DOI: 10.12700/AIS.2025.001

Comparative Analysis of Lightweight CNNs for Multilingual Handwritten Character Recognition

Gaye Ediboglu Bartos
Alba Regial Faculty
Obuda University
Szekesfehervar, Hungary
gaye.ediboglu@uni-obuda.hu
0000-0002-6235-0650

Serel Ozmen Akyol
Department of Computer Engineering
Eskisehir Osmangazi University
Eskisehir, Turkey
sozmen@ogu.edu.tr
0000-0002-5344-4065

Abstract— Handwritten character recognition remains a fundamental problem in pattern recognition, with applications ranging from digital document processing to historical manuscript digitization. Multilingual handwritten character recognition introduces additional challenges due to variations in scripts, diacritics, and handwriting styles. In this work, we explore lightweight convolutional neural networks (CNNs) for recognizing handwritten characters across multiple languages using the merged T-H-E dataset, which combines visually identical upper- and lower-case characters into 54 classes. Four CNN architectures, as LeNet-5 (baseline), Simple CNN, Depthwise-CNN, and MiniVGGNet, are evaluated based on their efficiency, number of parameters, training time, and misclassification behavior. Experiments were conducted under standardized hyperparameters and using a fixed random seed to ensure a fair comparison. Our analysis shows that lightweight networks can achieve competitive performance while requiring minimal computational resources, and the study provides insights into common misclassification patterns related to visually similar characters and diacritics. These findings establish a fast and resource-efficient baseline for multilingual handwritten character recognition on small binary datasets.

Keywords—deep learning, handwritten character recognition, convolutional neural network, lightweight CNNs

I. Introduction

Handwritten character recognition (HCR) is a core problem in document analysis and pattern recognition. Automated recognition of handwritten characters enables applications such as digital document processing, historical manuscript digitization, postal sorting, and educational tools. While recognition of digits and single-language alphabets has been extensively studied, multilingual HCR introduces additional challenges due to variations in scripts, handwriting styles, and diacritics.

Deep learning, a subfield of machine learning, involves multilayered neural networks that can learn complex patterns in large datasets. This enables the development of artificial intelligence systems that surpass human performance in tasks such as image and document recognition [1].

Deep learning, and in particular convolutional neural networks (CNNs), was introduced in 1998 by French researcher Yann LeCun as an extension of the neocognitron model [2] and have become the standard approach for HCR. Classical architectures such as LeNet-5 [3] have demonstrated strong performance on MNIST [4], and extensions such as EMNIST [5] provide large-scale benchmarks for handwritten letters. Lightweight CNNs have shown that small networks can achieve competitive accuracy with significantly fewer parameters [6], [7]. For multilingual datasets, efficient models are particularly valuable because large pretrained networks often fail on small binary datasets, as confirmed in experiments with MobileNet on the T-H-E dataset [8].

Multilingual datasets, such as T-H-E, extend this concept by including multiple alphabets and special characters, providing a benchmark for evaluating models on English, Hungarian, and Turkish scripts. The merged version of T-H-E reduces ambiguity by combining visually identical upper- and lower-case characters, resulting in 54 classes.

In this study, we evaluate four CNN architectures on the merged T-H-E dataset: LeNet-5 (baseline), Simple CNN, Depthwise-CNN, and MiniVGGNet. All models were trained using a fixed random seed and early stopping to ensure reproducibility and fair comparison. We analyze not only accuracy but also training efficiency, parameter requirements, and patterns of misclassification among visually similar letters and diacritics. By comparing these architectures, we aim to identify accurate, fast, and resource-efficient networks suitable for multilingual handwritten character recognition in low-resource settings.

