International Journal of Computational and Experimental Science and ENgineering
(IJCESEN)

Vol. 11-No.3 (2025) pp. 5777-5789
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Capsule Embedded ResNet for Arabic Handwriting Character Recognition

Khaled Nasri [1]*, Abdelouahab Moussaoui [2]

[1]Department of Computer Science, Faculty of Sciences, University Setif 1 - Ferhat Abbas, Setif, 19000, Algeria
* **Corresponding Author Email:** nasri.khaled@univ-setif.dz - **ORCID:** 0009-0002-0217-9625

[2]Department of Computer Science, Faculty of Sciences, University Setif 1 - Ferhat Abbas, Setif, 19000, Algeria
Email: abdelouahab.moussaoui@univ-setif.dz - **ORCID:** 0000-0003-3669-1264

Article Info:

DOI: 10.22399/ijcesen.3626
Received : 16 June 2025
Accepted : 01 August 2025

Keywords

Handwriting
Arabic characters recognition
Characters recognition
Residual Network
Capsule Networks

Abstract:

This paper presents novel hybrid architectures for Arabic handwritten character recognition, integrating capsule networks with residual neural networks (ResNets) across various embedding strategies. The proposed Custom Caps-ResNet models explore low-, mid-, high-, and multilevel capsule embeddings to synergize hierarchical feature learning with spatial relationship preservation. Evaluated on four benchmark datasets, the models achieve competitive accuracy—99.64% on OIHACDB-28 and 94.14% on Dhad—while consistently reducing test loss by up to 80% on Dhad and 66% on HMBD_V1 compared to baselines. These reductions in loss indicate enhanced prediction certainty and improved feature representation. Multilevel and mid-level embeddings perform robustly across diverse script complexities, whereas high-level embeddings excel in semantic abstraction. The variation in dataset performance reveals how capsule networks mitigate challenges in cursive connections, overlaps, and positional character forms. Overall, the integration of capsule embeddings into ResNet hierarchies leads to not only strong accuracy but also significantly more confident predictions—advancing Arabic handwriting recognition toward reliable real-world deployment.

1. Introduction

Handwriting recognition systems, including digit, character, and word recognition, are widely utilized in various applications. These systems play a crucial role in banking check processing [1], office automation [2], document digitization [3], content-based document retrieval [4], signature verification [5], postal code recognition [6], and digital character identification [4]. By automating these complex tasks, handwriting recognition enhances operational efficiency, reduces human error, and improves accuracy, thereby driving advancements in technological and industrial sectors.

The recognition of Arabic handwritten characters introduces unique challenges and opportunities within the broader field of handwriting recognition. With over 400 million native speakers, Arabic is one of the most widely spoken languages globally [7], holding cultural, historical, and religious significance. Its cursive nature, contextual letterforms, and diacritical marks make character segmentation and recognition more complex compared to non-cursive scripts [8]. Additionally, Arabic has a rich historical manuscript tradition, with numerous texts remaining un-digitized or poorly preserved. These manuscripts, often written in calligraphic styles such as Kufic, Naskh, and Thuluth, present both linguistic and computational challenges in the development of robust recognition models [9]. Despite the growing interest in handwriting recognition, the field of Arabic handwritten character recognition (AHCR) faces a significant shortage of comprehensive and high-quality datasets [10]. The scarcity of well-annotated datasets poses a significant challenge to building robust deep learning models [11], as it hampers their ability to perform consistently across diverse handwriting styles and unseen data. Enhancing the robustness of Arabic OCR systems is essential not only for improving recognition accuracy but also for ensuring reliable performance in real-world applications.

To address these challenges, researchers have explored various deep learning methods tailored for Arabic script recognition. Altwaijry et al. [12]

established the Hijja dataset (children's writing), achieving 88% (Hijja) and 97% (AHCD) accuracy. Alwagdani and Jaha [13] enhanced recognition of children’s handwriting to 93% via hybrid training (Hijja + AHCD) and improved writer-group discrimination to 94% with feature fusion. For efficiency, Kamal et al. [14] introduced the lightweight Huruf model, achieving 96.93% (AHCD) and 99.35% (MadBase). Balaha et al. [15] set new benchmarks with HMB1/HMB2 architectures, attaining 100% accuracy on CMATER and AIA9K datasets. Architectural refinements include Ullah and Jamjoom [16] boosting AHCD accuracy to 96.78% using batch normalization, and Alghyaline [17] comparing VGG, ResNet, and Inception models (optimal: 98.30% on AHCD).

Complementing CNN-based methods, researchers have increasingly explored Hybrid and Integrated Approaches to leverage complementary architectural strengths. Najam and Faizullah [18] developed an integrated CNN-RNN model effective for multi-font datasets. AlShehri [19] advanced this with DeepAHR, incorporating specialized segmentation and contextual components to achieve 98.66% (AHCD) and 88.24% (Hijaa). Transfer learning has proven valuable; AlMuhaideb et al. [20] adapted MobileNet for AHR, achieving 93.59% on the Dhad dataset (Hijja-based) efficiently, demonstrating practical utility in educational tech.

Sophisticated feature extraction techniques and advanced architectures have further driven recognition improvements. Torki et al. [21] combined SIFT descriptors with SVMs for 94.28% accuracy on AIA9K. Loey et al. [22] used Stacked Autoencoders for unsupervised learning, achieving 98.5% on MADBase. Advanced models include Ghofrani et al. [23]'s Capsule Network (CapsNet) with dynamic routing, achieving state-of-the-art 99.87% on Hoda by modeling spatial relationships robustly. Al-Taani and Ahmad [24] utilized Residual Networks' deep feature learning, achieving 99.8% on MADBase via effective gradient flow. Specialized approaches address unique script features: Lutf et al. [25]'s diacritic-focused CCRP method attained 98.73% accuracy efficiently, outperforming traditional methods. Kada et al. [26] employed SVM to enhance segmentation and character extraction quality at the word level.

This is particularly vital for the digitization of historical Arabic manuscripts [27], which supports academic research, cultural preservation, and broader access to Arabic literary heritage, thereby unlocking their potential for linguistic, historical, and technological advancement.

2. Background

The Arabic script consists of 28 distinct characters (letters), written from right to left, and unlike many other writing systems, it does not employ upper-or lower-case forms. A defining feature of Arabic script is the contextual shaping of its characters, where the form of each letter varies depending on its position within a word. Specifically, most letters exhibit two primary forms: disconnected (isolated) and connected, with the latter further categorized into initial (beginning of a word), medial (middle of a word), and final (end of a word) forms (see Table. 1). This unique morphological characteristic not only contributes to the script's visual elegance and aesthetic appeal but also introduces significant challenges for automated recognition, processing, and computational analysis systems.

