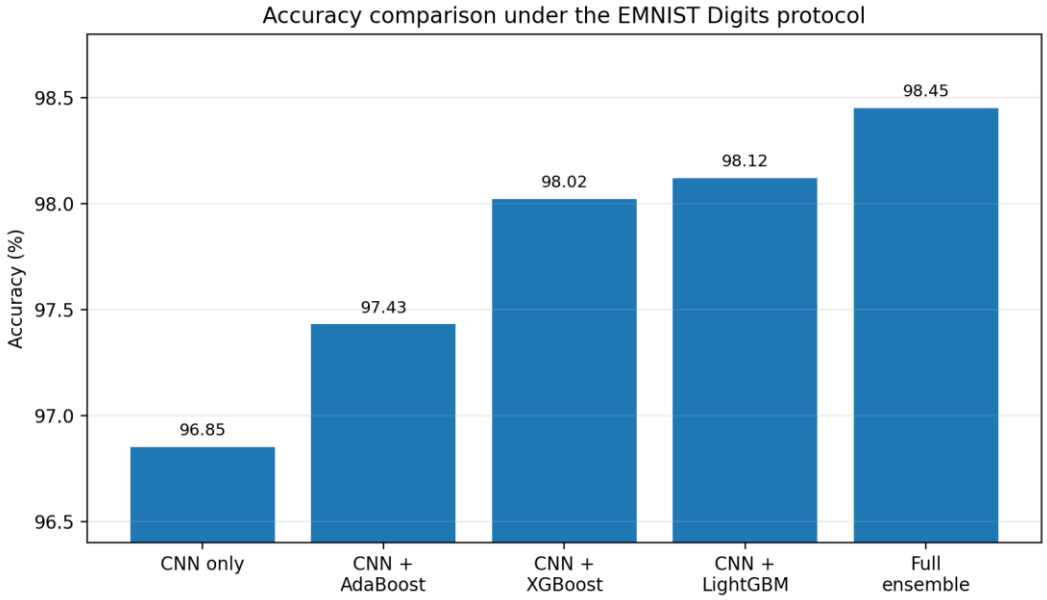

Figure 2. Accuracy comparison of the CNN and hybrid configurations.

5.3 Ablation and augmentation analysis

The removal of any of each of the classifiers resulted in a drop in performance, thus proving that it was not just one model that generated the voting result. If we exclude AdaBoost, it decreases accuracy by 0.25%, excluding XGBoost by 0.50% and excluding LightGBM by 0.65%. Within this implementation, it is LightGBM that made the biggest marginal contribution, but all three members were still needed for the best performance.

The ablation result is not indicative of LightGBM's ability to be better than all other performance; it is based on the selected representation, parameters and dataset. An augmentation effect was also noticeable. The accuracy without augmentation was 97.10% and the gap between accuracy at epoch-50 training and epoch-50 validation was 2.5 percentage points. The full set of rotation, scaling, translation and shear yielded an accuracy of 98.45% and brought the difference down to 0.7 percentage points. The result shows that controlling the morphological variability did help in increasing the performance of the in-domain generalization but it did not prevent the later external-domain degradation.

Configuration	Accuracy (%)	Macro F1	CPU latency (ms/image)
Full ensemble	98.45	0.984	150
Without AdaBoost	98.20	0.982	120
Without XGBoost	97.95	0.979	110

Without LightGBM	97.80	0.978	105

Table 5. Component ablation results.

5.4 Robustness and external-domain evaluation

Moderate synthetic corruptions resulted in less reduction than external domain transfer. The accuracy was lowered to 98.45% with gaussian noise and 97.82% in salt and pepper noise. The USPS result was 94.80%, while accuracy with CEDAR was 90.20%, which is 8.25 percentage points less than the accuracy achieved on the clean EMNIST test set. This comparison demonstrates that two levels of robustness to injected pixel corruption are not identical in terms of robustness to acquisition process change, background texture, resolution, writer distribution and document artefacts. This result is in line with the domain-adaptation theory, which suggests degradation occurs when the data distribution in the source domain differs from the data distribution in the target domain (Ben-David et al., 2010; Ganin et al., 2016).

The standalone CNN obtained 82.10% and the ensemble 85.30% in the case of $\varepsilon = 0.1$ under FGSM. The ensemble was thus more accurate, but still quite fragile in both. The outcome does not help with a broad conclusion of adversarial robustness or certified robustness, but rather a limited conclusion of relative robustness to the attacks considered.

