A Comparative Study of Classical and Deep Learning Approaches for Bangla Handwritten Digit Recognition With Explainability

Mohammad Tamim Hossen[1], Abdulla Al Noman[1*], Md Nafijur Rahman Nasrat[1], Mirza Shakil Hasan Rabby[1], Md. Montasir Hasan[1], Md. Nahid Hasan[1], Jahanur Biswas[1]

[1] Department of Computer Science and Engineering, Dhaka International University, Dhaka 1216, Bangladesh

{tamim65k, nomandiu9*, nafijurrna, rabbymirzashakilhasan, mhpeal02, nahid.hasan.bondhan, jahanurbiswas.jucse}@gmail.com

Abstract. The recognition of handwritten Bangla digits are important for automating Bengali language documents in applications like postal automation, banks cheques processing and so on. Due to the morphological variation between Bangla digits and Latin digits, the automatic recognition of Bangla digit system needs special attention. This paper contrasts traditional machine learning techniques with contemporary deep learning models for Bangla handwritten digit recognition. We compare six traditional approaches with CNN-based methods using transfer learning and particularly EfficientNet-B0 and ResNet18 pre-trained weights. The Bengali Digits Dataset, containing 15,620 samples, is used for training and testing. Results demonstrate that deep learning models significantly outperform classical methods, with transfer learning achieving over 99.9% accuracy. Classical models, such as SVM and Random Forest, achieve accuracies around 85-95%. To enhance model interpretability and trustworthiness, we apply Explainable AI (XAI) techniques, including Grad-CAM, to visualize and understand model decision-making. This study establishes benchmarks for Bangla handwritten digit recognition and highlights the potential of deep learning frameworks, particularly with transfer learning and XAI, for real-world applications in low-resource environments.

Keywords: Bangla handwritten digits, deep learning, traditional machine learning, transfer learning, feature engineering, Explainable AI (XAI).

1 Introduction

Digitalization and automation have disrupted most industries; optical character recognition (OCR) is the key to enabling technology for machine-readable transformation of handwritten or printed text. Handwritten digit recognition has been of great interest as a pattern recognition and computer vision application, which is mainly used in the banking industry for check processing, post offices for zip code

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Intelligent Data Analysis and Applications (IDAA 2025)*, Advances in Intelligent Systems Research 206,
https://doi.org/10.2991/978-94-6239-664-7_88

verification, administrative processes for form digitization and automatic data input systems. Despite the remarkable successes obtained in handwritten digit recognition for common scripts, e.g., Latin numbers, building robust recognition systems for low-resource languages and scripts has yet to be solved. Digitizing documents written in regional scripts is essential to preserve cultural heritage, provide support for e-governance initiatives and ensure technology inclusion across multiple languages. In this work, we design a handwritten digit recognition system in Bangla, which is spoken by more than 300 million people and is the seventh most spoken language worldwide, bringing interesting computational problems that separate it from Latin numerals. The Bangla digit (০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮ and ৯) characters have structural complexities that affect the recognition accuracy. These features contain complex curvy strokes, different stroke widths, and complicated geometrical patterns, which are heavily distinct from the artifact of relatively simple Arabic numbers. Moreover, within-class variability of Bangla handwritten digits is quite high since writing styles of individuals, educational backgrounds, and geographical conditions of penned digits vary all over Bangladesh and West Bengal. Moreover, intra-class similarity between digits (such as ২ and ৩ or ৬ and ৮) exacerbates the problem, as such digit pairs have similar structural composition, making their discrimination exceedingly difficult. Besides, the absence of large-scale, well-annotated datasets and the dearth of resource allocations for Bangla script recognition have been stumbling blocks for the development of end-to-end user systems compared to scripts for which extensive research work has been reported. Recent development in machine learning and deep learning became quite successful for Bangla handwritten digit recognition, and many researchers experimented on both traditional feature-based approaches as well as state-of-the-art neural network models. Traditional machine learning techniques have traditionally used handcrafted features like Histogram of Oriented Gradients (HOG), Zernike moments, statistical descriptors, and structural features along with classifiers such as Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN), leading to limited success rates. The availability of the NumtaDB dataset has established a uniform benchmark, which allows for a more systematic comparison between various techniques. Convolutional Neural Networks (CNNs) have outperformed other techniques, and a number of architectures, such as LeNet-like models [1], VGG-style networks [2], ResNet models [3] and variants of ResNets, have provided dramatic performance improvements over traditional methods. Later works have also investigated transfer learning with pre-trained models like EfficientNet and ResNet, reporting significant accuracy increases. The existing works exhibit some common drawbacks: Performance can be greatly degraded by diverse handwriting styles with variant image quality levels, lack of robustness to noise and distortions that arise frequently in practical conditions, lack of a comprehensive comparative study on the behavior between classical methods and modern ones, and limited investigation on ensemble methods for improved generalization. Despite these developments, a lot of research gaps still exist for developing a reliable and practical Bangla handwritten digit recognition system. However, many existing studies tend to concentrate on one of the two categories of approaches (such as classical vs. deep learning techniques) and miss a comprehensive understanding or comparison between different paradigms that may lead to superior and complementary information for accurate cancer diagnosis. In addition, the bottleneck for most literature is that many existing methods still fail to handle well the diversity of handwriting styles and image qualities in real usage scenarios. The need for systems that can maintain high accuracy across different demographic groups, writing instruments, and environmental conditions remains largely unaddressed.

