An Efficient Deep Learning Model for Bangla Handwritten Digit Recognition

1st A K M Nihalul Kabir
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
akm.nihalul.kabir@g.bracu.ac.bd

2nd Md. Shahadat Hossain Shamim
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
shahadat.hossain.shamim@g.bracu.ac.bd

3rd Md. Rumman Shahriar
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
md.rumman.shahriar@g.bracu.ac.bd

4th Md. Asif Hasan Biplob
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
asif.hasan.biplob@g.bracu.ac.bd

***Abstract*—This paper introduces a low-latency and high-performance deep learning model for the recognition of handwritten Bangla digits. Recognizing handwritten digits in Bengali, which is the fifth most widely spoken language all over the world and its automated digit recognition is essential to digital archiving and automation. Nonetheless, the calligraphic variability is significant, which makes an adequate classification challenging, and existing approaches tend to handle one or the other variable of accuracy at the expense of the other, computational efficiency, making them impractical to apply. Convolutional Neural Network (CNN) models in the presence of a performance bottleneck, when used in fine-grained class domains with images of size 32×32 pixels. We alleviate some of these limitations by proposing a custom-designed Convolutional Neural Network (CNN), termed *EfficientCNN*, specifically designed for the 32×32 pixel images of the BHaND dataset. The model will demonstrate high efficiency in terms of processing with achieving the similar task accuracy in comparison to state-of-the-art alternatives. The unique aspect of our proposed architecture is the use of a Global Average Pooling layer instead of the traditional Flatten layer, which significantly reduces the parameter count of the model without compromising its feature-learning capabilities. The suggested model achieves a test accuracy of 99.78%. Benchmarking analysis of the model depicts that it performs considerably better as compared to the established models, including AlexNet, LeNet-5, and MobileNetV2-0.5, also offers an excellent balance between model size, predictive accuracy, as well as inference speed. This convergence of the latest accuracy with computational cost-effectiveness makes the model especially suited to deployment in resources constrained, real world applications.**

***Index Terms*——Bangla, handwritten digit recognition, convolutional neural network, deep learning, BHAND dataset, computational efficiency.**

I. Introduction

The automatic recognition of handwritten text is a cornerstone of modern Optical Character Recognition (OCR) systems that allows physical documents to become an integrated part of the digital working environment. This kind of technology is unavoidable in automating data entry, scanning of historic archives, and delivering financial services. The importance of the development of correct and efficient OCR becomes paramount where the Bengali language is used by more than 265 million people. However, the very writing properties of Bengali script, such as the complicated shape of writing their digits, wide-ranging styles of writings and cursive relatedness, and form resemblance of some numerals pose significant challenges to recognition systems.

The first machine learning attempts to deal with problems of handwriting recognition were noted in generalization to the broad scope of human scripts. Ever since, deep learning and in this situation, Convolutional Neural Networks (CNNs) have led to a pleasing extent of enhancement. Whilst these models have led to significant improvements in scores, most modern state-of-art models are computationally intensive, hence limiting their applicability in highly resource-constrained settings, such as mobile devices and continuous inference, real-time systems.

This research is expected to bridge the gap between high accuracy and low computation by constructing both accurate and efficient deep learning models. The principal objective is to design, implement, and test a novel Convolutional Neural Network (CNN) to recognize Bangla handwritten digits specifically on the standardized BHaND dataset.

The main contributions of this paper are fourfold:

- **A Novel and Efficient Architecture:** *EfficientCNN* is architecture-specific and has been developed with focus on parameter-efficiency goals: it uses Global Average Pooling, which has become a common option in modern network architecture, and has also been optimized to work within the 32×32 BHaND dataset.
- **State-of-the-Art Accuracy:** The current model achieves a test accuracy of 99.78%, hence setting a state-of-the-art performance on a broadly adopted and posterior benchmark.
- **Comprehensive Benchmarking:** We provide a methodological comparative assessment of *EfficientCNN* with

the seminal and highly efficient convolutional architectures—AlexNet, LeNet-5, and MobileNetV2—in terms of a variety of performance indicators such as classification accuracy, the number of parameters, inference latency, and model size.

- **Demonstrated Robustness:** We confirm the reliability of the model's excellent generalization and lack of overfitting using the percentage-point gap (Δpp) metric, confirming its reliability.
- **Structure of the Paper:** Section II provides a review of the relevant literature. Section III details the dataset and our proposed methodology, including the model architecture and training protocol. Section IV describes the experimental setup and implementation. Section V presents and discusses the experimental results. Finally, Section VI concludes the paper and suggests directions for future work.

