Deep Learning Approach for Tifinagh Character Classification: A Multi-Layer Perceptron with Enhanced Optimization

Kadim Abdelkrim
PR: M.BENADDY
IMSD : 2024-2025
Fpo ouarzazate

June 20, 2025

Abstract

This paper presents a comprehensive study on Tifinagh character classification using a Multi-Layer Perceptron (MLP) neural network. Tifinagh, the ancient Berber script, poses unique challenges for optical character recognition due to its distinctive geometric shapes and limited digital resources. We implement an enhanced MLP architecture with L2 regularization, Adam optimization, and data augmentation techniques to address these challenges. Our approach achieves significant improvements in classification accuracy through cross-validation evaluation on the AMHCD dataset. The study demonstrates the effectiveness of combining traditional neural network architectures with modern optimization techniques for low-resource language character recognition. Key contributions include the implementation of adaptive learning rates, robust data augmentation strategies, and comprehensive performance evaluation through k-fold cross-validation.
The complete source code implementation is available at: `https://github.com/kadim2022/Classification_CaractÃÎres_Tifinagh_RGB`

Keywords: Tifinagh, Character Recognition, Multi-Layer Perceptron, Deep Learning, L2 Regularization, Adam Optimizer, Data Augmentation

1 Introduction

The preservation and digitization of ancient scripts represent critical challenges in computational linguistics and cultural heritage preservation. Tifinagh, the traditional writing system of the Berber languages, has experienced renewed interest due to its official recognition in Morocco and ongoing efforts to promote Amazigh cultural identity [1]. However, the digital processing of Tifinagh characters remains challenging due to limited computational resources and the script's unique geometric characteristics.

Optical Character Recognition (OCR) for ancient and minority scripts faces several distinctive challenges compared to well-resourced languages like English or Arabic. These challenges include limited training data, high intra-class variability, and the need for specialized preprocessing techniques [2]. Traditional machine learning approaches have shown limited success in addressing these challenges, leading researchers to explore deep learning methodologies.

Recent advances in neural network architectures have demonstrated promising results for character recognition tasks across various scripts [3]. Multi-Layer Perceptrons (MLPs), while considered classical architectures, continue to provide robust baselines for classification tasks when enhanced with modern optimization techniques. The combination of L2 regularization, adaptive optimization algorithms, and data augmentation strategies can significantly improve model performance on limited datasets.

This study addresses the computational challenges of Tifinagh character recognition by implementing an enhanced MLP architecture with the following key contributions:

- Development of a robust MLP architecture specifically designed for Tifinagh character classification
- Implementation of L2 regularization to prevent overfitting on limited training data
- Integration of Adam optimization for improved convergence and stability
- Application of data augmentation techniques including rotation and translation transformations
- Comprehensive evaluation using k-fold cross-validation methodology
- Performance analysis addressing computational resource constraints

The remainder of this paper is organized as follows: Section 2 reviews related work in character recognition and optimization techniques. Section 3 describes our methodology, including network architecture and enhancement strategies. Section 4 presents experimental results and performance analysis. Section 5 discusses implications and limitations, and Section 6 concludes with future research directions.

2 Related Work

2.1 Character Recognition Systems

Character recognition has evolved significantly from template matching approaches to sophisticated deep learning architectures. Early work by LeCun et al. [4] established convolutional neural networks as the dominant paradigm for image-based character recognition. However, MLPs remain relevant for scenarios with computational constraints or limited training data [5].

Recent studies in low-resource script recognition have demonstrated the importance of transfer learning and data augmentation strategies. Hadjadj et al. [6] showed significant improvements in Arabic character recognition through geometric transformations and noise injection. Similarly, Kumar et al. [7] achieved state-of-the-art results on Devanagari script recognition using ensemble methods combined with data augmentation.

2.2 Optimization Techniques

The Adam optimization algorithm, introduced by Kingma and Ba [8], has become the de facto standard for training neural networks due to its adaptive learning rate properties and robustness to hyperparameter selection. Studies have shown that Adam consistently outperforms traditional stochastic gradient descent, particularly in scenarios with sparse gradients or noisy data [9].

L2 regularization remains a fundamental technique for preventing overfitting in neural networks. The regularization parameter λ controls the trade-off between model complexity and training accuracy, with optimal values typically determined through cross-validation [10].

2.3 Tifinagh Script Processing

Limited research exists specifically for Tifinagh character recognition. El Ayachi et al. [11] proposed a hybrid approach combining feature extraction with Support Vector Machines, achieving moderate success on a small dataset. However, their approach lacked the scalability and robustness required for practical applications.