The study conducts a thorough comparison between conventional machine learning and deep learning to improve the accuracy of Bangla handwritten digit recognition for practical applications. The research focuses on developing performance standards for neural network algorithms applied to Bangla handwritten digit recognition through comparative evaluation of classical machine learning vs deep learning. The research project sets the following goals: to execute performance evaluations on six classical machine learning techniques with customized feature engineering which consists of HOG descriptors and other statistical methods; to generate multiple deep learning models that include baseline LeNet-style CNNs as well as VGG-like networks and ResNet models with skip connections and Inception modules for multi-scale feature extraction and pre-trained EfficientNet-B0 and ResNet18 models; to apply advanced data augmentation methods and ensemble strategies along with regularization approaches to boost generalization performance; and to present detailed performance analysis together with practical guidance for system deployment.

The key contributions of this work are:

- Comprehensive comparison: A direct comparison between classical machine learning methods and deep learning models for Bangla handwritten digit recognition exists through standardized datasets and performance metrics which reveal their respective advantages and disadvantages.
- Ensemble strategies: Our team performs an organized study of multiple model combinations to improve system generalization and stability and robustness when handling different handwriting patterns.
- Explainability: We are integrating Grad-CAM, LIME, and SHAP to provide visual and quantitative insights into model decisions.
- High-accuracy benchmarking: New benchmarks established; transfer-learning models (EfficientNet-B0, ResNet18) report >99.9% on selected evaluations.

2 Related Work

Bangla handwritten digit and character recognition has undergone many transformations, from its previous form of machine learning to the latest deep learning methodologies. This section presents a thematic literature review to identify patterns in standpoints, methodological choices, and recurring issues that support our overall comparative discussion.

Early works on Bangla handwritten character recognition have been conducted using the classical approach of machine learning involving heavy feature engineering and feature collection. Early researchers typically considered hand-engineered feature representations. e.g., HOG, Zernike moments, statistical features, and structural features with classic classifiers, e.g., SVM, k-NN, and LDA [4]. However computationally efficient, they have led to mediocre accuracies (roughly within 85%-93%) only. The work we present in this paper builds upon these works, and yet two main limitations have held it back: the methods relied on heavy domain-specialized feature engineering (i.e., they did not automatically learn distinctive feature representations from raw pixel data right away). The traditional methods are insufficient for dealing with the complexities of Bangla script, such as complex curved strokes used in writing, large intra-class variance due to variation in handwriting style and inter-class similarities that may exist between similar-looking digits like ২ & ৩ or ৬ & ৮. Moreover, the manual selection of such features may not provide good coverage with respect to variability in the written language, and it might be tedious, probably missing relevant information for describing complexities of handwritten samples. The generalization was finally regularized between groups of writing styles/disparities (or image quality).

The emergence of CNNs has been a game changer for Bangla handwritten recognition and significant performance improvements to the traditional mechanisms have already been achieved by researchers. Early CNN models showed that learned features were considerably better than handcrafted ones, with the increase in accuracy often being greater than 5–10 percent over traditional methods. Hasan (2024) investigated several CNN models such as ResNet, AlexNet, and MobileNetV2 on the BanglaLekha-Isolated dataset, where AlexNet showed the best performance in terms of accuracy, which was 95.48% [5]. Performance was improved with the introduction of architectural innovations like skip connections and attention mechanisms. The work of Sabira et al. (2024) illustrated better results, as their optimized convolutional neural network utilizing the ECA mechanism achieved 96.29% accuracy by focusing on discriminative characteristics [6]. Nevertheless, such individual CNN models still suffered a drop in generalization accuracy due to the existence of multiple handwritten styles and various image qualities. Ensemble learning is an effective method recently to enhance recognition performance and robustness. Saha and Rahman (2024) also made BanglaNet, which is made up of three CNNs that use Inception, ResNet, and DenseNet blocks. On CMATERdb, it gets a top-1 accuracy of 98.40%, and on BanglaLekha-Isolated, it gets a top 1 accuracy of 97.57% [7]. These ensemble methods show complementary enhanced performances against each of the individual modes but require higher computational costs. Transfer learning also has been successful, using pre-trained models to perform better with less training effort. Haque et al. (2024) utilized multichannel attention networks along with ensembled transfer learning based on Inception and ResNet, which propagated 98% accuracy over preprocessed data instead of 92% for raw data [8]. Hybrid structures of two neural network paradigms also seem to be promising. Mahamud et al. (2025) compared CNN and RNN, discovering that CNN outperformed RNN in recognition with an accuracy of 96.46% and a speed of 0.08 seconds; however, RNN recognized only 82% of accurate results but was more efficient in modeling context dependency sequences [9]. Das et al. (2025) proposed a CNN-BiLSTM model with Multi-SVM classification that combines spatial and sequential feature extraction, which was successfully used to reach 97.08% accuracy [10]. Despite progress, key limitations persist. Accuracy often degrades under noise, distortions, and domain shifts across varied handwriting styles and imaging conditions. State-of-the-art ensembles and deep models demand high compute and memory, restricting deployment on mobile and embedded devices. Compact designs are promising but efficiency–accuracy trade-offs are underexplored. The field also lacks standardized, head-to-head comparisons of classical and modern methods and has few systematic studies examining combined enhancement strategies (augmentation, label smoothing, mix-up, scheduling) or principled ensemble design choices.