Table 1. *Arabic letters variations.*

N°	Letter	Letter’s Form Variations			
		Disconnected (Isolated)	Connected		
			Initial	Medial	Final
1	Alif	أ	-	-	-
2	Baa	ب	بـ	ـبـ	ـب
3	Taa	ت	تـ	ـتـ	ـت
4	Thaa	ث	ثـ	ـثـ	ـث
5	Jeem	ج	جـ	ـجـ	ـج
6	Haa	ح	حـ	ـحـ	ـح
7	Khaa	خ	خـ	ـخـ	ـخ
8	Daal	د	-	-	ـد
9	Dhal	ذ	-	-	ـذ
10	Raa	ر	-	-	ـر
11	Zaa	ز	-	-	ـز
12	Seen	س	سـ	ـسـ	ـس
13	Sheen	ش	شـ	ـشـ	ـش
14	Saad	ص	صـ	ـصـ	ـص
15	Dhad	ض	ضـ	ـضـ	ـض
16	Tta	ط	طـ	ـطـ	ـط
17	Dha	ظ	ظـ	ـظـ	ـظ
18	Ain	ع	عـ	ـعـ	ـع
19	Ghain	غ	غـ	ـغـ	ـغ
20	Faa	ف	فـ	ـفـ	ـف
21	Qaf	ق	قـ	ـقـ	ـق
22	Kaaf	ك	كـ	ـكـ	ـك
23	Laam	ل	لـ	ـلـ	ـل
24	Meem	م	مـ	ـمـ	ـم
25	Noon	ن	نـ	ـنـ	ـن
26	Haa	ه	هـ	ـهـ	ـه
27	Waaw	و	-	-	ـو
28	Yaa	ي	يـ	ـيـ	ـي

In Arabic script, the letter Alif (أ) has a simplified form known as Hamza (ء). The Hamza exhibits distinct graphical variations depending on its position within a word, such as at the middle, or end. These positional forms are integral to the orthographic rules of Arabic writing and contribute to the script's unique visual and structural characteristics Isolated: ء Middle: ـئـ End: ئ, above waw: ؤ, end above Alif: ـأ, end below Alif: ـإ ، Initial connected: ئـ.

However, datasets such as AHCD [28], ALIF [29] which exclusively consist of isolated character forms, and not include Hamza, are limited in their applicability to tasks like Arabic writing recognition. This is because they fail to account for the contextual and positional variations of characters that constitute the majority of Arabic script. By omitting these contextual forms, such datasets overlook a fundamental aspect of Arabic writing, thereby restricting their utility in real-world applications.

Arabic handwriting recognition faces challenges rooted in the script's cursive nature, letter similarity, and stylistic variability Figure 1. Characters often connect via horizontal strokes or ligatures Figure 2, complicating segmentation due to overlapping components and merged letters [30]. Ambiguity arises from letters sharing near-identical shapes (distinguished only by dots/diacritics, often omitted in handwriting) Figure 3 [31], compounded by contextual dependencies. Additionally, individual writing styles (stretched, compressed, or slanted characters) introduce variability [32], further challenging recognition systems. Researchers address these obstacles with advanced segmentation, feature extraction, and classification techniques.

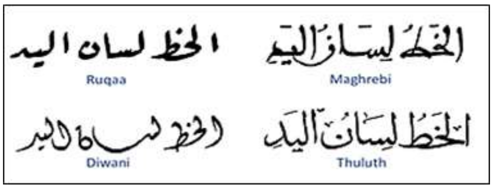


Figure 1. *Variability in Arabic Handwriting Style.*

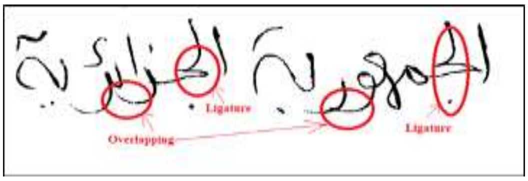


Figure 2. *Ligature and overlapping.*

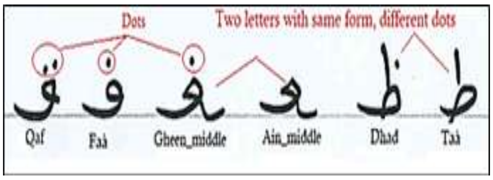


Figure 3. *Same form with different dots.*

3. Material and Methods

Recent hybrid frameworks, including Capsule Embedded ResNet [33], CapsuleNet-ResNet Fusion Model [34], and Dilated Residual Capsule Networks [35], combine ResNet's deep feature extraction with CapsNet's spatial reasoning, yet their use in Arabic script recognition remains limited, as most studies apply these techniques separately. To address this, we propose ResCapNet, a Residual Capsule Network that embeds CapsNet layers within a ResNet backbone, merging residual learning for stable training and robust feature extraction across diverse handwriting samples with embedded capsules to capture spatial hierarchies for distinguishing subtle positional variations in Arabic characters, such as س vs. ش.

3.1 Overview of the Proposed Approach

In our proposed Capsule Embedded ResNet architecture (Figure 4), capsule layers are strategically inserted after residual blocks at multiple stages of the ResNet backbone to synergistically combine ResNet's deep feature extraction with capsule networks' ability to model spatial hierarchies.

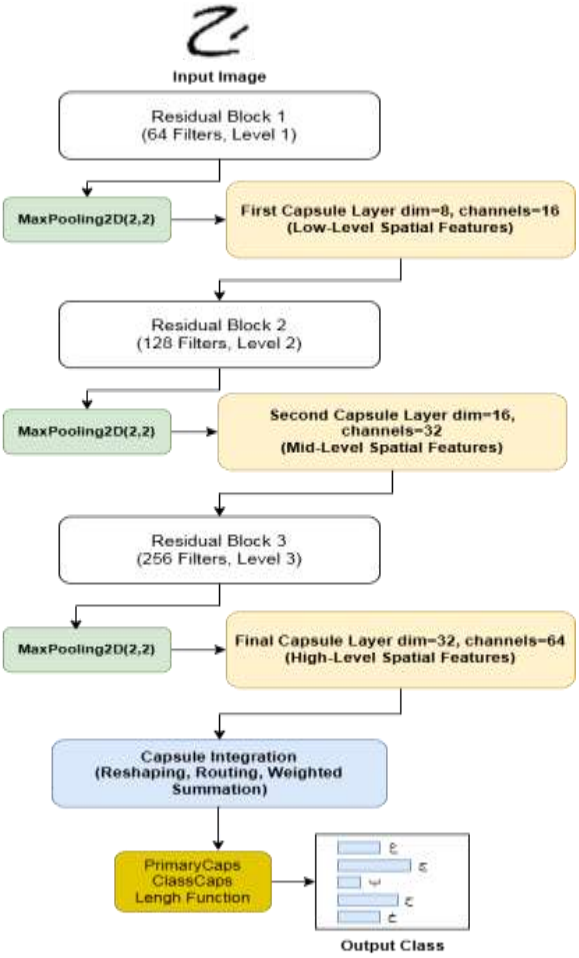

Input Image
Residual Block 1
(64 Filters, Level 1)
MaxPooling2D(2,2)
First Capsule Layer dim=8, channels=16
(Low-Level Spatial Features)
Residual Block 2
(128 Filters, Level 2)
MaxPooling2D(2,2)
Second Capsule Layer dim=16, channels=32
(Mid-Level Spatial Features)
Residual Block 3
(256 Filters, Level 3)
MaxPooling2D(2,2)
Final Capsule Layer dim=32, channels=64
(High-Level Spatial Features)
Capsule Integration
(Reshaping, Routing, Weighted Summation)
PrimaryCaps
ClassCaps
Lengh Function
Output Class

Figure 4. *Multi-level Capsule Embedding Architecture.*

3.1.1 Preservation of Spatial Hierarchies

Capsule networks, as introduced by Sabour et al. [36], use vector-based activations and dynamic routing to encode spatial relationships (e.g., stroke curvature, diacritic placement). The squash function, defined as:

$$\boldsymbol{v_j} = \frac{\left\|s_j\right\|^2}{1+\left\|s_j\right\|^2} \cdot \frac{s_j}{\left\|s_j\right\|^2} \qquad (1)$$

Where: S_j is the input vector, V_j is the output vector (squashed), and $||S_j||$ is the length (norm) of the input vector. Ensures that the magnitude of capsule outputs represents the probability of feature presence while preserving orientation. By placing capsule layers after residual blocks, we leverage ResNet’s hierarchical features to feed spatially rich inputs into capsules, enabling robust modeling of Arabic characters’ part-whole relationships (e.g., distinguishing س vs. ش).