II. Literature Review

The that is in search of precise and effective handwritten-digits recognition is an old problem of pattern recognition and computer vision. In the case of the Bengali language development has been has been very swift, shifting out of classical machine-learning method- The pre-polish physics to the advanced deep-learning re-imagining of the present day. dominate. Our work fits into that trajectory, as this review shows. by looking at (i) the movement of traditional approaches to deep learning, (ii) CNN architectures evolution, (iii) the the pivotal position of standardized datasets and (iv) recent high technologies and feasibility.

A. *From Traditional Methods to Deep Learning*

Old Bangla-digit reading machine software was based on conventional machine-learning pipelines [3], [11]. These models re- quired massive manual feature engineering traction e.g. Histogram of Oriented Gradients (HOG) or Local Binary Patterns (LBP) [12]—then vintage classifiers such as SVMs or k-NN [17]. Those methods were constrained by the fact that they require manually engineered features, which had difficulty in reproducing high variability and non-linear patterns of handwritten Bangla numerals [2], [11]. Various forms of writing, cursive interrelations and structural ambiguities were particularly troublesome to rigid feature-based systems [3]. Convolutional Neural Networks (CNNs) have changed the face of Neural Network research. CNNs removed manual feature extraction by learning hierarchical represen- tations directly from raw pixels: crude edges and curvature are captured by initial layers, while more advanced structures record complicated numbering [12]. This end- to-end learning constantly performed better compared to traditional techniques across datasets and metrics [1], [5], [6], defining deep learning as the de facto standard for character recognition [15], [18].

B. *Evolution of CNN Architectures for Bangla Digits*

The field of architecture design soon became one of the points of study. Bayanno-Net (2019) [4] revealed that even a compara- tively modest CNN having three convolutional and three max- pooling layers, in addition to dropout regularization [23], may accomplish 96.5% accuracy. Alom et al. [5] took performance a step further by augmenting CNNs with Gaussian and Gabor filters; the Gabor- improved version achieved 98.78% accuracy on CMATERdb 3.1.1. Gabor filters are texture and edge sensitive at several scales and orientations, complementing CNN feature learning. More recently, Sikder et al. [7] proposed an optimized, task-specific CNN yielding 99.44% validation accuracy on the BHaND dataset. Their tailor-made architecture surpassed universal versions like AlexNet [15] and Inception V3, emphasizing one of the major threads of our work: carefully customized architectures can outperform off-the-shelf networks for niche tasks [1], [7].

C. *The Foundational Role of Standardized Datasets*

The significant role of standardized datasets is foundational. Rapid development has been possible due to large, uniform datasets, which (i) provide the data volume deep networks require and (ii) enable fair benchmarking [2], [11], [17]. Major resources include:

CMATERdb – the originating benchmark for early CNN studies [5], [25].

NumtaDB – a large-scale collection considered the Bangla analogue of MNIST [13], [21]; essential for training generalizable models [24].

ISI dataset – often used in cross-dataset comparisons to test robustness.

BHaND – the dataset for our work [13]; its size and quality have enabled recent state-of-the-art results, such as Sikder et al.'s 99.44% accuracy [7].

The wide use of multiple datasets has stimulated cross-dataset validation, where models trained on one corpus are tested on others [11]. Such protocols better illustrate generalization to real-world data obtained from diverse sources.

D. *Advanced Techniques and Practical Considerations*

In addition to architecture design, scholars have adopted advanced techniques. Sikder et al.'s semi-supervised GAN (SGAN) generates realistic synthetic digits to address data imbalance and scarcity [7]. At the same time, practical deployment demands computational efficiency. Recently, research has focused on "lightning-fast" models for embedded and mobile devices [8]. This trend balances precision with constraints on parameter count, memory foot- print [19], and inference speed. Designing parameter-efficient networks and applying optimization techniques ensure that high-accuracy models can be used in real-world applications for the 265 million Bengali speakers [2]. Our study situates itself at the intersection of performance and efficiency.