II. Releated Works

Handwritten character recognition (HCR) has been extensively studied, particularly for Latin scripts. Early approaches relied on feature extraction combined with classical classifiers such as k-nearest neighbors or support vector machines [9], [10]. With the advent of deep learning, convolutional neural networks (CNNs) became the dominant method, starting with LeNet-5, which achieved state-of-the-art results on MNIST [3], [4]. Subsequent works extended CNNs to larger and more complex datasets, including EMNIST, which includes both letters and digits [5]. EMNIST offers multiple splits such as ByClass, ByMerge, Letters, and Balanced, providing a range of challenges for character recognition systems.

Other MNIST-like character datasets include KMNIST (Kuzushiji-MNIST), which consists of 28×28 grayscale images of classical Japanese cursive characters [11]. These datasets provide benchmarks for evaluating CNNs on non-Latin scripts and demonstrate the importance of handling diverse character shapes and writing styles. The T-H-E dataset [8] expands this concept by providing a multilingual dataset with 54 classes, merging visually identical upper- and lower-

case characters, thus reducing misclassification and class imbalance.

A deep learning model using convolutional neural networks (CNNs) for multilingual handwritten character recognition covering English, Hindi, and Bengali scripts achieved 92.47% accuracy, outperforming individual single-language models [12]. This recent study emphasized tuning hyperparameters and activation functions to improve performance and demonstrated its effectiveness on multiple publicly available datasets.

Offered a model for multilingual character recognition, introducing a robust convolutional recurrent neural network (CRNN) model to accommodate the vagaries of both Tamil and English handwritten character recognition [13]. The proposed CRNN architecture includes a convolutional neural network, a long-term short-term memory network, and a gated recurrent unit for feature extraction. Using a comprehensive dataset of more than 60,000 images, the model distinguishes Tamil alphabet characters with 92% and English alphabet characters with 95% recognition rate.

Multilingual handwritten character recognition presents additional challenges due to diverse scripts, diacritics, and handwriting styles. Existing approaches include classical CNN architectures trained from scratch, as well as attempts to apply transfer learning from pretrained models such as MobileNet or ResNet [14], [15]. However, transfer learning often fails on small, binary datasets due to differences in input channels, image resolution, and feature distribution. Our experiments confirm this limitation, with MobileNet achieving only ~30% accuracy on the merged T-H-E dataset.

Lightweight CNN architectures have recently gained attention for their efficiency and suitability for low-resource environments. Small networks with few convolutional layers, including LeNet-5 (baseline) [3], Depthwise-CNN [14], MiniVGGNet [16], and simple CNNs [17], provide fast training and reasonable accuracy on MNIST, EMNIST, and KMNIST [18]. Such models have been applied to real-time OCR on mobile devices and embedded vision applications [19], [20].

Using small CNNs is particularly advantageous for the T-H-E dataset because the input images are binary, 28×28 pixels, and the dataset size is moderate, reducing overfitting while enabling rapid experimentation. Inspired by these works, we evaluate multiple lightweight CNNs on the merged T-H-E dataset, achieving competitive accuracy with minimal computational cost.

III. Material and Methods

This section describes the dataset, experimental setup, model architectures, and evaluation metrics employed in this study. The goal is to provide sufficient detail for reproducibility and critical evaluation of the methods used for multilingual handwritten character recognition. The dataset preparation includes merging and preprocessing steps designed to reduce class ambiguity while preserving representational integrity across English, Hungarian, and Turkish scripts. Experiments were standardized across multiple lightweight CNN architectures to compare their accuracy and computational efficiency under consistent training conditions. Evaluation metrics focus on performance at both overall and per-class levels, with attention to common misclassification patterns and resource use.

A. Dataset

For this study, we used the merged and augmented version 5 of the T-H-E dataset [8]. This version merges visually identical upper- and lower-case characters, reducing the total number of classes to 54 and minimizing misclassifications caused by case differences. The dataset contains 152,000 binary (pixel values 0 or 255) handwritten character samples spanning English, Hungarian, and Turkish scripts. Each image is 28×28 pixels.

The dataset was loaded from CSV, normalized to the [0,1] range, reshaped to 28×28×1 for CNN input, and one-hot encoded for classification. A stratified 90/10 train-test split ensured balanced representation across classes.