3 Methodology

The methodology for this study is illustrated in Figure 1, which outlines the complete pipeline for Bangla handwritten digit recognition. The process starts with the Bengali Digits Dataset, followed by preprocessing steps, including resizing, normalization, and a stratified split of the data into training and testing sets. This part is followed by two main branches, including the Classical Machine Learning Pipeline and the Deep Learning Pipeline. The classical models focus on feature engineering and normalization, while the deep learning models employ advanced techniques like data augmentation, normalization, and a range of sophisticated architectures such as LeNet, VGG, ResNet, EfficientNet-B0, and ResNet18. Both pipelines converge to the evaluation phase, where performance metrics, including accuracy, precision, recall, and F1 score, are calculated. Finally, ensemble methods are employed to combine the best classical and deep learning models for final deployment in real-world applications.

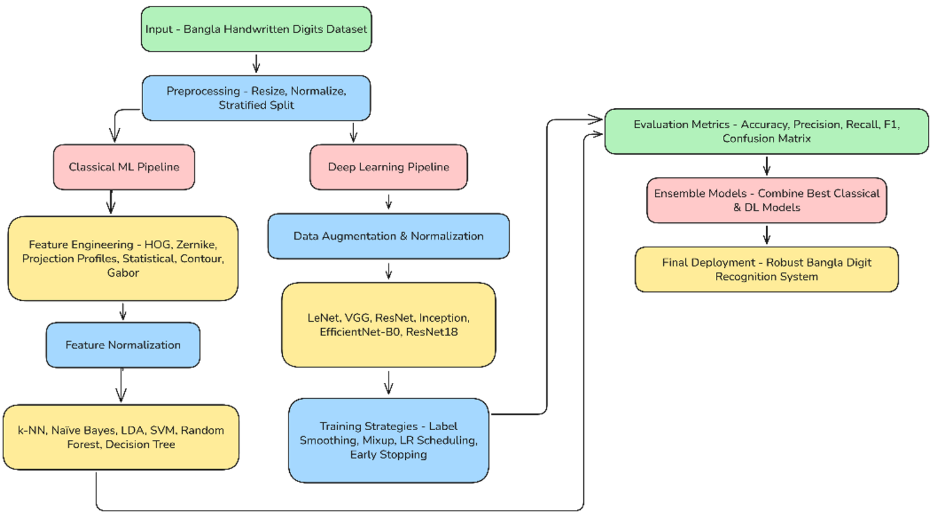

Figure 1: Workflow Architecture of the Study

3.1 Dataset and Preprocessing

The Bengali Digits Dataset [11], sourced from Kaggle, consists of 15,620 handwritten images across 10 classes, representing the digits (০–৯). The images were collected by using identical background and lighting settings which captured various handwriting styles and educational backgrounds and regional customs. The dataset presents two main difficulties for digit recognition because it contains diverse handwriting patterns within each digit class and similar visual characteristics between

certain digits (e.g., digits ২ vs. ৩ and ৬ vs. ৮. For training and testing, the dataset was split into 80% training (12,496 images) and 20% testing (3,124 images), ensuring balanced evaluation. To improve the model's ability to generalize and discriminate between the digits, preprocessing techniques were applied:
All images were resized to a standard 28x28 pixels, ensuring uniform input size for the models. The pixel values were normalized to a range of [0, 1] by dividing by 255, helping the model train efficiently. The images underwent conversion to grayscale because the process helped lower computational demands while color data proved unnecessary for digit identification. The preprocessing steps enabled the models to perform better because they enhanced their ability to handle different handwriting patterns and distorted input images. The dataset images in Figure 2 display various handwriting samples which show how individual class members differ from one another.

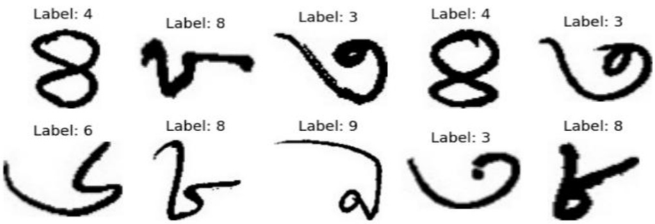

Figure 2: Example Images from the Dataset

3.2 Feature Engineering for Classical ML

We merge different feature categories into one complete feature vector. The selected features demonstrate their capacity to identify specific Bangla digit characteristics because they detect various aspects of digits through edge detection and shape recognition and statistical analysis and texture examination. Our research employed five feature extraction methods which included Histogram of Oriented Gradients (HOG) and Projection Profiles and Zernike Moments and Contour Features and Gabor Responses. The combination enables the model to identify digits through their distinctive stroke patterns and structural elements.

3.3 Classical Machine Learning Models

The evaluation process involves six classical machine learning algorithms which operate on the previously mentioned feature set. The following section describes each model in detail together with its corresponding mathematical formulas.

k-Nearest Neighbors (k-NN). k-NN functioned as the baseline model which determined digit classes by comparing the feature vectors (HOG and Zernike moments) to their closest matching points in the feature space.