III. METHODOLOGY

A. *Dataset*

This paper uses the Bengali Handwritten Numerals Dataset (BHaND), a well-established dataset designed to resemble MNIST in structure while capturing the unique characteristics of Bangla numerals. It contains 70,000 grayscale images of the ten digits (0 through 9), split into 50,000 training, 10,000 validation, and 10,000 testing samples. Each image has dimensions of 32 x 32 pixels, with pixel intensities scaled to the range [0,1]. To enhance model learning efficiency, pixel values were inverted so that the digit strokes appear as 1 (black) and the background as 0 (white). In its raw form, each sample is represented as a flattened 1024-dimensional vector and assigned an integer label corresponding to one of the 10 digit classes. For CNN-based processing, these vectors are reshaped into tensors of shape (32, 32, 1). The dataset is compressed using GZIP into a single pickle file, enabling fast loading and storage efficiency. This structure makes it suitable for deep learning experiments and comparative benchmarking against other datasets.

B. *Proposed EfficientCNN Architecture*

The EfficientCNN was specifically designed to maximize classification accuracy while minimizing computational cost, particularly suited for small grayscale images. It takes an input tensor of shape (32, 32, 1) and processes it through four convolutional blocks with increasing filter sizes: 32, 64, 128, and 256. Each block contains two convolutional layers with 3x3 kernels, ReLU activations, and same padding, followed by batch normalization to ensure stable training. Spatial dimensions are reduced using MaxPooling2D with a 2x2 kernel, and dropout with a rate of 0.25 is applied to mitigate overfitting. After the last convolutional block, a Global Average Pooling layer significantly reduces the number of parameters and improves robustness to spatial variations. The network then passes through two fully connected layers: first a Dense layer with 512 neurons followed by batch normalization, then a second Dense layer with 256 neurons also followed by batch normalization and dropout with a rate of 0.5. The final output layer consists of 10 softmax neurons for multi-class classification. The architecture has a total of 1,442,154 parameters, with 1,439,658 being trainable.

C. *Training Strategy and Data Augmentation*

Training combined data augmentation with adaptive learning techniques to prevent overfitting and enhance generalization. The Adam optimizer was used with an initial learning rate of 0.001, and sparse categorical cross-entropy served as the loss function. Real-time augmentation of training images included:

- Random rotations up to ±10 degrees
- Horizontal and vertical shifts up to 10%
- Shear transformations up to 0.1
- Zoom adjustments up to 0.1

Training stability was maintained using callbacks:

- `ModelCheckpoint` monitored validation accuracy to save the best model
- `ReduceLROnPlateau` reduced the learning rate by half if validation loss plateaued for 5 epochs
- `EarlyStopping` halted training after 10 stagnant epochs and restored the best weights
- `ExponentialLR` applied a 5% learning rate decay every epoch

Training was capped at 50 epochs but stopped early at epoch 40, restoring the best model from epoch 30.

D. *Experimental Environment and Implementation*

Experiments were conducted on Google Colaboratory with GPU acceleration. The implementation utilized TensorFlow/Keras for deep learning, NumPy for numerical computations, Matplotlib and Seaborn for visualization, Scikit-learn for evaluation metrics, and OpenCV for image preprocessing. The entire workflow was encapsulated within a `BanglaDigitRecognizer` class containing methods to create, train, evaluate, visualize, and save models. The process consisted of loading the dataset, applying augmentation, training with callbacks, evaluating performance, visualizing results, and saving trained weights.

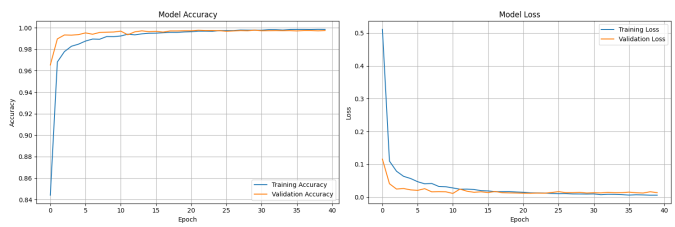

Fig. 1. Model Training History.

Figure 1 shows the model training history, depicting the progress of training and validation accuracy and loss over epochs. The smooth convergence and absence of major overfitting indicate the effectiveness of the training strategy and augmentation techniques.