B. Experimental Setup

All experiments were conducted on Google Colab using a T4 GPU. To ensure reproducibility across models and runs, random seeds were fixed for NumPy, TensorFlow, and Python's random module. All images were binary and 28×28 pixels, and no color channels or grayscale normalization beyond scaling to [0,1] were applied.

Hyperparameters were standardized across all models to ensure a fair comparison: a learning rate of 3×10^{-a}, batch size of 128, and early stopping with patience of 8 epochs, restoring the best weights. Models were trained from scratch using categorical cross-entropy loss and the Adam optimizer, with up to 100 epochs. A 10% validation split from the training set was used for monitoring early stopping. Training time per model was recorded to compare computational efficiency.

C. Baseline Model

LeNet-5 [3] serves as a classical benchmark for handwritten character recognition. Its architecture includes two convolutional layers with average pooling, followed by fully connected layers and a softmax classifier for the 54 merged character classes. LeNet-5 provides a baseline reference in terms of both accuracy and computational efficiency on small binary images.

D. Lightweight CNN Architectures

The Simple CNN is a minimal, two-convolution-layer architecture designed for speed and efficiency. Each convolutional layer is followed by batch normalization and max pooling, and the dense head consists of a single fully connected layer with 128 units and a dropout of 0.5 before the softmax output. This architecture is intended as a fast and low-parameter reference.

Depthwise-CNN [7] employs three blocks of depthwise separable convolutions with 32, 64, and 128 filters. Max pooling follows the first two blocks, while global average pooling follows the third block. A dense layer of 128 units with dropout of 0.3 precedes the softmax output, maintaining a lightweight yet expressive architecture suitable for small binary inputs.

MiniVGGNet is a deeper variant inspired by VGG16 but smaller in scale. It consists of two convolutional blocks with 32 and 64 filters, each block containing two convolutional layers, batch normalization, max pooling, and dropout. The dense head has 512 units with batch normalization and dropout, followed by a softmax output. While larger than

LeNet-5, Depthwise-CNN, and the Simple CNN, MiniVGGNet provides stronger representational power at the cost of increased parameters and longer training time.

To ensure a fair comparison, all architectures were trained using the same hyperparameters and fixed random seed.

E. Metric Evaluation

Confusion matrices were computed to visualize per-class performance and identify patterns of misclassification. Models were evaluated on the held-out test set using accuracy, precision, recall, and F1-score [21].

Accuracy measures the overall proportion of correct predictions among all instances, providing a straightforward estimate of the model's general correctness.

Precision quantifies the fraction of true positive predictions over all positive predictions, thus indicating the model's reliability in identifying positive instances.

Recall (or sensitivity) measures the ability of the model to find all relevant positive cases by calculating the ratio of true positives to all actual positives.

The F1-score, the harmonic mean of precision and recall, balances these two metrics to provide a single value reflecting model performance in cases where class imbalance or asymmetric costs of errors may exist.

The top misclassified class pairs were analyzed to reveal challenges in multilingual character recognition, particularly for visually similar characters and merged upper- and lower-case letters. Training times were recorded to allow direct comparison of computational efficiency versus accuracy across the four architectures.

IV. RESULTS

The proposed lightweight CNN models were evaluated on the merged T-H-E dataset, consisting of 54 classes obtained by merging visually identical upper- and lower-case characters. The dataset comprises binary handwritten character images, which were converted to floating-point format and reshaped into 28×28×1 arrays. Stratified train/test splits were applied to maintain class distribution across all sets. To ensure a fair comparison, the same random seed (42) was used for all experiments, making the input splits and training initialization identical. Early stopping with a patience of 8 epochs and restoration of the best weights was applied to all models, allowing efficient training while preventing overfitting.