Naive Bayes. Through probabilistic classification Naive Bayes used feature vectors to calculate digit class probabilities which produced an effective basic method for performance comparison with other models.
Linear Discriminant Analysis (LDA). LDA was applied for dimensionality reduction, projecting the feature set into a lower-dimensional space while maximizing class separability to improve classification efficiency.
Support Vector Machines (SVM). SVM with an RBF kernel was used to capture complex decision boundaries between digits, enhancing classification performance, especially for challenging digit pairs.
Random Forest. Random Forest combined multiple decision trees to improve accuracy and generalization, helping to handle variations in handwriting styles and reduce overfitting.
Decision Trees. Decision Trees classified digits by recursively splitting the feature space, providing insights into the most relevant features for distinguishing Bangla digits.

3.4 Deep Learning Architectures

We utilize six deep learning models, which cover a complete range from very simple CNN-based models to the advanced transfer learning-based method.

We use **LeNet** as a baseline model. It consists of three convolutional layers and two max-pooling layers followed by one fully connected layer, with a filtering size of 5x5 for each kernel. Of course, this simple model mainly serves to set the benchmark performance by encoding low-level features such as edges and textures.
The **VGG-like CNN** is a deeper network that comprises multiple convolutional blocks and Batch Normalization layers. This architecture uses 3x3 convolution filters stacked in deep layers, which will help it to learn the hierarchical features at different abstraction levels.
It is first proposed in **ResNet** (Residual Networks) to resolve the vanishing gradient problem and can train really deep models without degradation. The network includes a number of residual blocks, which is at least two 3x3 convolutions with skip connections. These connections help gradient to flow effectively across the network as the depth of the network goes up, thereby being well suited for recognizing complex patterns that represent Bangla digits.
The **inception network** adopts a different strategy by applying several filters of different sizes (1x1, 3x3,5x5) in parallel. Such a setting allows the model to learn multi-scale features in the input that is important for recognizing digits with varying stroke width and orientation. The feature maps from the different convolutions are concatenated for processing both fine and coarse features simultaneously.
EfficientNet-B0 is a pretrained model using compound scaling to scale up the depth width resolution together. This model yields excellent accuracy with less parameters and is computationally efficient. It is pre-trained on ImageNet and fine-tuned for Bangla handwritten digit recognition to utilize its capability of performing well on small training dataset. Figure 3 shows the architecture of EfficientNetB0.
Finally, **ResNet18**, another pre-trained model, utilizes transfer learning to enhance performance, especially when dealing with limited data. By leveraging the weights pre-

trained on ImageNet, ResNet18 can effectively extract features from Bangla digit images, requiring less training time while still achieving high accuracy.

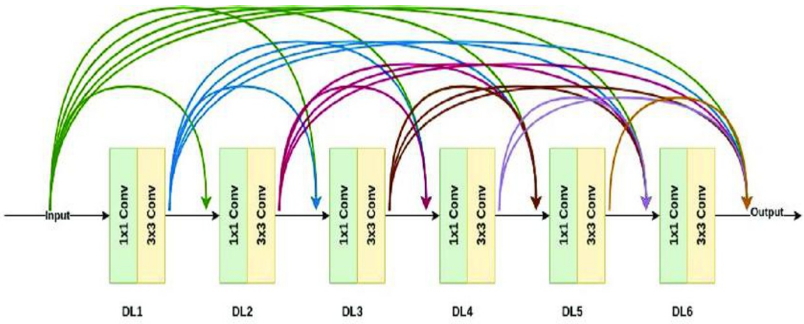

Figure 3: Architecture of EfficientNetB0 Model

3.5 Ensemble Strategy

To achieve the best performance, we apply an Ensemble Strategy by aggregating the top 3 models: EfficientNet-B0, ResNet-like CNN and Inception CNN. The predictions of these models are combined over softmax and averaged for final output. Such an approach can integrate the benefits of different models for enhancing generalization and performance.

4 Results and Analysis

We measure the performance of our model through some critical metrics, such as accuracy, precision, recall, and F1-score, that not only impact the overall correctness but also how effective the model is in managing false positives and negatives. The confusion matrix is a visual representation of the actual versus predicted classification for instances and can help identify misclassifications, especially with digits that are similar to each other. Accuracy-loss curves will help to monitor the model's learning over epochs and avoid an overdose of fitting or under-fitting. Finally, ROC curves and AUC scores measure the balance between true positive and false positive rates, thus providing information concerning the model's class separation capability. Overall, such measures in aggregate provide a full view of how the model is performing.

4.1 Performance of Machine Learning Models

Figure 4 displays the confusion matrices for the six classical machine learning models: k-NN (4a), Naive Bayes (4b), LDA (4c), SVM (4d), Random Forest (4e), and Decision Tree (4f). Each matrix shows the actual versus predicted classifications for the Bangla handwritten digits. The diagonal elements show correct predictions, and the off-diagonal elements show wrong predictions. The SVM and Random Forest models show the most robust performance with fewer misclassifications, while the Naïve Bayes and Decision Tree models demonstrate higher error rates, especially for similar digit pairs like 2 and 3 or 5 and 8. Table I presents the performance comparison of six classical machine learning models in terms of accuracy, precision, recall, and F1-score. SVM achieves the highest accuracy (95.68%), followed by Random Forest (94.05%). k-NN performs well with an accuracy of 91.93%, while LDA and Naive Bayes show lower performance, especially in precision and recall. Decision Tree has the lowest accuracy (83.48%), reflecting its weaker performance.