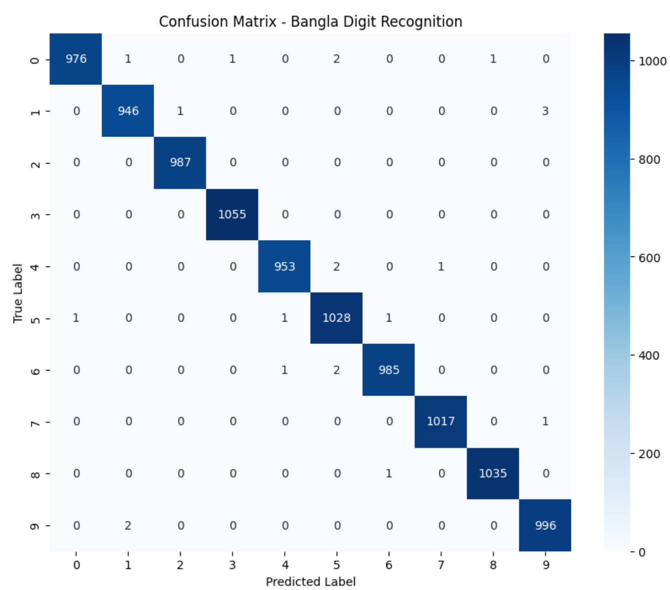

Fig. 2. Confusion Matrix.

Figure 2 presents the confusion matrix on the test set, illustrating the classification performance across all digit classes.

High accuracy and low misclassification rates demonstrate the model's robustness and discriminative capability.

E. *Benchmarking Procedure*

To evaluate performance, EfficientCNN was benchmarked against three well-established models:

- LeNet-5, a classic lightweight CNN
- AlexNet, a deeper classical network adapted for small images
- MobileNetV2 (alpha = 0.5), an efficient architecture optimized for low-resource environments and Android deployment

All models were trained on the same dataset and evaluated on the 10,000-image test set. Metrics used for comparison included classification accuracy, number of parameters, inference latency (milliseconds per 1,024 images), and model size in megabytes. This benchmarking highlighted EfficientCNN's balance between accuracy and computational efficiency, making it suitable for deployment in resource-constrained environments.

IV. Results and Discussion

The experimental findings illustrate that the efficiency and accuracy of our suggested *EfficientCNN* are superior to those of other well-known models, proving its effectiveness.

A. *Model Performance*

The *EfficientCNN* model achieved a Test Accuracy of 99.78% and Test Loss of 0.0114 when tested on the 10,000 unseen images of the BHaND test dataset. The high accuracy of this model proves its ability to recognize the complex and distinctive features of the ten Bangla digits.

B. *Analysis of Training Dynamics*

The history of training, as visualized in the "Model Accuracy" and "Model Loss" plots, gives us an idea about the learning process.

- **Accuracy Plot:** Both the validation accuracy and training accuracy curves exhibit an initial steep increase during the early epochs, then a gradual stabilization to a plateau level of over 99%. The fact that these two curves are close to one another during the training session means that the model was able to avoid overfitting and generalize well from the training data to the validation data.
- **Loss Plot:** The training and validation loss curves show a steep decrease at the beginning, ultimately settling at very low levels. Validation loss is not only low but also steady, pointing to the significant effectiveness of the aggressive regularization methods used (Batch Normalization, Dropout, and data augmentation). The maximum validation accuracy of 0.99783 was reached at epoch 30, after which the model weights were restored through the EarlyStopping callback.

C. *Classification Analysis*

Class-by-class detailed analysis validates the strength of the model. The classification report showed high precision, recall, and F1 scores for all digits, with the majority of the metrics above 0.996. For instance, Digit "2" and Digit "3" both had perfect recall values of 1.0000, i.e., all instances of these digits in the test set were correctly classified. The confusion matrix is a graphical affirmation of the same, with an approximate perfect diagonal line, which suggests that instances of misclassification between different digits were extremely rare.

D. *Comparative Benchmark Discussion*

The comparison with the benchmark is at the heart of our novelty argument and provides the complete justification of the superiority of our model for the task in question. The findings are summarized in Table I.