Evaluation condition	Accuracy (%)	Macro F1	Interpretation
Clean EMNIST	98.45	0.984	In-domain reference
Gaussian noise, $\sigma = 0.1$	97.82	0.978	Moderate synthetic corruption
Salt-and-pepper, $p = 0.05$	97.53	0.975	Impulse-noise corruption
USPS external test	94.80	0.948	Moderate domain shift
CEDAR external test	90.20	0.902	Substantial document-domain shift
FGSM ensemble, $\varepsilon = 0.1$	85.30		Relative adversarial retention
FGSM CNN, $\varepsilon = 0.1$	82.10		Baseline adversarial retention

Table 6. Robustness and generalization results.

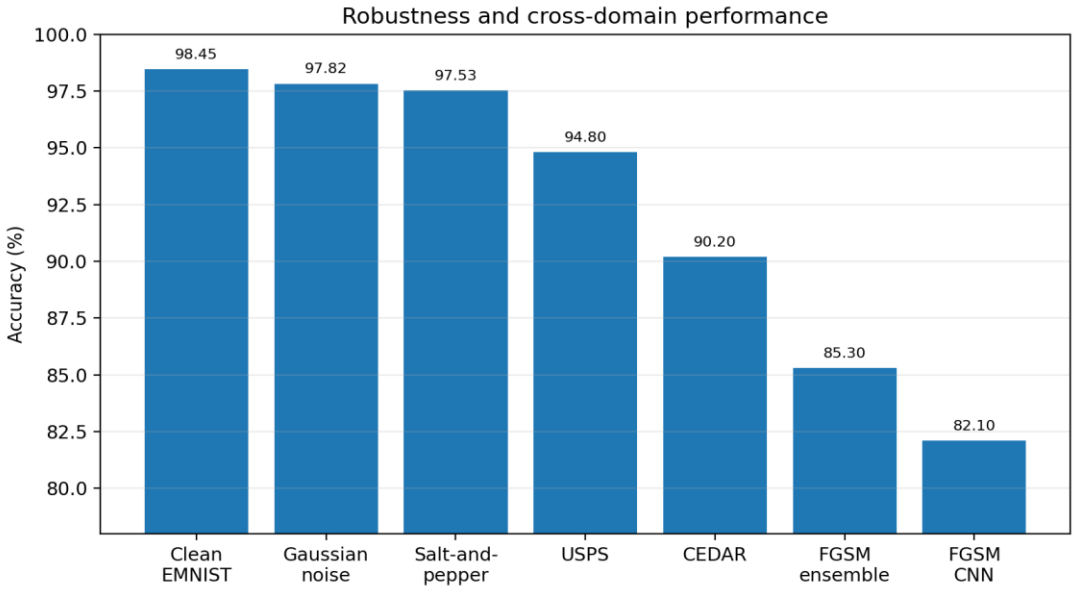

Figure 3. Performance under synthetic corruption, external domains and FGSM.

5.5 Computational profile

There was an increase in resource consumption but also an increase in accuracy with the gain. CNN was a stand-alone, which needed 10 minutes to train and 50ms/image CPU inference and 2.5GB memory in the reported environment. This took 45 minutes, 150 MS/image and 3.5 GB to complete the full ensemble. Therefore, an accuracy gained of +1.60 percentage points equated to a three-fold latency level as well as an additional memory of 1.0 GB. This profile is better suited to batch processing on the edge in a non-real-time environment, such as off-line processing or a server-side application, than to a true real-time environment.

Model	Training time (min)	CPU latency (ms/image)	Memory (GB)
Standalone CNN	10	50	2.5
CNN + AdaBoost	15	75	2.8
CNN + XGBoost	18	80	3.0
CNN + LightGBM	14	70	2.7
Proposed ensemble	45	150	3.5

Table 7. Computational profile in the reported hardware environment.