3.1.2 Multi-Level Feature Aggregation

Capsule layers are embedded at multiple abstraction levels, with features aggregated via learned weighted summation:

$$F_{aggregated} = \sum_{i=1}^{N} w_i . F_i \qquad (2)$$

Where: F_i is the feature output from the i-th capsule layer, w_i is the learned weight via a fully connected layer, and N is the number of capsule layers (here N=3 embedding levels). This approach captures both low-level (stroke details) and high-level (character identity) features, critical for handling Arabic script’s positional variations (initial, medial, final forms). The dynamic routing algorithm [36] refines predictions by iteratively updating coupling coefficients Cij computed via a softmax function:

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k b_{ik}} \qquad (3)$$

Where: b_{ij} is represents the log-prior probabilities of capsule i contributing to capsule j.
This mechanism ensures that spatial hierarchies are preserved across abstraction levels, enhancing discriminability for ligature-dependent and diacritic-rich characters.

3.1.3 Loss Reduction and Prediction Confidence

The margin loss function defined as:

$$L_k = T_K\, max(0, m^+ - \parallel v_k \parallel)^2 + \lambda(1 - T_k)\, max(0, \parallel v_k \parallel - m^-)^2 \qquad (4)$$

Where: Lk is the loss for capsule k, Tk is 1 if class k is present and 0 otherwise, ||vk|| is the length of the output vector of capsule k, which represents the probability of an object of class k being present, m+ is the upper margin (typically 0.9), m- is the lower margin (typically 0.1), and λ is a down-weighting factor (typically 0.5) that reduces the contribution of absent classes to avoid initial learning from shrinking all capsule outputs.
Multi-level capsule placement allows reduces loss values by refining feature representations at different stages, improving prediction certainty for complex Arabic script features.

3.1.4 Dynamic Routing for Adaptive Feature Agreement

The dynamic routing algorithm, a hallmark of capsule networks, is employed within capsule layers to create adaptive connections between lower-mid and higher-level capsules [36].

Procedure ROUTING ($\hat{u}_{j|i}, r, l$)

```
1. for all capsule i in layer l and capsule j (l+1): $b_{ij} \leftarrow 0$
2. for r iteration do
3.   for all capsule i in layer l: $c_i \leftarrow softmax(b_i)$
4.   for all capsule j in layer (l+1): $s_j \leftarrow \sum_i c_{ij}\, \hat{u}_{j|i}$
5.   for all capsule j in layer (l+1): $v_j \leftarrow squash(sj)$
6.   for all capsule i in layer l and capsule j (l+1): $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} . v_j$
7. return $v_j$
```

For each routing iteration (typically 3), coupling coefficients are computed via a softmax function, weighting predictions based on agreement (e.g., spatial consistency of strokes or diacritics). This process ensures that capsules representing Arabic character components (e.g., loops, dots) align with higher-level capsules representing whole characters, enhancing robustness to handwriting variability and improving classification accuracy.

3.2 Experimental Protocol

This section details our experimental methodology for evaluating the proposed Capsule Embedded ResNet architecture for Arabic handwritten character recognition. Our experiments aim to assess both the performance and robustness of our model through comprehensive cross-dataset validation using samples from diverse writer demographics and various Arabic character formations.

3.2.1 Used Datasets

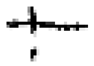
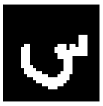
We conduct four primary experiments across four complementary Arabic handwriting datasets with total of 16 experiments. Summarized in Table 2, used datasets represent diverse writer demographics (children and adults) and contain various Arabic character configurations (isolated, initial, medial, and final positions) with different class distributions, and sizes.

All datasets contain Arabic characters with strokes and diacritics intact Table 3. These features are critical for distinguishing between similar letters (e.g., س vs. ش or ب vs. ت vs. ث) and are preserved without simplification or removal during preprocessing. This ensures the models learn to recognize the script's morphological and contextual nuances, which are essential for effective Arabic handwriting recognition.

Table 2. *Characteristics of the benchmark datasets used.*

Dataset	Size	Main Characteristics	Train 80%	Val 10%	Test 10%
Dhad [26]	55587	Child writers (ages 6–14) variations: (initial/medial/final forms) 29 classes (based on Hijja)	44471	5558	5558
AHCD [13]	16800	Adult, only isolated 28 classes	13440	1678	1678
OIHACDB 28 [40]	5600	Adult, only isolated 28 classes	4 480	560	560
HMBD V1 [21]	54110	Adult, with all letter's variations, 115 classes, include digits	43288	5411	5411

Table 3. *Samples of characters from used datasets.*

 HMBD_V1 (Adulte)	 Dhad (Children)	 Baã middle
 HMBD_V1 (Adulte)	 Dhad (Children)	 Gheen middle
 AHCD (Adulte)	 OIHACDB (Adulte)	 Seen isolated

3.2.2 Conducted Experiments

- **Caps-ResNet18**: Evaluates the Caps-ResNet18 architecture, where capsule layers are embedded at the end of a ResNet18 backbone. We implement the ResNet18 architecture trained from scratch. This experiment is systematically applied to all four datasets to assess how the integration of capsule networks with residual networks performs across different writing styles and character complexities.
- **Standalone ResNet18:** This experiment evaluates standalone ResNet18 architecture without any capsule layers. We implement the ResNet18 model trained from scratch, serving as a baseline to assess the performance of hierarchical feature extraction alone. This configuration isolates the contributions of the ResNet backbone in recognizing Arabic handwritten characters, without the influence of capsule networks.
- **Caps-ResNet:** Investigates a custom-built residual network constructed from scratch, comprising three specialized residual blocks as detailed in the previous section. This experiment aims to determine whether a purpose-built architecture with fewer residual blocks can achieve comparable or superior performance to ResNet18 model.
- **ResNet with Single Capsule Unit:** This experiment investigates a custom ResNet architecture enhanced with a single capsule layer embedded at multiple levels: low, mid, and high. The configuration is designed to explore minimal yet effective integration of capsule networks into ResNet, with the capsule layer responsible for modeling spatial hierarchies at the respective levels of abstraction.

3.3 Pipeline Overview

This section presents a structured framework for handwritten script recognition, encompassing three core stages: data preprocessing, model training, and systematic evaluation (Figure 5). Each phase is designed to ensure robustness, reproducibility, and performance across diverse datasets. The subsequent subsections detail the methodology and rationale behind these critical components.

3.3.1 Data Preprocessing

Data preprocessing transforms raw, unstructured data into a refined format suitable for training machine and deep learning models. It addresses challenges like variability in input quality, class imbalance, and limited dataset diversity through normalization, augmentation, and balancing techniques. This foundational step ensures models generalize effectively and achieve reliable performance on real-world tasks.