Table 1 summarizes the parameter count, training time, and test accuracy for the four models. Simple CNN and Depthwise-CNN achieved competitive accuracy with minimal computational cost, while MiniVGGNet, a deeper network, achieved the highest accuracy but required almost double the training time. LeNet-5, the classical baseline, had the fewest parameters but longer training time, highlighting the trade-offs between model depth, parameter efficiency, and training speed in small binary datasets.

TABLE I. NUMBER OF PARAMETERS, TRAINING TIME AND TEST ACCURACY OF THE MODELS

Model	Parameters	Training Time (s)	Test Accuracy (%)
Simple CNN	427,702	119.669	91.70
MiniVGGNet	3,091,302	238.387	94.30
Depthwise-CNN	287,204	131.93	92.80
LeNet-5	62,938	363.453	89.50

The results show that lighter architectures achieve impressive accuracy at a fraction of the training time of deeper networks. Depthwise-CNN, with the smallest number of parameters among high-performing models, demonstrates that depthwise separable convolutions efficiently capture discriminative features while remaining computationally light.

Simple CNN achieves nearly the same accuracy as Depthwise-CNN but with slightly more parameters and slightly shorter training time. LeNet-5, despite its simplicity, requires longer training, likely due to fewer convolutional filters and less expressive capacity for capturing subtle variations in handwritten characters.

Detailed analysis of misclassifications reveals systematic patterns in the errors, particularly among visually similar letters or those with diacritic marks. Certain merged or upper/lower-case characters, such as é/e, w/W, ı/i, and ó/Ó, were frequently confused across all models. These misclassifications are likely due to insufficient learning of subtle strokes and diacritics in binary images, compounded by the inherent ambiguity in merged classes.

TABLE II. ANALYSIS OF MISCLASSIFICATIONS

True Class	Most Frequent Misclassification	Count	Notes on Visual Similarity
8 (h)	10 (j)	12	Descender similarity
13 (m-M)	32 (ü-Ü)	15	Upper/lowercase merging causes ambiguity
17 (q)	36 (Q)	15	Subtle differences in tail shapes
20 (t)	59 (T)	18	Capitalization confusion
28 (ğ)	53 (Ğ)	15	Diacritic not consistently learned
34 (é)	34 (O)	17	Handwriting variations resemble O
8 (h)	27 (ç)	12	Descenders/hooks create confusion
30 (ş-Ş)	8 (h)	14	Stroke similarity in compact handwriting

These misclassification patterns indicate that while lightweight CNNs efficiently learn general shapes and coarse features, fine-grained distinctions, particularly diacritics and merged letters, remain challenging. MiniVGGNet shows some improvement in resolving these subtle distinctions, but at the cost of longer training and a threefold increase in parameters. The use of a fixed random seed across all experiments ensures that these comparisons are fair, as all models were exposed to the same training and validation splits.

Overall, these results demonstrate that lightweight CNNs, such as Simple CNN and Depthwise-CNN, provide a highly

efficient and reasonably accurate baseline for multilingual handwritten character recognition. While deeper networks can slightly improve accuracy, the efficiency gains of compact models make them especially suitable for low-resource environments or rapid prototyping on binary handwritten datasets. The misclassification analysis further provides insights into which character groups present the greatest challenge, guiding potential improvements such as targeted data augmentation or refined preprocessing for diacritics.

V. Conclusion

In this study, we explored the performance and efficiency of four CNN architectures LeNet-5, Simple CNN, Depthwise-CNN, and MiniVGGNet on the merged T-H-E multilingual handwritten character dataset. By using a fixed random seed across all experiments, we ensured a fair comparison of training dynamics, parameter efficiency, and predictive performance. Our results show that lightweight architectures such as Simple CNN and Depthwise-CNN achieve competitive test accuracy while requiring minimal computational resources and training time, making them highly suitable for small binary datasets and low-resource environments.

Detailed analysis of misclassifications revealed that visually similar characters, merged upper- and lower-case letters, and diacritics remain challenging across all models. While MiniVGGNet slightly improved accuracy in resolving these subtle distinctions, it did so at the cost of substantially increased parameters and longer training times. LeNet-5, as a classical baseline, demonstrated the lowest parameter count but relatively slower training and lower accuracy compared to the lightweight CNNs.