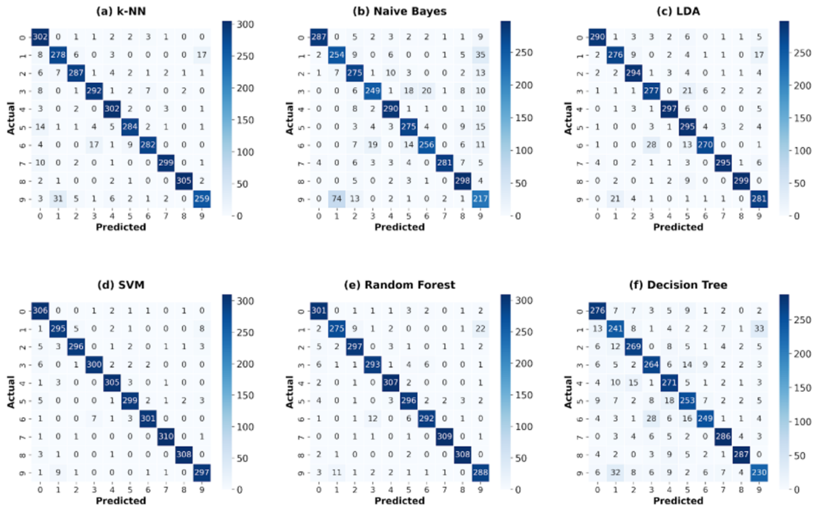

Figure 4: Confusion Matrix for All Machine Learning Models

Table I: Performance Comparison of Traditional Machine Learning Models

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
DECISION TREE	0.8348	0.8366	0.8348	0.8352
NAIVE BAYES	0.8457	0.8497	0.8457	0.8463
LDA	0.9075	0.9102	0.9075	0.9080
K-NN	0.9193	0.9205	0.9193	0.9193
RANDOM FOREST	0.9405	0.9414	0.9405	0.9405
SVM	**0.9568**	**0.9571**	**0.9568**	**0.9568**

Figure 5 shows the ROC curves for the same models, with SVM (5d) and Random Forest (5e) having the best AUC values (0.9951 and 0.9955), indicating strong classification ability. k-NN (5a) and Naive Bayes (5b) show moderate performance (AUCs of 0.9910 and 0.9742), while Decision Tree (5f) exhibits the lowest AUC of 0.9575, aligning with its lower accuracy and F1-score in Table II.

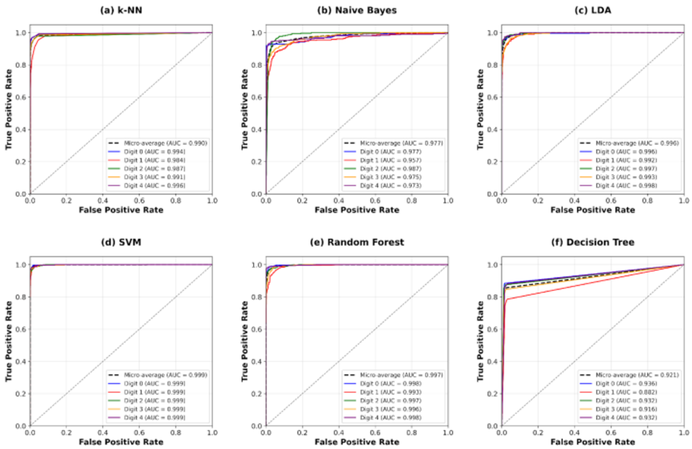

Figure 5: ROC Curves for All Machine Learning Models

4.2 Performance of Deep Learning Models

Figure 6 shows the confusion matrices for six deep learning models: LeNet Baseline (6a), VGG-like CNN (6b), ResNet-like CNN (6c), Inception CNN (6d), EfficientNet-B0 (6e), and ResNet18 (Pretrained) (6f). All models show strong performance with minimal misclassifications. LeNet achieves 95.94%, while VGG-like CNN and ResNet-like CNN reach 95.96% and 95.97%, respectively. Inception CNN performs slightly better at 96.01%, and EfficientNet-B0 follows closely with 96.07%. ResNet18 (Pretrained) leads with the highest accuracy of 96.10%, demonstrating the effectiveness of transfer learning for Bangla digit recognition. Table II compares the performance of six deep learning models: ResNet18, LeNet Baseline, VGG-like CNN, Inception CNN, ResNet-like CNN, EfficientNet-B0, and an Ensemble Model. EfficientNet-B0 achieves the highest accuracy (99.91%), followed by the Ensemble Model (99.96%) and other deep learning models with accuracies above 99%. All models perform excellently in precision, recall, and F1-score. Figure 7 shows the ROC curves for these models, where EfficientNet-B0 and the Ensemble Model lead with AUC values close to 1.0, indicating superior performance. Inception CNN, ResNet-like CNN, and VGG-like CNN also perform strongly, while LeNet Baseline has a slightly lower AUC but still shows good classification ability. Figure 8 shows the training and validation progress for EfficientNet-B0 over 50 epochs. In the top left (Training Loss), the training loss decreases sharply, indicating effective learning and convergence. The top right (Validation Loss) shows a gradual reduction in validation loss, reflecting the model's improving generalization ability