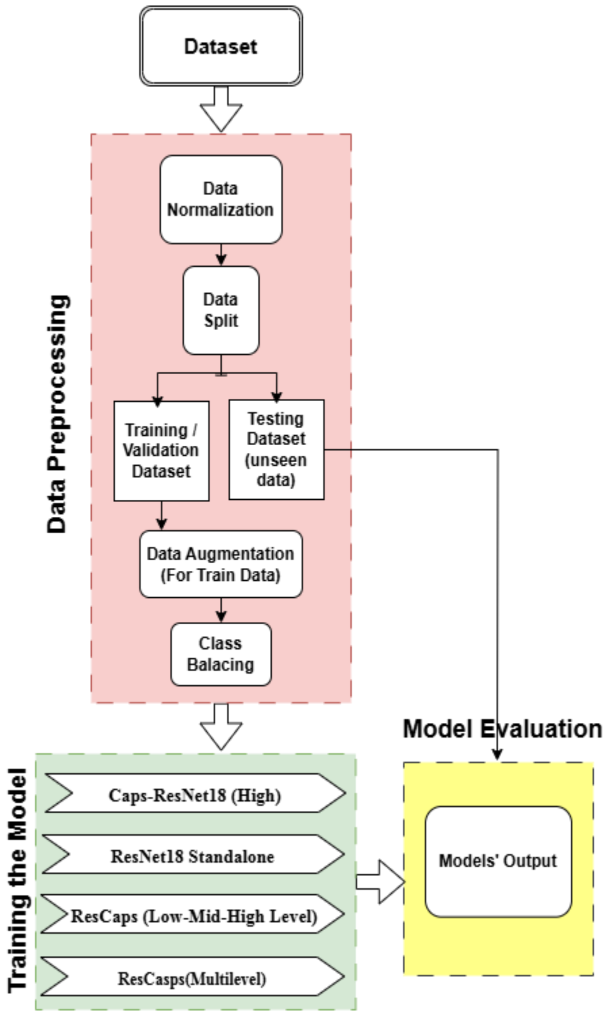


Figure 5. *Proposed Pipeline.*

a) Min-Max Normalization: Grayscale images (0–255) are normalized to a 0–1 range to reduce intra-class variability caused by scanning artifacts or ink density differences. b) Dataset Splitting: The dataset is divided into 80% training (model learning), 10% validation (hyper parameter tuning and overfitting prevention), and 10% testing. c) Data Augmentation: To improve generalization and robustness against real-world variations such as stylized, distorted, or incomplete handwriting, we applied a comprehensive augmentation strategy to the training set. This strategy encompassed handwriting variation simulation and appearance robustness techniques. For handwriting variation simulation, we used geometric transformations, including rotations of up to ±12°, zooming by ±20%, and shearing by ±0.2 radians, along with elastic deformations (σ=5, α=50) to mimic natural handwriting distortions. To ensure appearance robustness, we adjusted contrast by ±30% to handle varying lighting conditions and injected noise, including Gaussian noise (σ=0.05) and salt-and-pepper noise (density=0.03), to emulate capture artifacts. d) Class Balancing: Addressed imbalance using SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic minority-class samples.

These steps collectively improve model robustness, accuracy, and fairness in recognizing diverse handwriting styles.

3.3.2 Training Models

Models training phase employs a systematic exploration of hyperparameters to optimize performance across multiple experimental setups. Key parameters under investigation include batch size (balancing memory efficiency and gradient stability), routing iterations (refining capsule feature dynamics), epochs (controlling training duration and convergence), and optimizer selection (Adam, AdaMax). These experiments aim to identify configurations that maximize recognition accuracy while minimizing computational overhead. Detailed descriptions of used parameters are summarized in the Table 4.

Table 4. *Summarize of hyperparameters used in training*

Hyper Parameter	Value/Range
Batch Size	32; 64;128
Routing Iterations	3; 5
Epochs	50; 75;100
Optimizer	Adam,AdaMax
Weight Decay	1e-4; 1e-3
Loss Function	Margin Loss, categorical_crossentropy

3.3.3 Baseline and Model Evaluation

The model's performance is rigorously evaluated on a comprising unseen data not exposed during training or validation. This ensures an unbiased assessment of its capacity to generalize to new handwriting styles and real-world scenarios. Key metrics include:

- **Accuracy**: Measure the proportion of correct predictions (both true positives and true negatives) across all classes:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (5)$$

- **Precision**: Measures the accuracy of positive predictions:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (6)$$

- **Recall**: Measures the ability to identify all relevant instances:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

- **F1-score**: Is the harmonic mean of Precision and Recall, balancing both metrics:

$$\text{F}_1 \text{ score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (8)$$

Where: TP is the True positive, FP is the False positive, TN is the True negative, and FN is the False negative.

Table 5 presents the state-of-the-art accuracy benchmarks achieved by classic approaches on the AHCD, HMBD_V1, Dhad, and OIHACDB-28 datasets, establishing the performance standards against which our proposed Caps-ResNet hybrid architectures will be evaluated.

Table 5. *Baseline performances across used benchmarks*

Authors	Model	Dataset	Accuracy
Boufenar et al. [37]	AlexNet + TL	AHCD	99.98%
Boufenar et al. [38]	Custom-designed CNN	OIHACDB-28	97.32%
Balaha et al. [39]	DL-Genetic Algorithm	HMBD_V1	91.96%
AlMuhaideb et al. [20]	MobilNet	Dhad	93.59%

3.3.4 Computational Setup and Efficiency Analysis

All experiments were conducted on Google Colab Pro leveraging the NVIDIA Tesla A100 GPU, which offers significant acceleration through its high memory bandwidth (40 GB VRAM), allowing efficient handling of large batches and rapid computation during model training.

Table 6 summarizes the number of trainable parameters and average time per epoch across the evaluated architectures

Table 6. *Baseline performances across used benchmarks*

Model	Number of Parameters	Time per Epoch
Custom ResNet Low-level	61,134,272	36s
Custom ResNet Mid-level	63,094,464	46s
Custom ResNet High-level	238,796,992	73s
Custom ResNet Multilevel	484,622,016	96s
ResNet18 High-level	501,927,088	95s
ResNet18 Standalone	326,365,785	85s

The Low-level and Mid-level Custom ResNet models involve relatively fewer parameters (~61–63 million), making them computationally efficient while still achieving strong performance on simpler recognition tasks. In contrast, the High-level and multilevel embedding models particularly the Custom ResNet Multilevel (484 million parameters) and ResNet18 High-level (502 million parameters), introduce substantially greater model complexity. This increase corresponds with improved capacity to capture nuanced spatial hierarchies in Arabic script, enhancing both recognition accuracy and prediction confidence, though at the expense of higher computational cost. The Standalone ResNet18 model, with approximately 326 million parameters, offers a middle ground in terms of complexity; however, its elevated test loss across datasets indicates reduced prediction certainty and highlights the added value of capsule-augmented architectures in achieving more robust feature representation.

4. Results and Discussions

Our proposed architectures, Caps-ResNet18 and Caps-ResNet with multi-level embedding, were rigorously evaluated on four Arabic handwriting datasets:

Dhad, AHCD, OIHACDB-28, and HMBD_V1. Table 7 provides a comprehensive comparison of our models' performance.