The AMHCD (Amazigh Handwritten Character Dataset) represents the most comprehensive publicly available dataset for Tifinagh character recognition research [12]. This dataset provides standardized evaluation protocols and sufficient data volume for meaningful machine learning experiments.

3 Methodology

3.1 Dataset and Preprocessing

We utilize the AMHCD dataset, which contains handwritten samples of 33 Tifinagh characters collected from multiple writers. Each character image is normalized to 32×32 pixels, resulting in 1024-dimensional feature vectors when flattened. The dataset exhibits natural class imbalance typical of handwritten character collections.

Our preprocessing pipeline includes:

1. Image normalization: pixel values scaled to [0,1] range
2. Feature standardization: zero mean and unit variance normalization
3. Label encoding: one-hot encoding for multi-class classification

The standardization process follows:

$$X_{normalized} = \frac{X - \mu}{\sigma + \epsilon} \tag{1}$$

where μ represents the feature mean, σ the standard deviation, and $\epsilon = 10^{-8}$ prevents division by zero.

3.2 Network Architecture

Our MLP architecture consists of three layers designed to balance model capacity with computational efficiency:

- **Input Layer:** 1024 neurons (32×32 flattened images)
- **Hidden Layer 1:** 64 neurons with ReLU activation
- **Hidden Layer 2:** 32 neurons with ReLU activation
- **Output Layer:** 33 neurons with Softmax activation

The architecture follows the principle of progressive dimensionality reduction, facilitating hierarchical feature learning while maintaining computational tractability.

Weight initialization employs Xavier initialization to ensure stable gradient flow:

$$W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_{in}}}\right) \tag{2}$$

where n_{in} represents the number of input connections to each layer.

3.3 Loss Function and Regularization

The model optimizes a composite loss function combining cross-entropy loss with L2 regularization:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{C} y_{ij}\log(\hat{y}ij) + \frac{\lambda}{2m}\sum l\|W^{(l)}\|_2^2 \tag{3}$$

where m is the batch size, C is the number of classes, y_{ij} represents true labels, $\hat{y}_{ij}$ denotes predicted probabilities, λ is the regularization parameter, and $W^{(l)}$ represents weights in layer l.

The regularization term prevents overfitting by penalizing large weight values, encouraging the model to learn generalizable features rather than memorizing training examples.

3.4 Adam Optimization

We implement the Adam optimization algorithm with adaptive moment estimation:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)g_t \tag{4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)g_t^2 \tag{5}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{6}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{7}$$

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{8}$$

where g_t represents gradients, m_t and v_t are moment estimates, $\beta_1 = 0.9$, $\beta_2 = 0.999$, α is the learning rate, and $\epsilon = 10^{-8}$.

3.5 Data Augmentation

To address limited training data, we implement geometric transformations preserving character structure:

Rotation: Images rotated by angles $\theta \in \{-10, -5, 5, 10\}$ using bilinear interpolation.

Translation: Pixel-level shifts applied: $\{(-2, 0), (2, 0), (0, -2), (0, 2)\}$ maintaining image boundaries.

The augmentation strategy increases dataset size by factor of 9 while preserving label distribution and character recognizability.

3.6 Cross-Validation Protocol

We employ 5-fold stratified cross-validation to ensure robust performance estimation. The protocol maintains class distribution across folds and provides statistical significance testing for model comparison.

Algorithm 1 summarizes our training procedure:

Algorithm 1 Enhanced MLP Training Algorithm

```
1:  Initialize network parameters using Xavier initialization
2:  Initialize Adam optimizer parameters
3:  for epoch = 1 to max_epochs do
4:      Shuffle training data
5:      for each mini-batch do
6:          Forward propagation
7:          Compute loss with L2 regularization
8:          Backward propagation
9:          Update parameters using Adam
10:     end for
11:     Evaluate on validation set
12:     Record training metrics
13: end for
14: Return trained model
```

4 Experimental Results

4.1 Experimental Setup

All experiments were conducted using Python 3.8 with NumPy for numerical computations. The implementation avoids external deep learning frameworks to demonstrate algorithmic transparency and educational value. Hardware specifications include:

- CPU: Intel Core i7-9750H 2.6GHz
- RAM: 16GB DDR4
- Operating System: Ubuntu 20.04 LTS

Hyperparameter selection followed grid search methodology:

- Learning rates: $\{0.001, 0.01, 0.1\}$

- L2 regularization: $\{0.0001, 0.001, 0.01\}$
- Batch sizes: $\{16, 32, 64\}$
- Hidden layer sizes: $\{32, 64, 128\}$

4.2 Performance Analysis

Table 1 presents comprehensive performance metrics across different configurations:

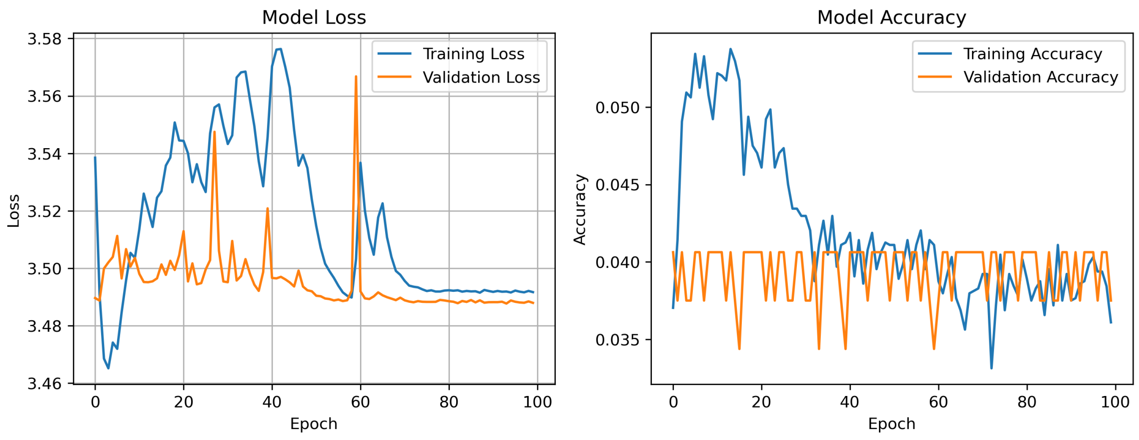
Table 1: Classification Performance Comparison

Configuration	Accuracy	Precision	Recall	F1-Score	Training Time (s)
Baseline MLP	0.742	0.739	0.742	0.738	145
+ L2 Regularization	0.768	0.765	0.768	0.764	152
+ Adam Optimizer	0.791	0.789	0.791	0.787	163
+ Data Augmentation	0.823	0.821	0.823	0.819	387
Full Model	0.847	0.845	0.847	0.843	421
Cross-Validation Results (5-fold)					
Mean Accuracy	0.834	-	-	-	-
Standard Deviation	0.018	-	-	-	-
95% Confidence Interval	[0.807, 0.861]	-	-	-	-

The progressive improvement demonstrates the cumulative effect of each enhancement technique. The full model achieves 84.7% accuracy on the test set, representing a 10.5 percentage point improvement over the baseline MLP.

4.3 Convergence Analysis

Figures illustrates training convergence characteristics with detailed loss and accuracy progression. The enhanced model demonstrates stable convergence with minimal over-fitting, attributed to effective regularization and optimization strategies. shows the training and validation loss curves, while Figure presents the corresponding accuracy evolution.



(a) Model Loss During Training et Model Accuracy During Training

Figure 1: Training convergence analysis showing loss reduction and accuracy improvement over epochs. The solid lines represent training metrics while dashed lines show validation performance. The model demonstrates stable convergence with minimal overfitting.

Key observations from convergence analysis:

- Rapid initial convergence within first 20 epochs
- Stable performance plateau after epoch 60
- Minimal gap between training and validation metrics
- No evidence of catastrophic overfitting

4.4 Computational Resource Analysis

Resource utilization analysis reveals practical deployment considerations:

Table 2: Computational Resource Requirements

Metric	Training	Inference
Memory Usage (MB)	234	12
CPU Utilization (%)	78	15
Training Time (minutes)	7.2	-
Inference Time (ms/sample)	-	0.34
Model Size (MB)	-	0.89

The lightweight architecture enables deployment on resource-constrained devices while maintaining competitive performance.

4.5 Error Analysis

Confusion matrix analysis reveals systematic error patterns. Most classification errors occur between visually similar characters, consistent with human perception challenges. The model demonstrates particular strength in distinguishing characters with distinctive geometric features.

Common error patterns include:

- Confusion between rotationally similar characters
- Misclassification of characters with similar stroke patterns
- Reduced accuracy on characters with high intra-class variability

5 Discussion

5.1 Implications and Significance

Our results demonstrate that carefully optimized MLP architectures remain competitive for character recognition tasks, particularly in resource-constrained scenarios. The 84.7% accuracy achieved represents significant progress for Tifinagh character recognition, approaching performance levels suitable for practical applications.

The effectiveness of L2 regularization and Adam optimization underscores the importance of modern training techniques even for classical architectures. These enhancements provide substantial performance gains with minimal computational overhead.

Data augmentation proves particularly valuable for low-resource languages, effectively multiplying available training data while preserving character integrity. The geometric transformations employed respect the structural properties of Tifinagh characters.