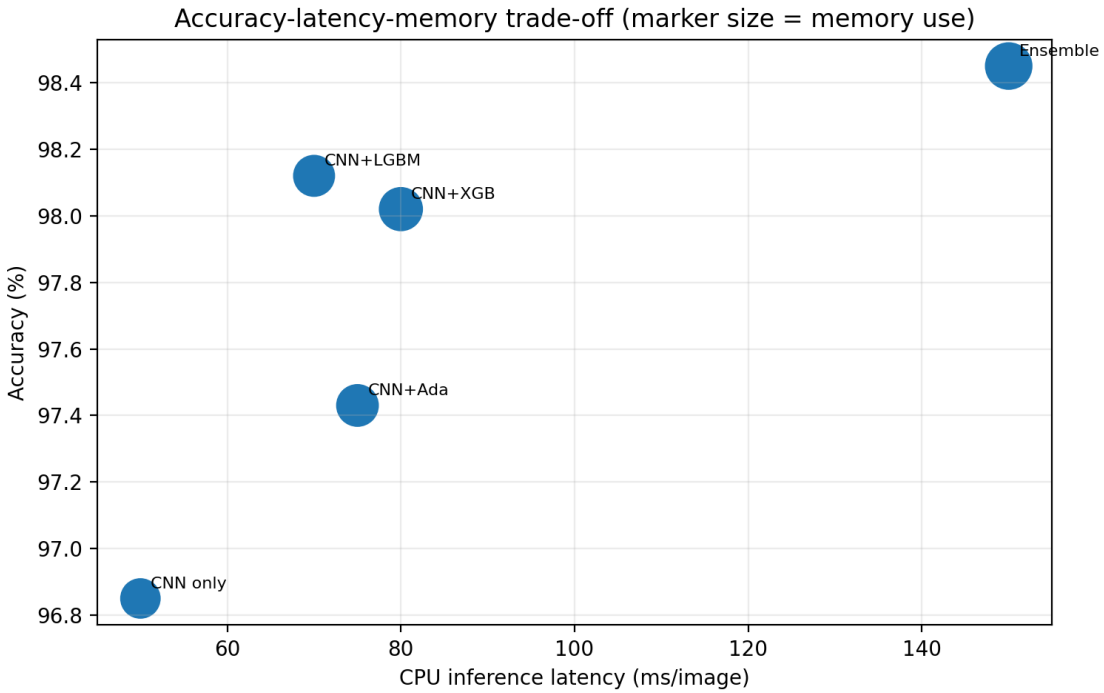

Figure 4. Accuracy-latency-memory trade-off across model configurations.

6. Discussion

6.1 Why feature decoupling improved the implemented baseline

The results reveal a gradual improvement in the neural baseline through single-classifier hybrids to the full ensemble. In Dense1, dense feature map representations in the CNN are compressed into tabular representations to make convolutional feature maps learnable by each boosting method without changing the CNN. Non-linear interactions between embedding dimensions can be modeled by XGBoost or LightGBM and AdaBoost keeps re-weighting difficult observations. They have distinct learning rules which predict complementary reduction in accuracy, as each was dropped.

The gain should be considered with respect to the implemented CNN and not as a guarantee that an increase in gain is always better than a neural head. The comparison may change if a deeper CNN or other loss function, attention or another larger transfer-learning architecture are used. It has been reported that CNN-VI and transfer learning has achieved good performance on digit recognition with different partitions and model capacity (Agrawal et al., 2024; Azawi, 2023). The present study is controlled by the matched comparison within one pipeline.

The rules of robustness, domain shift and operational meaning are followed. Robustness, domain shift and operational meaning rules are adhered to.

6.2 Robustness, domain shift and operational meaning

The ability to retain useful structure in the presence of relatively uncorrupted source distribution, as reflected in the smaller degradation under synthetic noise, indicates that augmentation and convolutional features retain some useful structure when corruption is relatively low. An even more serious weakness was identified when the system was tested by an outside group. The acquisition, preprocessing, background, resolution and document context are different for USPS and particularly CEDAR than for EMNIST. The CEDAR result shows that we cannot expect to have a high benchmark accuracy without domain validation.

The same was found in the FGSM experiment. The accuracy of tree-based decisions remained better than CNN baseline after perturbation; this could be explained by the fact that the final class was not generated by the attacked SoftMax head. However, this embedding was produced by the perturbed CNN and the ensemble did not improve by more than 13 percentage points over clean testing. Future work needs to focus on more wide-ranging attacks, adaptive attacks, calibration and abstention, not in the sense of presenting the ensemble as inherently secure.

6.3 Comparison with recent hybrid and multilingual work

The effectiveness of learned features in classification or feature fusion across Latin and regional numeral datasets has been seen in Hybrid CNN-SVM (Ahlawat & Choudhary, 2020), pretrained feature-fusion and attention-transfer systems (Bhatti et al., 2023; Fateh et al., 2024; Rajpal & Garg, 2023). The present framework is unique in the use of three boosting algorithms on a compact representation at the same time and because of the explicit measurement of ablation, adversarial retention, external-domain transfer, latency and memory.