Table 7. *Empirical results across experimental setups.*

Model	**Metrics**	**Dataset**			
		Dhad	**AHCD**	**HMBD V1**	**OIHACD 28**
Custom ResNet Low-level	Loss	0.047	0.0565	0.1244	0.1838
	Acc	90. 77%	98.57%	90.80%	98.75%
	M-F1	90.14%	98.57%	90.77%	98.74%
	W-F1	90.73%	98.57%	90.83%	98.74%
Custom ResNet Mid-level	Loss	0.0732	0.0557	0.0945	0.0505
	Acc	93.90%	**98.48%**	**91.39%**	99.64%
	M-F1	93.30%	**98.48%**	**91.34%**	99.64%
	W-F1	93.88%	**98.48%**	**91.40%**	99.64%
Custom ResNet High-level	Loss	0.0497	0.0167	0.078	0.0152
	Acc	93.67%	98.15%	89.96%	98.57%
	M-F1	93.00%	98.15%	89.95%	98.56%
	W-F1	94.00%	98.15%	89.99%	98.56%
Custom ResNet Multi-level	Loss	0.0549	0.0364	0.0843	0.013
	Acc	93.24%	98.42%	89.98%	98.93%
	M-F1	92.61%	98.42%	89.95%	98.93%
	W-F1	93.22%	98.42%	90.01%	98.93%
ResNet18 High-level	Loss	**0.0467**	**0.0169**	**0.2048**	**0.0285**
	Acc	**94.14%**	98.12%	88.91%	99.46%
	M-F1	**93.50%**	98.00%	88.61%	99.46%
	W-F1	**94.10%**	98.00%	88.69%	99.46%
Standalone ResNet18	Loss	0.283	0.137	0.294	0.298
	Acc	93.24%	97.11%	87.60%	**99.64%**
	M-F1	92.60%	97.12%	87.17%	**99.64%**
	W-F1	93.26%	97.11%	87.26%	**99.64%**

4.1 Capsule Embedding Effect on Custom ResNet models

Capsule units, integrated into the Custom ResNet models, enhance feature representation by capturing spatial hierarchies and part-whole relationships through vector-based embeddings, unlike traditional convolutional layers that rely on scalar activations. This dynamic routing mechanism allows capsules to model complex patterns, improving robustness to variations in data. Capsule embeddings are applied at different feature extraction levels (Low-level, Mid-level, High-level) and combined in the hybrid Caps-ResNet (Multilevel), enabling a nuanced analysis of their impact.

4.1.1 Low-level Capsule Embedding

The Custom ResNet Low-level model, which injects capsule embeddings in the earliest convolutional layers to capture fine-grained spatial cues (e.g., diacritics, dots), delivers competitive—but dataset-dependent—performance. On Dhad it attains a Test Accuracy of 90.8 % with a Test Loss of 0.0470, indicating strong detail extraction yet limited robustness for complex character shapes. For OIHACDB, accuracy rises to 98.8 % but the higher Test Loss of 0.1838 signals less efficient optimisation. On the more structured AHCD dataset, the model achieves 98.6 % accuracy with a moderate loss of 0.0565, whereas on HMBD-V1, accuracy drops to 90.8 % and loss rises to 0.1244, underscoring that fine-grained features alone cannot fully capture the contextual variability of this set. Overall, low-level capsule embeddings excel at preserving local patterns but exhibit limited generalisation when higher-level semantics dominate.

4.1.2 Mid-level Capsule Embedding

The Custom ResNet Mid-level model, applying Capsule embeddings at intermediate layers to capture abstract features (e.g., Hamza above), demonstrates strong and consistent performance by balancing fine detail and abstraction. On Dhad, it achieves a Test Accuracy of 93.9 % with a Test Loss of 0.0732, showing improved classification due to enhanced feature hierarchies. On OIHACDB-28, it attains an outstanding Test Accuracy of 99.64 % and a low-Test Loss of 0.0505, reflecting excellent generalization and optimization. On AHCD, the model reaches 98.48 % accuracy with a Test Loss of 0.0557, and on HMBD-V1, it records 91.39 % accuracy with a Test Loss of 0.0945, outperforming other variants. These results highlight the effectiveness of mid-level Capsule embeddings in capturing intermediate semantic features, making this architecture particularly robust across d

atasets with varied structural complexity.

4.1.3 High-Level Embedding

The Custom ResNet High-level model, incorporating Capsule embeddings at deeper layers to capture semantic features (e.g., base forms of Arabic letters), demonstrates strong performance on datasets with clear class boundaries. On Dhad, it achieves a Test Accuracy of 93.67 % and a Test Loss of 0.0497, showing effective semantic-level abstraction for character classification. It performs exceptionally well on OIHACDB-28 and AHCD, reaching 98.57 % and 98.15 % accuracy, respectively, with very low-test losses (0.0152 and 0.0167), indicating efficient optimization and robustness for structured datasets. However, performance drops on HMBD-V1, with a Test Accuracy of 89.96 % and a Test Loss of 0.0780, suggesting that semantic features alone are insufficient for datasets requiring fine-grained or multi-scale discrimination. These findings highlight the strength of high-level Capsule embeddings in capturing semantic relationships, while also pointing to their limitations in scenarios that demand more localized or hierarchical feature modeling.

4.1.4 Multilevel Embedding

The Custom ResNet Multilevel model, or hybrid Caps-ResNet, integrates Capsule embeddings across low-, mid-, and high-level layers, yielding consistent and robust performance across all datasets. On Dhad, it achieves a Test Accuracy of 93.24 % with a Test Loss of 0.0549, demonstrating balanced feature extraction and effective optimization for Arabic character recognition. On OIHACDB-28, the model excels with 98.93 % accuracy and the lowest Test Loss of 0.0130 across all configurations, highlighting its superior generalization and optimization for structured data. For AHCD, it achieves 98.42 % accuracy with a Test Loss of 0.0364, maintaining strong performance on clean, well-structured input. On HMBD-V1, it records a Test Accuracy of 89.98 % and a Test Loss of 0.0843, showing resilience to complex or ambiguous patterns, though slightly trailing the Mid-level variant in accuracy. The key strength of this hybrid Caps-ResNet lies in its ability to combine fine-grained, abstract, and semantic features through multi-scale Capsule integration, enabling improved robustness and generalization across datasets of varying complexity. Figure 6 and Figure 7 illustrates the comparative test loss and accuracy achieved by each of the proposed Custom ResNet architectures

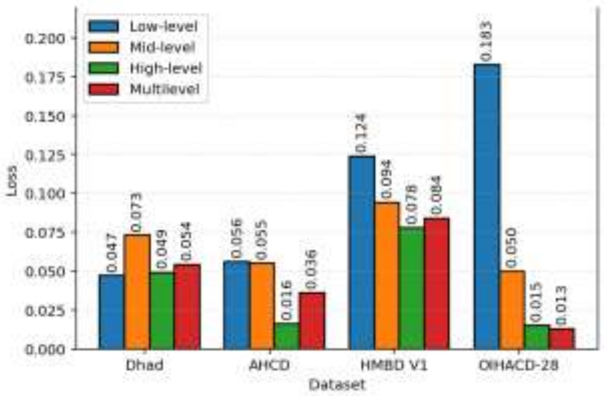


Figure 6. *Comparative Test Loss (Custom CapsNet).*

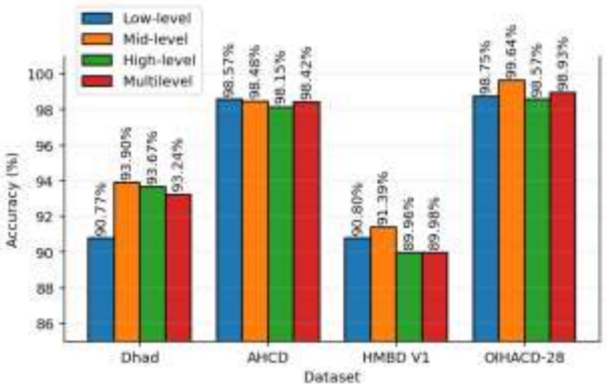


Figure 7. *Comparative Test Accuracy (Custom CapsNet).*

4.2 Capsule Embedding Effect on ResNet18 models

The experiments evaluate the impact of Capsule units in the ResNet18 High-level model, which applies Capsules at deeper layers to capture semantic features, compared to the ResNet18 Standalone model using standard convolutions, across evaluated datasets

.