TABLE I
Benchmark Results on BHaND

Model	Accuracy	Parameters	Latency (ms)	Size (MB)
EfficientCNN	**0.9978**	**1,442,154**	**0.724**	**16.63**
AlexNet	0.9925	5,686,858	0.613	65.16
LeNet-5	0.9896	82,826	0.350	0.99
MobileNetV2-0.5	0.9824	719,040	5.033	8.59

- **Higher Accuracy:** Our *EfficientCNN* has the best accuracy of 99.78%, beating all the other models.
- **Efficiency vs. Large Models:** Our model is not just more accurate than AlexNet by 0.53 percentage points but also far more efficient. It takes up just 25.4% of the parameters (1.4M vs. 5.7M) and has a file size that is 75% less (16.6MB vs. 65.2MB). This makes it much more viable for deployment.
- **Lightweight vs. Performance Models:** LeNet-5 is the lightest in terms of size and parameters, but at 98.96% accuracy, it is considerably lower. This shows that while extreme lightness is possible, there is a compromise with lower performance.
- **Validation against Specialized Efficient Models:** One of the most significant results is the comparison with MobileNetV2-0.5. Our custom architecture is much more accurate at 99.78% compared to 98.24%, and it is significantly faster in inference latency at 0.724 ms vs. 5.033 ms. This strongly supports our approach, showing that a carefully designed, task-specific architecture can outperform well-known general-purpose efficient models.

E. *Generalization and Robustness*

In order to have a quantitative measure of the generalization capability, we computed the percentage-point gap (Δpp) between the ultimate training accuracy and the ultimate validation accuracy. Our calculation, with an 80/20 division of the training data to replicate the conditions of the training run, yielded a Δpp of -0.07. As per a practical rule of thumb, a model is said to generalize perfectly when the absolute value of Δpp is below 0.3, with no obvious indications of overfitting.

This very small value provides strong statistical indication of our model's solidity.

V. CONCLUSION AND FUTURE WORK

The paper has highlighted the effective design, implementation, and stringent verification of EfficientCNN, a deep learning model for handwritten Bangla digit recognition that achieves both state-of-the-art accuracy and high computational efficiency. In our model, the test accuracy achieved was 99.78% on the BHaND dataset, which we have demonstrated to be superior to that of several existing architectures.

The main contribution of this work lies in showing that a well-engineered CNN, tailored for a particular task, can outperform both larger, traditional architectures such as AlexNet and modern optimized efficient models like MobileNetV2. Important architectural choices, particularly the use of Global Average Pooling to decrease parameterization, were instrumental in achieving an optimal performance–resource consumption trade-off. The model's outstanding generalization, as proven by quantitative analysis, further highlights its dependability and readiness for practical use.

The implications of these findings are significant. The EfficientCNN model is a strong candidate for deployment in real-world systems where both speed and accuracy are critical. This includes applications in the banking and financial services sector for automating cheque processing, integration into mobile OCR programs, and large-scale governmental and commercial document digitization projects.

Future Research Directions

- **Advanced Architectures:** Future work might involve exploring the use of newer and more advanced architectures such as Vision Transformers (ViT) or models incorporating attention mechanisms, which have shown excellent promise in other computer vision domains.
- **Lack of Data and Flexibility:** Exploring few-shot learning and meta-learning methods may result in models that are more adaptable to rare or new styles of writing, or that can be fine-tuned on novel datasets with very small training samples.
- **End-to-End Systems:** One major research challenge is the development of end-to-end systems capable of processing entire documents both text and numerals seamlessly, thereby extending capabilities beyond isolated digit recognition.
- **Adverse Conditions:** Further research could focus on increasing the model's resilience to real-world degradations such as noise, low-quality scans, or aged documents, to ensure high performance outside of clean and standardized dataset conditions.

VI. ACKNOWLEDGMENT

The authors express their sincere gratitude to Dr. Md. Ashraful Alam, Associate Professor, Department of Computer Science and Engineering, BRAC University, for his invaluable guidance, insightful feedback, and continuous support throughout the course of this research. The authors also thank Shammo Biswas, Research Assistant, Department of Computer Science and Engineering, BRAC University, for his technical assistance and constructive suggestions, which significantly contributed to the development of this work.

REFERENCES

[1] M. Shopon, N. Mohammed, and M. A. Hashem, "Bangla handwritten digit recognition using an efficient CNN architecture," in *Proc. 2nd Int. Conf. Adv. Inf. Commun. Technol. (ICAICT)*, Dhaka, Bangladesh, Nov. 2020, pp. 317–322, doi: 10.1109/ICAICT50962.2020.9310966.

[2] A. K. Sarker, N. A. Urmi, and M. A. R. Ahad, "A comprehensive survey on Bangla handwritten numeral recognition," *IEEE Access*, vol. 9, pp. 77936–77954, 2021, doi: 10.1109/ACCESS.2021.3083989.

[3] A. Al-Waisy, M. Al-Fahdawi, S. Mohammed, and A. Al-Dulaimi, "A survey on Bangla handwritten numeral recognition," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–36, 2018, doi: 10.1145/3234151.