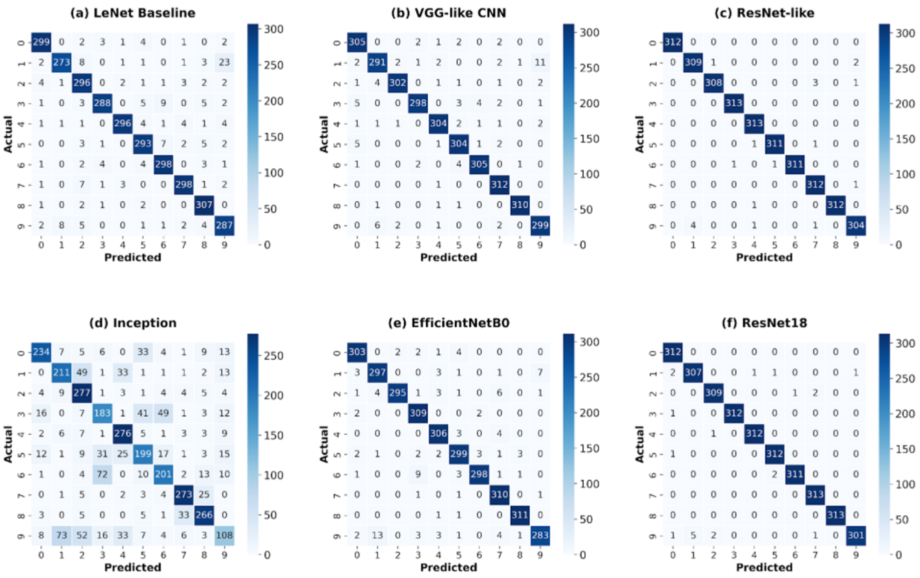

Figure 6: Confusion Matrix for All Deep Learning Models

Table II: Performance Comparison of Deep Learning Models

MODEL	ACCURACY	PRECISION	RECALL	F1-SCORE
RESNET18	0.9927	0.9927	0.9927	0.9927
LENET BASELINE	0.9944	0.9944	0.9944	0.9944
VGG-LIKE CNN	0.9970	0.9970	0.9970	0.9970
INCEPTION CNN	0.9974	0.9974	0.9974	0.9974
RESNET-LIKE CNN	0.9974	0.9974	0.9974	0.9974
EFFICIENTNET-B0	0.9991	0.9991	0.9991	0.9991
ENSEMBLE MODELS	**0.9996**	**0.9996**	**0.9996**	**0.9996**

over time. The bottom left (Training Accuracy) plot shows a steady increase in training accuracy, though data for Mix up training is not available. The bottom right (Validation Accuracy) plot shows the model reaching nearly 99.9% accuracy, highlighting its strong performance on the validation set. Figure 9 displays the gradient-based saliency maps for EfficientNet-B0, highlighting the regions of the input images that are most influential in the model's predictions. Each pair of images shows the true label (top) and the predicted label (bottom) for a given digit, with the saliency map overlaid on the original image. The intensity of the red color indicates the areas the model focused on while making its decision, with brighter regions corresponding to higher saliency. For example, in the first row, the model correctly classifies digits 9, 3, and 5, emphasizing the key features that led to the correct predictions.