4.2.1 ResNet18 High-level Capsule Embedding

The ResNet18 High-level model, incorporating Capsule embeddings at deeper layers to capture semantic features, demonstrates strong performance across datasets with distinct class boundaries. On Dhad, it achieves a Test Accuracy of 94.14 % and a Test Loss of 0.0467, reflecting effective semantic-level abstraction for Arabic character recognition. On OIHACDB-28, it reaches 99.46 % accuracy with a low-Test Loss of 0.0285, confirming efficient optimization and strong performance for structured scripts. For AHCD, the model yields 98.12 % accuracy with a Test Loss of 0.0169, showcasing its reliability on well-structured datasets. However, on HMBD_V1, it records a lower accuracy of 88.91 % and a significantly higher Test Loss of 0.2048,

indicating limitations in handling complex, variable patterns where semantic features alone are insufficient. These results underline the model's strength in capturing high-level relationships but also reveal that deeper semantic embeddings,

in isolation, may not generalize as well as multi-scale Capsule-based approaches.

4.2.2 ResNet18 Standalone

The ResNet18 Standalone model, based on the standard ResNet18 architecture without Capsule embeddings, demonstrates consistent but generally lower performance compared to Capsule-integrated variants across most datasets. On Dhad, it achieves a Test Accuracy of 93.24 % with a notably high-Test Loss of 0.283, suggesting less precise classification despite reasonable accuracy. For OIHACDB-28, it ties for the highest accuracy at 99.64 %, but with a significantly elevated Test Loss of 0.298, pointing to

suboptimal generalization. On AHCD, the model reaches 97.11 % accuracy and a Test Loss of 0.137, reflecting moderate performance for structured inputs. On HMBD_V1, it records the lowest performance among models, with a Test Accuracy of 87.60 % and a high-Test Loss of 0.294, indicating substantial challenges in handling variability and complexity. These outcomes highlight that while the ResNet18 Standalone model performs adequately on datasets with clear feature hierarchies, its high-test losses and reduced robustness on complex data emphasize the value of Capsule embeddings in enhancing feature representation and model generalization.

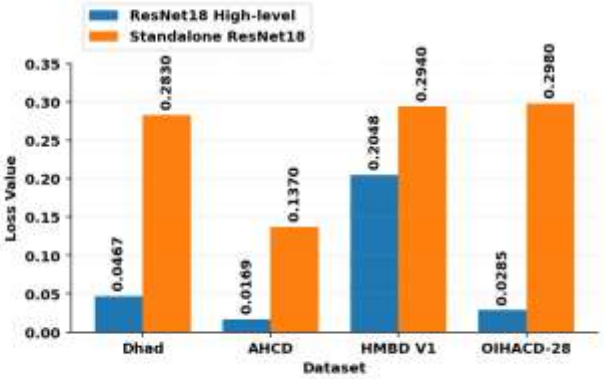


Figure 8. *Comparative Test Loss (Caps-ResNet18)*

Figure 8 and 9 presents the test loss and accuracy of the Caps-ResNet18 across the evaluated datasets.

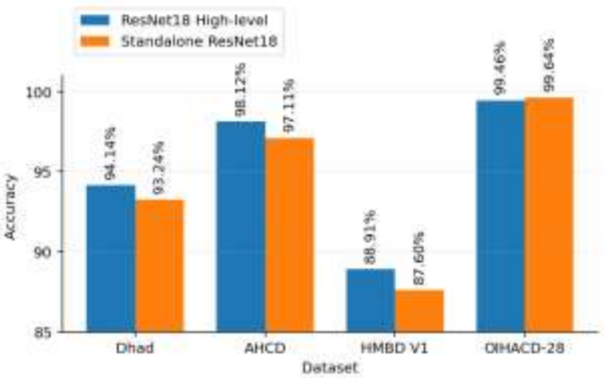


Figure 9. *Comparative Test Accuracy (Caps-ResNet18)*

4.3 Performance Comparison of Models Across Datasets with Baseline

This section presents a comparative analysis of the proposed models against baseline methods across multiple evaluated datasets, focusing on key performance metrics such as test loss and test accuracy to highlight improvements and robustness. Table 8 summarizes the key comparisons.

Table 8. *Comparative performance analysis of our proposed models against established baselines.*

Dataset	Model	Accuracy	Test Loss	Macro F1	Weighted F1
Dhad	Baseline	93.59%	0.2468	94.00%	94.00%
	ResNet18 High-level	94.14%	0.0467	93.50%	94.10%
	Custom ResNet Mid-level	93.90%	0.0732	93.30%	93.88%
AHCD	Baseline	99.98%	-	-	-
	ResNet18 High-level	98.12%	0.0169	98.00%	98.00%
	Custom ResNet Low-level	98.57%	0.0565	98.57%	98.57%
HMBD V1	Baseline	91.96%	0.23	-	-
	ResNet18 High-level	88.91%	0.2048	98.48%	98.48%
	Custom ResNet Mid-level	91.39%	0.0945	91.34%	91.40%
OIHAC DB-28	Baseline	97.32%	-	-	-
	ResNet18 Standalone	99.64%	0.298	99.64%	99.64%
	Custom ResNet Mid-level	99.64%	0.0505	99.64%	99.64%

4.3.1 Dhad Dataset

On the Dhad dataset, the baseline model achieved a Test Accuracy of 93.59% with an estimated Test Loss of 0.2468. Capsule-integrated models, particularly those using mid- and high-level embeddings, outperformed the baseline in both accuracy and optimization. The ResNet18 High-level model led with the highest accuracy (94.14%) and the lowest loss (0.0467), indicating strong robustness to character complexity. The Custom ResNet Mid-level model followed closely (93.90%, loss: 0.0732), while the Custom High-level variant also performed well (93.67%, loss: 0.0497). The Custom Multilevel and ResNet18 Standalone models matched in accuracy (93.24%), but the former achieved better optimization (loss: 0.0549) than the latter (0.2830). The Custom Low-level model recorded the lowest accuracy (90.77%) despite a low loss (0.0470), suggesting limited generalization from fine-grained features alone. These results highlight that Capsule integration at deeper or mid-levels significantly enhances performance on challenging datasets like Dhad.

4.3.2 AHCD Dataset

On the AHCD dataset, the baseline model achieved a near-perfect Test Accuracy of 99.98%, outperforming all other models in accuracy. Despite this, several Capsule-integrated models demonstrated competitive performance and significantly lower Test Losses, indicating better optimization. The Custom ResNet High-level and ResNet18 High-level models achieved accuracies of 98.15% and 98.12% with the lowest losses of 0.0167 and 0.0169, respectively, suggesting effective semantic abstraction. The Custom Multilevel model offered the best balance between accuracy (98.42%) and Test Loss (0.0364), while the Low-level and Mid-level models maintained solid accuracy (~98.5%) with moderate losses (~0.056). The ResNet18 Standalone model lagged with the lowest accuracy (97.11%) and highest loss (0.1370). These findings indicate that while the baseline excels on AHCD’s structured data, Capsule-based architectures optimize better, offering enhanced generalization and robustness.