[4] S. A. Chowdhury and T. K. Bhowmik, "Bayanno-Net: Bangla handwritten digit recognition using convolutional neural networks," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Kolkata, India, Jun. 2019, pp. 1–6, doi: 10.1109/TENSYMP46218.2019.8971167.

[5] S. Z. Alom, P. P. Das, K. C. Tumpa, and M. A. M. Hasan, "Handwritten Bangla character recognition using a deep convolutional neural network," *arXiv:1705.02680*, 2017.

[6] M. I. H. Bhuiyan, M. A. H. Akhand, and M. A. M. Hasan, "A comprehensive study on Bangla handwritten digit recognition," *Khulna Univ. Stud.*, vol. 19, no. 1, pp. 1–13, 2022.

[7] A. Sikder, S. A. Chowdhury, and F. A. Siddique, "SGAN-BD: Semi-supervised generative adversarial network for Bangla digit recognition and generation," *arXiv:2103.07905*, 2021.

[8] S. Z. Alom, M. M. Rahman, and M. S. Hossain, "A lightning-fast approach to Bangla character recognition," *arXiv:2403.13465*, 2024.

[9] S. Yasmin, S. A. Chowdhury, and T. K. Bhowmik, *Bangla Handwritten Digit Recognition – Report*, Scribd, 2022.

[10] S. Basu, P. Mitra, and M. K. Kundu, "Handwritten Bangla digit recognition using a deep belief network," *arXiv:1806.02452*, 2018.

[11] S. U. Laskar, M. A. Z. Hridoy, and M. A. I. Bhuiyan, "A comprehensive study on Bangla handwritten digit recognition," *arXiv:2101.05081*, 2021.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[13] S. Chowdhury, "BHaND – Bengali handwritten numerals dataset," GitHub repository, 2020. [Online]. Available: https://github.com/SaadatChowdhury/BHaND

[14] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York, NY, USA: Academic, 1963, pp. 271–350.

[15] M. M. Rahman, M. A. H. Akhand, S. Islam, P. C. Shill, and M. M. H. Rahman, "Bangla handwritten character recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 8, pp. 42–49, 2015. [Online]. Available: https://www.mecs-press.org/ijigsp/ijigsp-v7-n8/v7n8-5.html

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.

[17] P. P. Acharjee, T. K. Bhowmik, S. A. Chowdhury, and S. Yasmin, "A study on handwritten Bangla numeral recognition," *Int. J. Adv. Res. Comput. Sci.*, vol. 11, no. 6, pp. 1–6, 2020.

[18] R. Das, S. K. Ghosh, and M. A. M. Hasan, "A comparative analysis of machine learning techniques for Bangla handwritten digit recognition," in *Proc. 3rd Int. Conf. Sustainable Technol. for Industry 4.0 (STI)*, Dhaka, Bangladesh, Dec. 2021, pp. 1–6, doi: 10.1109/STI53254.2021.9677004.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[21] M. A. A. Al-Qaness, A. A. Ewees, H. Fan, and M. A. Elaziz, “An optimized model for handwritten Arabic digit recognition based on deep learning,” *J. King Saud Univ. – Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4349–4356, Jul. 2022, doi: 10.1016/j.jksuci.2020.10.020.

[22] T. Truth, ”BanglaMNIST dataset,” Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/truthr/banglamnist

[23] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980*, 2014.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[25] M. R. Islam, M. S. Uddin, and M. A. M. Hasan, “A deep learning approach for NumtaDB dataset,” B.Sc. thesis, Dept. of Computer Science and Engineering, BRAC Univ., Dhaka, Bangladesh, 2018. [Online]. Available: http://dspace.bracu.ac.bd/xmlui/handle/10361/8870

[26] S. Pal, U. Pal, and P. P. Roy, “Handwritten Bangla character recognition using deep convolutional neural network,” *Int. J. Appl. Pattern Recognit.*, vol. 5, no. 1, pp. 1–15, 2018, doi: 10.1504/IJAPR.2018.10018292.

[27] N. Kabir, M. S. H. Shamim, M. R. Shahriar, and M. A. H. Biplob, “[FINAL]_CSE463_Group_3_spring_’25.ipynb,” GitHub, 463_project repository, Apr. 2025. [Online]. Available: https://tinyurl.com/3mdyar6f. [Accessed: May 26, 2025].