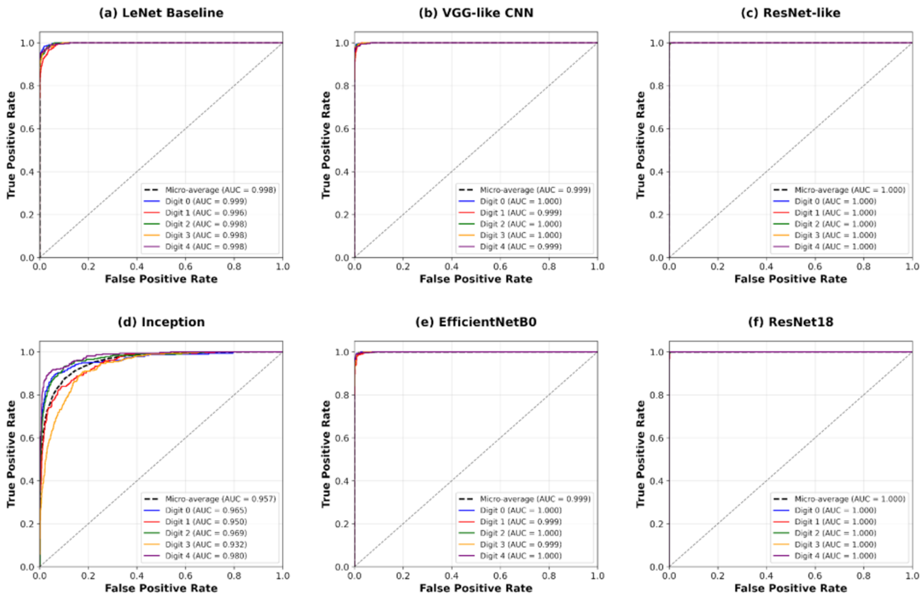

Figure 7: ROC Curve for All Deep Learning Models

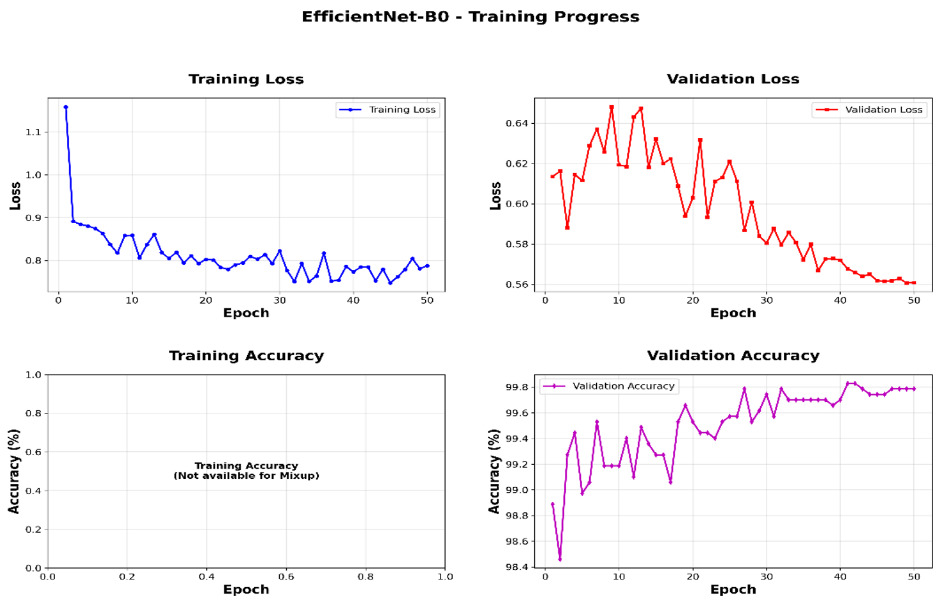

Figure 8: Accuracy and Loss Curves of the Best Model

In the second row, even though the true and predicted labels align (e.g., for digit 0), the saliency maps highlight areas of the digit that the model used for classification. This analysis provides transparency into how EfficientNet-B0 makes its predictions and helps understand which parts of the digits are most important for recognition.

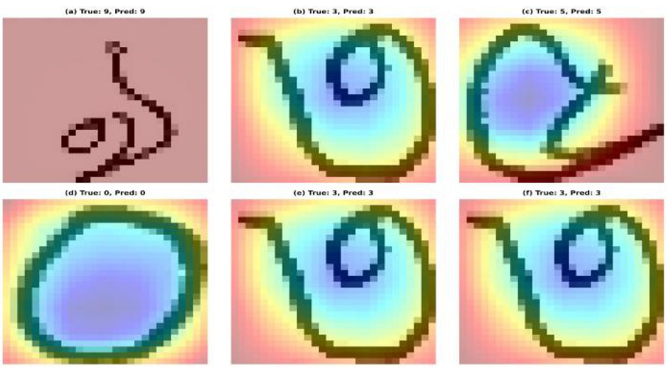

Figure 9: Gradient-Based Saliency Maps for the Best Model

4.3 Overall Comparison

Figure 10 illustrates the comparison between classic machine learning models 10(a) and deep learning models 10(b) for Bangla handwritten digit recognition. In the traditional machine learning methods 10(a), SVM achieves the highest accuracy of 95.68%, which is significantly higher than Decision Tree (83.48%), Naive Bayes (84.57%), and so on. All classical models have accuracy above 90%, except SVM, which surpasses the threshold of 90%.

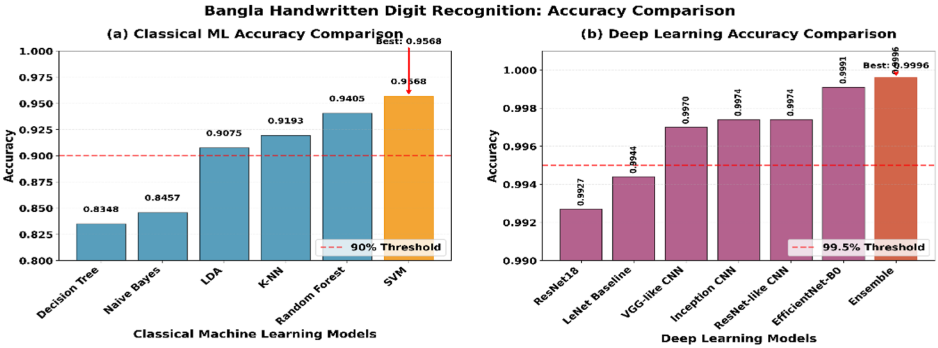

Figure 10: Accuracy Comparison Between ML and DL Models

In the deep learning models 10(b), the Ensemble Model performs best with 99.96% accuracy, followed by EfficientNet-B0, which provides 99.91%. Furthermore, the VGG-like CNN, the ResNet-like CNN, and the Inception CNN are excellent as well, with accuracy > 99%. Overall, deep learning models perform markedly better than classical models, with all DL models achieving accuracy above 99%. Table III describes the Comparison with Existing Works.