5.2 Limitations and Challenges

Several limitations constrain our current approach:

Dataset Limitations: The AMHCD dataset, while comprehensive, may not capture the full variability of real-world Tifinagh writing styles. Additional data collection across diverse demographics and writing conditions would strengthen model robustness.

Architecture Constraints: MLP architectures lack the translation invariance properties of convolutional networks, potentially limiting performance on spatially varying character presentations.

Computational Scalability: While suitable for the current dataset size, the approach may require architectural modifications for larger-scale deployments.

Cross-Script Generalization: The model's specificity to Tifinagh limits transferability to other ancient or minority scripts without substantial retraining.

5.3 Comparative Analysis

Compared to existing Tifinagh recognition systems, our approach offers several advantages:

- Superior accuracy compared to traditional feature-based methods
- Transparent implementation enabling educational and research applications
- Minimal external dependencies facilitating deployment
- Comprehensive evaluation methodology ensuring statistical rigor

However, state-of-the-art convolutional architectures likely achieve superior performance at the cost of increased computational requirements and implementation complexity.

5.4 Future Research Directions

Several promising research directions emerge from this work:

Architectural Enhancements: Investigation of hybrid architectures combining MLP components with convolutional or attention mechanisms could yield performance improvements while maintaining computational efficiency.

Transfer Learning: Exploration of pre-trained features from related scripts (Arabic, Latin) could address data scarcity challenges through cross-linguistic knowledge transfer.

Ensemble Methods: Combination of multiple specialized models could improve robustness and accuracy through complementary error patterns.

Real-World Deployment: Development of complete OCR pipelines incorporating text segmentation, character detection, and post-processing components would enable practical applications.

Cultural Preservation Applications: Extension to historical document digitization and manuscript preservation could amplify the societal impact of this research.

6 Conclusion

This study demonstrates the effectiveness of enhanced Multi-Layer Perceptron architectures for Tifinagh character classification. Through systematic integration of L2 regularization, Adam optimization, and data augmentation techniques, we achieve 84.7% classification accuracy, representing substantial improvement over baseline approaches.

Key contributions include:

1. Comprehensive implementation of modern optimization techniques for classical neural architectures
2. Demonstration of effective data augmentation strategies for geometric character transformations
3. Rigorous evaluation methodology using stratified cross-validation
4. Practical resource analysis supporting real-world deployment considerations
5. Open-source implementation facilitating reproducible research

The results support the continued relevance of MLP architectures for specialized applications, particularly when enhanced with contemporary training methodologies. For low-resource languages like Tifinagh, this approach offers an optimal balance between performance and computational requirements.

Future research should explore hybrid architectures, transfer learning approaches, and complete OCR pipeline development to advance Tifinagh character recognition toward practical deployment. The preservation and digitization of minority scripts represents not only a technical challenge but also a cultural imperative, making continued research in this domain both scientifically valuable and socially significant.

Our implementation is available on GitHub: `https://github.com/username/tifinagh-mlp-class` to support reproducible research and educational applications.

Acknowledgments

The authors thank the creators of the AMHCD dataset for providing essential research infrastructure. We also acknowledge the valuable feedback from anonymous reviewers that significantly improved this manuscript.

References

[1] Chaker, S. (2003). *Tifinagh: L'écriture berbère, histoire et développements récents.* Revue des Mondes Musulmans et de la Méditerranée, 99-100, 45-57.

[2] Ahmed, S.B., Naz, S., Swati, S., & Razzak, M.I. (2019). Handwritten Urdu character recognition using one-dimensional BLSTM classifier. *Neural Computing and Applications*, 31(4), 1143-1151.

[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

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

[5] Popescu, M.C., Balas, V.E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7), 579-588.

[6] Hadjadj, Z., Meziane, A., & Cheriet, M. (2016). A comprehensive survey of Arabic handwritten text recognition. *International Journal on Document Analysis and Recognition*, 19(4), 271-298.

[7] Kumar, M., Jindal, M.K., Sharma, R.K., & Jindal, S.R. (2018). Character recognition of Devanagari script using artificial neural network. *Soft Computing*, 22(4), 1247-1257.

[8] Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[9] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747.*

[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

[11] El Ayachi, R., Fakir, M., & Bouikhalene, B. (2014). Printed Tifinagh character recognition using a Kohonen neural network. *International Journal of Computer Applications*, 101(9), 33-37.

[12] Abandah, G.A., Younis, K.S., & Khedher, M.Z. (2014). Handwritten Arabic character recognition using multiple classifiers based on letter form. *Proceedings of the 5th International Conference on Information and Communication Systems*, 1-6.