Comparing directly to published accuracies with MNIST, EMNIST, Urdu, Hindi and multilingual datasets is misleading owing to the differences in class morphology, size of the datasets, number of writers and evaluation protocol. So, the proposed 98.45% should be considered as strong and competitive with the given EMNIST protocol. It isn't a statement of universal state-of-the-art performance that makes it truly scientifically valuable.

6.4 Practical implications

The reported latency is reasonable for asynchronous archival indexing, assisted form entry or server-side educational data capture when the digits to be captured have been segmented. These uses should have confidence thresholds, reject malformed inputs and log and review manually. For financial or healthcare deployments, they need more evidence because there can be a single digit mistake that alters the amount, identifier or dosage. The 90.20% CEDAR result is not adequate to trustfully interpret bank checks independently, without domain-specific retraining and human confirmation.

The ablation results indicate a lightened version of the proposed model (CNN + LightGBM) for latency-sensitive applications. This setup provided an accuracy of 98.12% and had lower latency than the entire ensemble. It is a practical illustration of the choice of an operating point, not assuming that the top score is the best system.

7. Limitations and Threats to Validity

There are six major limitations to the study. However, the main result was achieved on the EMNIST Digits set and the USPS and CEDAR sets provide a broader evaluation but are not samples of all writer groups, devices, document types or scripts. Second, the robustness analysis included only two kinds of synthetic corruptions and one FGSM setting, hence no claim of comprehensive or certified robustness can be made. Third, the statistical comparisons were based on the fold level and relied on the multiple observations in each of the overlapping cross-validation training folds; this can reduce the independence assumption. Fourth, the latency and memory results are implementation dependent and may vary due to batch size, processor, threading and serialization and library versions. Last, hard voting does not consider probability calibration and class-specific reliability. Sixth, embedding-level feature importance gives

only partial interpretability since, although a high ranked feature dimension is suggestive of a human-readable stroke region, it does not necessarily point to one.

Isolated digit classification is also a study of interest. To process a complete document would need to detect/segment the document, localise the fields, validate them in context and manage exception handling. The benchmark data lack sufficient metadata regarding the writers and do not allow for any cultural or demographic inferences.

8. Conclusion and Future Work

The feature-decoupled handwritten digit recognition framework was introduced in this paper, where a compact CNN is used to export a compact 256-dimensional representation to AdaBoost, XGBoost and LightGBM. The accuracy of the hard-voting ensemble under the reported EMNIST Digits protocol was 98.45% and the macro F1 score was 0.984, outperforming the implemented CNN by 1.60 percentage points and 0.984 macro F1 score. Single-classifier and ablation experiments indicated that the gain was due to the combination of classifiers and LightGBM gave the greatest marginal contribution among the selected classifiers.

These limits on this result have been set by the wider assessment. The model was able to tolerate moderate synthetic noise and maintained more accuracy than the CNN under the evaluated FGSM perturbation and more drastically on CEDAR. When all the pieces were combined, the CPU latency rose to 150 ms/image, while memory usage reached 3.5 GB. The evidence thus endorses its use in formats of accuracy-focused, offline or server-side processing and notes that real-time or high-stakes deployment is risky.

Further optimization of the clean-benchmark should come after domain adaptation and representative document information is integrated. The target hardware for evaluation should include quantization, structured pruning, knowledge distillation and reduced ensemble modes (Han et al., 2015; Jacob et al., 2018). The most important comparisons to be made with hard voting are with probability averaging, calibrated averaging and stacking and validation-weighted fusion. Other forms of adversarial evaluation, confidence-aware abstention, image-space explanations, multilingual datasets and writer-disjoint splits and prospective service-level testing are also required. Such developments would bring the validated prototype closer to creating an efficient, transparent and trusted proofreading system of documents.

Declarations

Funding: This research received no external funding.

Competing interests: The authors declare that they have no competing interests.

Ethics approval: Not applicable. The study used established benchmark image datasets and did not involve intervention with human participants or the collection of new personally identifiable data.

Data and code availability: EMNIST and USPS are publicly available benchmark datasets. CEDAR is subject to its applicable access and licence conditions. The implementation configuration, trained-model specifications and processed experimental outputs are available from the corresponding author on reasonable request.