4.3.3 HMBD_V1 Dataset

On the HMBD_V1 dataset with 115 classes, all Capsule-based and ResNet18 models underperformed the baseline accuracy of 91.96%. Among them, the Custom ResNet Mid-level model performed best, with a Test Accuracy of 91.39% and a moderate Test Loss of 0.0945, indicating reasonable robustness to complex patterns. Models with high-level or multilevel Capsule embeddings showed slightly lower accuracies (~89.9%) but had better optimization (lower losses). The ResNet18 Standalone model had the lowest accuracy (87.60%) and highest loss (0.2940), reflecting poor generalization. Overall, while none of the models surpassed the baseline, the Mid-level Capsule model stood out as the most balanced in handling HMBD_V1’s fine-grained discrimination challenges.

4.3.4 OIHACDB-28 Dataset

On the OIHACDB-28 dataset, all models outperformed the baseline accuracy of 97.32%, underscoring the effectiveness of Capsule-enhanced architectures. The Custom ResNet Mid-level and ResNet18 Standalone models tied for the highest accuracy (99.64%), though the Standalone model exhibited a much higher loss (0.2980) compared to the Mid-level's more optimized performance (0.0505). The Custom ResNet Multilevel model achieved the lowest Test Loss (0.0130) with a strong accuracy of 98.93%, reflecting excellent generalization. Similarly, the Custom ResNet High-level model recorded a low loss (0.0152) with 98.57% accuracy. The ResNet18 High-level model reached 99.46% with a loss of 0.0285, while the Custom Low-level model performed well (98.75%, loss: 0.1838). These results highlight that Capsule integration, particularly at multilevel and high-level layers, enables precise and efficient classification on structured datasets like OIHACDB-28.

Figures 10 and 11 present performance comparisons between the two proposed models (Caps-ResNet18 and Custom CapsNet) and the baseline, highlighting the best achieved results.

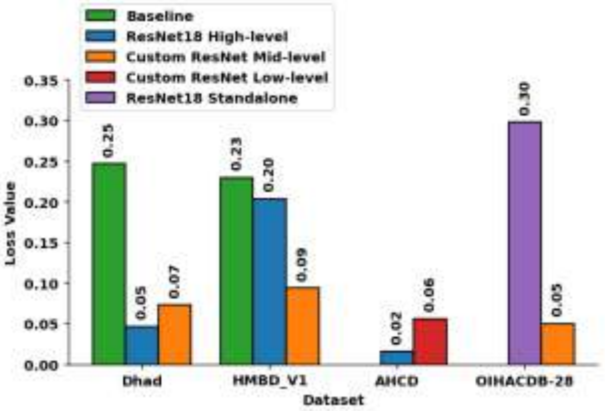


__Figure 10.__ Test Loss Comparison with Baseline.

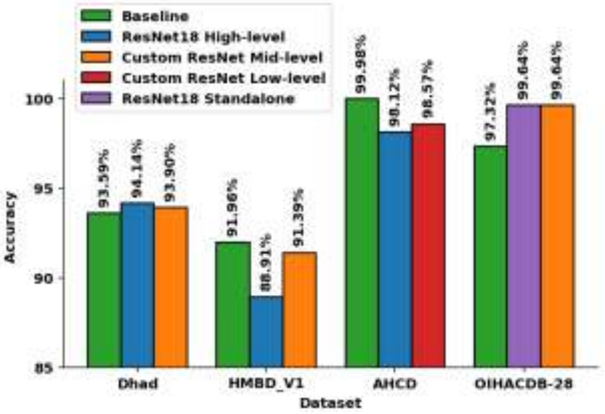


__Figure 11.__ Test Accuracy Comparison with Baseline

5. Conclusions

This study introduced novel Custom ResNet architectures augmented with capsule embeddings at distinct hierarchical levels—low, mid, high, and multilevel—to improve the recognition of Arabic handwritten characters. Evaluations were conducted across four benchmark datasets: AHCD, IOHACDB-28, Dhad, and HMBD_V1, each representing varying structural complexities and character morphologies.

The performance variation across embedding strategies reflects how different feature abstraction levels influence recognition capability. Low-level embeddings excel at capturing fine-grained spatial features like diacritics and localized strokes, which benefited cleaner, isolated-character datasets such as AHCD. However, they proved less effective for complex or context-rich datasets like Dhad and HMBD_V1, where broader structural and semantic cues are essential. Mid-level embeddings demonstrated consistent robustnessby balancing local and abstract features, achieving topperformance on moderate-complexity datasets like OIHACDB-28. The High-level embedding variant performed strongly on datasets with clear semantic boundaries but underperformed where fine morphological distinctions were critical.

The Multilevel Caps-ResNet, combining all three embedding levels, emerged as the most balanced architecture, achieving stable and strong results across all datasets. It delivered the best test loss on OIHACDB-28 (0.0130) and robust accuracy on challenging datasets like HMBD_V1 (89.98%), indicating its capacity to generalize across various handwriting styles and complexities.

In summary, the integration of capsule embeddings at strategic abstraction layers significantly enhances the spatial hierarchy modeling of ResNet architectures. These findings confirm that such hybrid models not only boost classification accuracy but also improve prediction certainty and feature robustness. Consequently, these advancements bring Arabic handwriting recognition closer to reliable deployment in real-world applications involving diverse and challenging handwritten input.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Impedovo, S., Wang, P. S. P., & Bunke, H. (1997). Automatic bankcheck processing (Vol. 28). World Scientific.

[2] Grabowski, H. (2023). Intelligent character recognition of handwritten forms with deep neural networks. In International TRIZ Future Conference (pp. 3–15). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-44768-61

[3] Louloudis, G., Gatos, B., Pratikakis, I., & Halatsis, C. (2008). Text line detection in handwritten documents. Pattern Recognition, 41(12), 3758–3772. https://doi.org/10.1016/j.patcog.2008.05.011

[4] Doermann, D. (1998). The indexing and retrieval of document images: A survey. Computer Vision and Image Understanding, 70(3), 287–298. https://doi.org/10.1006/cviu.1998.0695

[5] Kaur, H., & Kumar, M. (2023). Signature identification and verification techniques: State-of-the-art work. Journal of Ambient Intelligence and Humanized Computing, 14(2), 1027–1045. https://doi.org/10.1007/s12652-021-03503-7

[6] Filatov, A., Gitis, A., Kil, I., & Nikandrov, A. (1998). The address script recognition system for handwritten envelopes. In International Workshop on Document Analysis Systems (pp. 767–774). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-69709-8_77

[7] Eberhard, D. M., Simons, G. F., & Fennig, C. D. (Eds.). (2024). Ethnologue: Languages of the world (27th ed.). SIL International. https://www.ethnologue.com/

[8] Lorigo, L. M., & Govindaraju, V. (2006). Offline Arabic handwriting recognition: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(5), 712–724. https://doi.org/10.1109/TPAMI.2006.102

[9] Miloud, K., Benkhedda, Y., & Barkat, A. (2024). Restoration of ancient Arabic manuscripts: A deep learning approach. Studies in Engineering and Exact Sciences, 5(2), e7722. https://doi.org/10.54021/seesv5n2-033

[10] Alothman, A., Al-Turaiki, I., AlGhamdi, A. D., AlKhulaiwi, D., AlHassan, R., & AlOmran, H. (2025). A survey on the online Arabic handwriting recognition: Challenges, datasets, and future directions. In Seventeenth International Conference on Machine Vision (ICMV 2024) (Vol. 13517, pp. 1–12). SPIE. https://doi.org/10.1117/12.3057143

[11] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2023). A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications. Journal of Big Data, 10(1), 46. https://doi.org/10.1186/s40537-023-00727-2