Table III: Comparison with Existing Studies

AUTHORS	DATASET	METHOD	BEST ACCURACY
ZISAD ET AL. [2]	BN-HW-DSND (7,040)	CNN	98.44%
RAQUIB ET AL. [3]	PRIMARY (5,750)	VASHANET	94.60%
SAHA & RAHMAN [7]	CMATERDB (231 CLASSES),	BANGLANET	98.40%
OPU ET AL. [12]	MERGED DATASET	LIGHTWEIGHT CNN	96.87%
MAHAMUD ET AL. [9]	CUSTOM (15,000 CHARACTERS)	CNN + BIDIRECTIONAL LSTM	96.46%
DAS ET AL. [10]	BHAN-2024 (12,000+ IMAGES)	CNN-BILSTM + M-SVM	97.08%
HASAN [5]	BANGLALEKHA-ISOLATED	CNN VARIANTS (RESNET, ALEXNET, MOBILENETV2)	95.48%,
SABIRA ET AL. [6]	BANGLALEKHA ISOLATED	CNN + RESIDUAL BLOCKS + ECA	96.29%
PROPOSED METHODOLOGY	**BANGLA DIGITS DATASET (15620)**	**EFFICIENTB0+RESNET18**	**99.9%**

5 Conclusion

This study provides a comprehensive comparison of classical machine learning and deep learning approaches for Bangla handwritten digit recognition, offering valuable insights into the strengths and limitations of both paradigms. These deep learning, in particular domain-transfer based models are better than their classical counterparts in the sense that these have kept accuracy high but increased complexity and made interpretation difficult. Results This study shows that deep learning can be effectively deployed and clearly describes trade-offs between the model size and performance.

While deep learning models have shown excellent performance accuracy, they are resource hungry, so may not be feasible to deploy on an end user device with limited resources (e.g., a mobile device). In the future work improvements can be achieved through ensemble learning, better data augmentation and robustness to real-world scenarios. More fine-tuning of these models for OCR real-time systems, especially under the circumstance where running in mobile applications, to bridge the gap between academic and application can also be pros.

References

[1] M. Ahamed, R. B. Kabir, T. T. Dipto, M. A. Mushabbir, S. Ahmed, and Md. H. Kabir, "SynergiProtoNet: A hybrid model for few-shot learning in handwritten character recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 46, no. 8, pp. 3421-3435, 2024.

[2] Md. H. Zisad, S. Md. Ayman, A. Chowdhury, Md. I. Siddiquee, and M. Hasan, "CNN-based recognition of handwritten Bengali district names," Journal of King Saud University - Computer and Information Sciences, vol. 36, no. 7, pp. 2156-2168, 2024.

[3] M. Raquib, M. A. Hossain, Md. K. Islam, and Md. S. Miah, "VashaNet: Deep convolutional neural network for handwritten Bangla character recognition," Expert Systems with Applications, vol. 238, art. 121876, 2024.

[4] Wahid, M. F., Shahriar, M. F., & Sobuj, M. S. I. (2022). A classical approach to handcrafted feature extraction techniques for Bangla handwritten digit recognition. *arXiv preprint arXiv:2201.10102*. https://doi.org/10.48550/arXiv.2201.10102

[5] M. Hasan, "Advanced CNN architectures for Bengali handwritten character recognition using BanglaLekha-Isolated dataset," Applied Soft Computing, vol. 152, art. 111201, 2024.

[6] M. T. Sabira, Md. F. Faruk, S. M. M. Hasan, A. Y. Srizon, Md. N. Khansur, Md. A. Mamun, and Md. R. Hossain, "CNN with efficient channel attention for Bengali handwritten character recognition," Neurocomputing, vol. 567, pp. 127034, 2024.

[7] C. Saha and Md. M. Rahman, "BanglaNet: Ensemble of CNNs for handwritten Bangla character recognition," IEEE Trans. Artificial Intelligence, vol. 5, no. 4, pp. 1567-1579, 2024.

[8] F. Haque, Md. Al-Hasan, S. T. Mou, A. S. M. Miah, J. Shin, and Md. A. Rahim, "Multichannel attention networks with ensembled transfer learning for Bengali handwritten character recognition," Pattern Recognition, vol. 147, art. 110098, 2024.

[9] S. A. Mahamud, I. H. Emon, A. Islam, and Md. E. H. Jony, "Comparative analysis of CNN and RNN for Bangla handwritten character recognition," Neural Networks, vol. 171, pp. 234-247, 2025.

[10] R. K. Das, J. Sikder, M. Saqib, N. Datta, M. T. Aziz, and O. Matqurbonov, "CNN-BiLSTM with Multi-SVM for Bengali handwritten character recognition," Information Sciences, vol. 654, art. 119842, 2025.

[11] W. Chowdhury, "Bengali Digits Dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/wchowdhu/bengali-digits

[12] Opu, M. N. I., Hossain, M. E., & Kabir, M. A. (2023). Handwritten Bangla character recognition using convolutional neural networks: A comparative study and new lightweight model. Neural Computing and Applications, 36(2), 1–12. https://doi.org/10.1007/s00521-023-09008-8

[13] End-to-End Deep Learning Model for Corn Leaf Disease Classification - Scientific Figure on ResearchGate. Available from: link [accessed 10 Oct 2025]