Author contributions: Prashant Kumar: conceptualization, methodology, software, validation, investigation, data curation, visualization and writing original draft. Ragini Shukla: supervision, methodology, validation and writing review and editing. Both authors reviewed and approved the final manuscript.

Acknowledgements: The authors acknowledge the Department of IT & CS, Dr. C. V. Raman University, Bilaspur, India, for institutional support.

References

1. Agrawal, V., Jagtap, J., Patil, S., & Kotecha, K. (2024). Performance analysis of hybrid deep learning framework using a vision transformer and convolutional neural network for handwritten digit recognition. MethodsX, 12, 102554. https://doi.org/10.1016/j.mex.2024.102554
2. Ahlawat, S., & Choudhary, A. (2020). Hybrid CNN-SVM classifier for handwritten digit recognition. Procedia Computer Science, 167, 2554-2560. https://doi.org/10.1016/j.procs.2020.03.309
3. Azawi, N. (2023). Handwritten digits recognition using transfer learning. Computers & Electrical Engineering, 106, 108604. https://doi.org/10.1016/j.compeleceng.2023.108604
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. Machine Learning, 79, 151-175. https://doi.org/10.1007/s10994-009-5152-4

5. Bhatti, A., Arif, A., Khalid, W., Khan, B., Ali, A., Khalid, S., & Rehman, A. U. (2023). Recognition and classification of handwritten Urdu numerals using deep learning techniques. Applied Sciences, 13(3), 1624. https://doi.org/10.3390/app13031624
6. Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32. https://doi.org/10.1023/A:1010933404324
7. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785
8. Cireşan, D. C., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3642-3649. https://doi.org/10.1109/CVPR.2012.6248110
9. Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: Extending MNIST to handwritten letters. 2017 International Joint Conference on Neural Networks, 2921-2926. https://doi.org/10.1109/IJCNN.2017.7966217
10. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20, 273-297. https://doi.org/10.1007/BF00994018
11. Dietterich, T. G. (2000). Ensemble methods in machine learning. In J. Kittler & F. Roli (Eds.), Multiple Classifier Systems (LNCS 1857, pp. 1-15). Springer. https://doi.org/10.1007/3-540-45014-9_1
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2021). An image is worth 16 × 16 words: Transformers for image recognition at scale. International Conference on Learning Representations.
13. Fateh, A., Birgani, R. T., Fateh, M., & Abolghasemi, V. (2024). Advancing multilingual handwritten numeral recognition with attention-driven transfer learning. IEEE Access, 12, 41381-41395. https://doi.org/10.1109/ACCESS.2024.3378598
14. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139. https://doi.org/10.1006/jcss.1997.1504
15. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. Journal of Machine Learning Research, 17(59), 1-35.
16. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. International Conference on Learning Representations.
17. Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. Advances in Neural Information Processing Systems, 28.
18. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778. https://doi.org/10.1109/CVPR.2016.90
19. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.
20. Hull, J. J. (1994). A database for handwritten text recognition research. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(5), 550-554. https://doi.org/10.1109/34.291440
21. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, 448-456.
22. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., et al. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2704-2713. https://doi.org/10.1109/CVPR.2018.00286
23. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146-3154.
24. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1137-1143.
25. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. https://doi.org/10.1109/5.726791
26. Niu, X.-X., & Suen, C. Y. (2012). A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recognition, 45(4), 1318-1325. https://doi.org/10.1016/j.patcog.2011.09.021
27. Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.
28. Rajpal, D., & Garg, A. R. (2023). Ensemble of deep learning and machine learning approach for classification of handwritten Hindi numerals. Journal of Engineering and Applied Science, 70, 81. https://doi.org/10.1186/s44147-023-00252-2
29. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6, 60. https://doi.org/10.1186/s40537-019-0197-0
30. Simard, P. Y., Steinkraus, D., & Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. Proceedings of the Seventh International Conference on Document Analysis and Recognition, 958-963. https://doi.org/10.1109/ICDAR.2003.1227801
31. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. Advances in Neural Information Processing Systems, 25.
32. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002
33. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15, 1929-1958.

34. Srihari, S. N., Cha, S.-H., Arora, H., & Lee, S. (2002). Individuality of handwriting. Journal of Forensic Sciences, 47(4), 856-872.
35. van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579-2605.