Overall, this study demonstrates that careful design of compact CNN architectures can balance efficiency and accuracy for multilingual handwritten character recognition. Our findings provide a practical reference for future research on lightweight models, highlighting the trade-offs between model depth, parameter count, and the ability to capture fine-grained character details. Future work may focus on targeted data augmentation, diacritic-aware preprocessing, or hybrid architectures that further improve recognition of visually similar classes while maintaining computational efficiency.

References

[1] J. Katona and A. Pődör, “AI-supported Visual Lisp Programming in Geoinformatics,” in *AIS 2024 – 19th International Symposium on Applied Informatics and Related Areas*, 2024, pp. 71–81.

[2] O. Shvets, B. Smakanov, and Z. Bakatbayeva, “Monitoring of the State of a Person at Hazardous Work,” in *AIS 2021 – 16th International Symposium on Applied Informatics and Related Areas*, 2021, pp. 88–91.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, 1998, vol. 86, no. 11, pp. 2278–2324.

[4] E. Dar *et al.*, “Optimizing Convolutional Neural Network Hyperparameters for MNIST Image Classification: Insights and Implications,” in *2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, 2024, vol. 1, pp. 16–21.

[5] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “EMNIST: Extending MNIST to handwritten letters,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2921–2926.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[7] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[8] G. E. Bartos, Y. Hoşcan, A. Kauer, and É. N. Hajnal, “A Multilingual Handwritten Character Dataset: THE Dataset,” *Acta Polytechnica Hungarica*, vol. 17, no. 9, pp. 141–160, 2020.

[9] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: a review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[10] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, 2006.

[11] A. Lamb, A. Kitamoto, D. Ha, K. Yamamoto, M. Bober-Irizar, and T. Clanuwat, “Deep learning for classical japanese literature,” in *NIPS Workshop on Machine Learning for Creativity and Design*, 2018.

[12] H. Pandey and A. P. Agrawal, “Multi-lingual handwritten character recognition using Deep Learning.,” in *Symposium on Computing & Intelligent Systems*, 2024, pp. 19–28.

[13] S. K. Dash, S. Pranay, P. Hemanth, and A. Sekar, “Multi-Lingual Handwritten Recognition Using Convolutional Recurrent Neural Networks,” in *2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2024, pp. 1–7.

[14] A. G. Howard *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[15] M. Shafiq and Z. Gu, “Deep Residual Learning for Image Recognition: A Survey,” *Applied Sciences*, vol. 12, no. 18, 2022.

[16] A. Serdaly, D. Imanbekkyzy, A. Akhmetay, and B. Omarov, “Improving Network Intrusion Detection Using The Mini-VGGNet Architecture: Tackling Challenges of Imbalanced Data,” *Journal of Problems in Computer Science and Information Technologies*, vol. 2, no. 4, pp. 3–16, 2024.

[17] S. An, M. Lee, S. Park, H. Yang, and J. So, “An ensemble of simple convolutional neural network models for mnist digit recognition,” *arXiv preprint arXiv:2008.10400*, 2020.

[18] R. Pan and H. Rajan, “On decomposing a deep neural network into modules,” in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 889–900.

[19] S. Jeon, B. S. Ko, and S. H. Son, “ROMI: A Real-Time Optical Digit Recognition Embedded System for Monitoring Patients in Intensive Care Units,” *Sensors*, vol. 23, no. 2. 2023.

[20] S. Pundlik, A. Singh, G. Baghel, V. Baliutaviciute, and G. Luo, “A Mobile Application for Keyword Search in Real-World Scenes,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, pp. 1–10, 2019.

[21] O. Rainio, J. Teuho, and R. Klén, “Evaluation metrics and statistical tests for machine learning,” *Scientific Reports*, vol. 14, no. 1, p. 6086, 2024.