[12] Altwaijry, N., & Al-Turaiki, I. (2021). Arabic handwriting recognition system using convolutional neural network. Neural Computing and Applications, 33(7), 2249–2261. https://doi.org/10.1007/s00521-020-05070-8

[13] Alwagdani, M. S., & Jaha, E. S. (2023). Deep learning-based child handwritten Arabic character recognition and handwriting discrimination. Sensors, 23(15), 6774. https://doi.org/10.3390/s23156774

[14] Kamal, M., Shaiara, F., Abdullah, C. M., Ahmed, S., Ahmed, T., & Kabir, M. H. (2022). Huruf: An application for Arabic handwritten character recognition using deep learning. In 2022 25th International Conference on Computer and Information Technology (ICCIT) (pp. 1131–1136). IEEE. https://doi.org/10.1109/ICCIT57492.2022.10055758

[15] Balaha, H. M., Ali, H. A., Saraya, M., & Badawy, M. (2021). A new Arabic handwritten character recognition deep learning system (AHCR-DLS). Neural Computing and Applications, 33(11), 6325–6367. https://doi.org/10.1007/s00521-020-05397-2

[16] Ullah, Z., & Jamjoom, M. (2022). An intelligent approach for Arabic handwritten letter recognition using convolutional neural network. PeerJ Computer Science, 8, e995. https://doi.org/10.7717/peerj-cs.995

[17] Alghyaline, S. (2024). Optimised CNN architectures for handwritten Arabic character recognition. Computers, Materials & Continua, 79(3), 4907–4924. https://doi.org/10.32604/cmc.2024.048734

[18] Najam, R., & Faizullah, S. (2023). Analysis of recent deep learning techniques for Arabic handwritten-text OCR and post-OCR correction. Applied Sciences, 13(13), 7568. https://doi.org/10.3390/app13137568

[19] AlShehri, H. (2024). DeepAHR: A deep neural network approach for recognizing Arabic handwritten recognition. Neural Computing and Applications, 36(21), 12103–12115. https://doi.org/10.1007/s00521-024-09674-2

[20] AlMuhaideb, S., Altwaijry, N., AlGhamdy, A. D., AlKhulaiwi, D., AlHassan, R., AlOmran, H., & AlSalem, A. M. (2024). Dhad—A children's handwritten Arabic characters dataset for automated recognition. Applied Sciences, 14(6), 2332. https://doi.org/10.3390/app14062332

[21] Torki, M., Hussein, M. E., Elsallamy, A., Fayyaz, M., & Yaser, S. (2014). Window-based descriptors for Arabic handwritten alphabet recognition: A comparative study on a novel dataset. arXiv preprint arXiv:1411.3519. https://arxiv.org/abs/1411.3519

[22] Loey, M., El-Sawy, A., & El-Bakry, H. (2017). Deep learning autoencoder approach for handwritten Arabic digits recognition. arXiv preprint arXiv:1706.06720. https://arxiv.org/abs/1706.06720

[23] Ghofrani, A., & Toroghi, R. M. (2019). Capsule-based Persian/Arabic robust handwritten digit recognition using EM routing. In 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA) (pp. 168–172). IEEE. https://doi.org/10.1109/PRIA.2019.8785981

[24] Al-Taani, A. T., & Ahmad, S. (2021). Recognition of Arabic handwritten characters using residual neural networks. Jordanian Journal of Computers and Information Technology, 7(2), 192–205. https://doi.org/10.5455/jjcit.71-1615204606

[25] Lutf, M., You, X., Cheung, Y. M., & Chen, C. P. (2014). Arabic font recognition based on diacritics features. Pattern Recognition, 47(2), 672–684. https://doi.org/10.1016/j.patcog.2013.09.029

[26] Kada, B., Mohammed, A., & Abdelmajid, B. (2025). An optimized approach for handwritten Arabic character recognition based on the SVM classifier. Engineering, Technology & Applied Science Research, 15(2), 22232–22238. https://doi.org/10.48084/etasr.9292

[27] Cheriet, M., Al-Badr, B., & Suen, C. Y. (2012). A robust word spotting system for historical Arabic manuscripts. In Guide to OCR for Arabic scripts (pp. 453–484). Springer London. https://doi.org/10.1007/978-1-4471-4072-6_18

[28] El-Sawy, A., Loey, M., & El-Bakry, H. (2017). Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research, 5, 11–19.

[29] Yousfi, S., Berrani, S. A., & Garcia, C. (2015). ALIF: A dataset for Arabic embedded text recognition in TV broadcast. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 1221–1225). IEEE. https://doi.org/10.1109/ICDAR.2015.7333958

[30] Al-Ohali, Y., Cheriet, M., & Suen, C. Y. (2009). Arabic handwriting recognition using baseline-dependent features and hidden Markov modeling. Pattern Recognition, 42(10), 2854–2869. https://doi.org/10.1016/j.patcog.2009.04.003

[31] Abed Hussein, B., & Malik, L. (2020). An overview on Arabic handwriting recognition and database. International Journal of Electrical & Computer Engineering, 10(1), 105–114. https://doi.org/10.11591/ijece.v10i1.pp105-114

[32] Mesleh, A. M., & Khorsheed, M. S. (2022). A deep learning approach for Arabic handwriting recognition: A survey. IEEE Access, 10, 53215–53227. https://doi.org/10.1109/ACCESS.2022.3177818

[33] Liu, W., Chen, W., Wang, C., Mao, Q., & Dai, X. (2021). Capsule embedded ResNet for image classification. In Proceedings of the 2021 5th International Conference on Computer Science and Artificial Intelligence (pp. 143–149). ACM. https://doi.org/10.1145/3507548.3507571

[34] Kamal, A. (2024). Enhancing lung cancer detection through a novel CapsuleNet-ResNet fusion model: A comparative study on accuracy and robustness. International Journal of Intelligent Systems and Applications in Engineering, 12(3), 4357–4365.

[35] Allenki, J., & Soni, H. K. (2024). Enhancing liver tumour segmentation in CT images using dilated residual capsule networks. Journal Européen des Systèmes Automatisés, 57(6), 1775–1783. https://doi.org/10.18280/jesa.570625

[36] Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. arXiv preprint arXiv:1710.09829. https://arxiv.org/abs/1710.09829

[37] Boufenar, C., Kerboua, A., & Batouche, M. (2018). Investigation on deep learning for off-line handwritten Arabic character recognition. Cognitive Systems Research, 50, 180–195. https://doi.org/10.1016/j.cogsys.2017.11.002

[38] Boufenar, C., & Batouche, M. (2017). Investigation on deep learning for off-line handwritten Arabic character recognition using Theano research platform. In 2017 Intelligent Systems and Computer Vision (ISCV) (pp. 1–6). IEEE. https://doi.org/10.1109/ISACV.2017.8054968

[39] Balaha, H. M., Ali, H. A., Youssef, E. K., Elsayed, A. E., Samak, R. A., Abdelhaleem, M. S., ... & Mohammed, M. M. (2021). Recognizing Arabic handwritten characters using deep learning and genetic algorithms. Multimedia Tools and Applications, 80(21), 32473–32509. https://doi.org/10.1007/s11042-021-11103-2

[40] Boufenar, C., & Batouche, M. (2016). OIHACDB: A new database for offline isolated handwritten Arabic character recognition. In Actes des Posters du Treizième Colloque sur l’Optimisation et les Systèmes d’Information-COSI’2016 (p. 20). Université Ferhat Abbas Sétif.