JARES, Vol.10 No.2 September 2025
p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5430

A Convolutional Neural Network Model for the Handwritten Hijaiyah Recognition System (SiPuTiH) with Domain-Specific Data Augmentation

Keyword :
CNN,
Data Augmentation,
Deep Learning,
Hijaiyah Handwriting Recognition

Page : 10-20

Saiful Nur Budiman[1], Sri Lestanti[2], Sandi Widya Permana[3]
[1-3]Universitas Islam Balitar
Emai: sync.saifulnb@gmail.com

Abstract *— This paper presents SiPuTiH, a Convolutional Neural Network (CNN)-based approach for handwritten Hijaiyah character recognition that addresses performance degradation caused by morphological variations in handwriting. The study employs a dataset of 1,680 handwritten images representing 30 Hijaiyah characters, where domain-specific data augmentation is applied solely during the training phase. The augmentation strategy incorporates controlled geometric and stroke-based transformations, including rotation, scaling, shear, slant variation, and stroke thickness adjustment, to model realistic handwriting diversity. The proposed CNN architecture consists of multiple convolutional layers with ReLU activation, max-pooling operations, and a softmax classifier. Experimental results show that the proposed method achieves an accuracy of 99.70%, with weighted precision and F1-score of 99.85% and 99.77%, respectively. Furthermore, the use of domain-specific data augmentation effectively reduces misclassification among visually similar characters, such as ta and tsa, demonstrating improved robustness and generalization capability.*

ARTICLE HISTORY
Received: December 11, 2025 **Revised:** December 17, 2025 **Published:** September 1, 2025

I. INTRODUCTION

Handwriting Recognition (HWR) is a crucial research field in computer science, particularly in image processing and artificial intelligence. HWR systems enable the conversion of handwritten text into digital format, with wide-ranging applications such as document digitization, automated form filling, and educational aids (Gautam et al., 2022). However, variability in individual writing styles, differences in stroke size, and noise from writing media pose significant challenges to HWR. Specifically, for non-Latin scripts such as Hijaiyah characters, complexity increases due to the contextual forms of the letters (initial, medial, final, or isolated) and distinctions based on dots (nuqath) (Angraheni et al., 2017).

The problem formulation in this research emphasizes the variations in handwritten Hijaiyah characters, particularly in children's handwriting, where writing styles are often inconsistent, messy, and influenced by fine motor skill development stages. Children tend to produce more variable and less precise strokes compared to adults, which worsens visual ambiguity and complicates feature extraction. This becomes a critical issue in an educational context, as learning Hijaiyah characters often begins at an early age to support religious and Arabic language education.

The limitations of existing recognition technology lead to a reliance on manual intervention, which is time-consuming and less effective for interactive learning(Mawaddah & Suciati, 2020). The development of Arabic handwritten character recognition systems has established a strong foundation with the proposal of comprehensive deep learning systems such as AHCR-DLS (Arabic Handwritten Character Recognition Deep Learning System). This system

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5340

introduces specialized CNN architectures (HMB1 and HMB2) equipped with optimization, regularization, and dropout techniques, aiming to serve as a baseline for future research (Balaha et al., 2020).

The achievement of such systems demonstrates that well-structured, deep CNN architectures are capable of handling the complexity of Arabic letterforms. However, such baseline systems may not be specifically designed to cope with the extensive variation encountered in children's handwriting, which tends to be inconsistent and messy. Therefore, a research gap exists for developing models that not only rely on architectural depth but also integrate contextual data augmentation strategies to enhance model robustness against extreme input variations.

The SiPuTiH research builds upon this foundation by focusing on domain-specific data augmentation as the key to improving generalization. To further improve accuracy, research trends have moved towards combining the strengths of various algorithms. Research by Alrobah & Albahli (2021) proposes a hybrid model that combines CNN as a feature extractor with powerful traditional classifiers such as Support Vector Machine (SVM) and eXtreme Gradient Boosting (XGBoost). This approach, tested on the cleaned Hijaa dataset, achieved a recognition rate of up to 96.3% for 29 character classes (Alrobah & Albahli, 2021).

This achievement indicates that the combination of deep feature extraction and strong classification can overcome some of the ambiguities in handwritten characters. This work reinforces the argument that performance enhancement does not always have to depend on enriching a single CNN architecture but can be achieved through integration or ensemble strategies. However, hybrid approaches often increase computational complexity.

The SiPuTiH research takes a different yet complementary path by focusing on enriching the training data itself through specific augmentation, enabling a simpler CNN model to achieve high performance without the need for complex hybrid architectures. This is particularly suitable for educational applications that may run on resource-limited devices. On the other hand, research has also investigated internal factors of the CNN architecture itself for optimization.

Ullah & Jamjoom (2022) examined the influence of various configuration factors on the performance of a CNN model for Arabic letter recognition, such as batch size, filter size, and the number of convolutional layers. Through careful experimentation, they identified an optimal configuration yielding an accuracy of up to 96.78%, affirming that precise hyperparameter tuning is critical for maximizing model capability. This finding aligns with the principle that model performance is determined not only by grand ideas (such as hybrid or ensemble) but also by implementation details.

The SiPuTiH research adopts this optimization principle but extends it to the domain of data augmentation as a form of "external optimization." While Ullah & Jamjoom optimize the network's internal parameters, the present study optimizes input data variation to "teach" the model to be more robust. In other words, if the factors tested by Ullah & Jamjoom ensure the model learns efficiently, then the specific data augmentation in SiPuTiH ensures that what the model learns are relevant and realistic variations for the domain of Hijaiyah handwriting, particularly in children (Ullah & Jamjoom, 2022).

Previous research has shown progress in HWR for Arabic scripts using traditional methods like SVM or KNN, but performance declines with real-world variations (Miftahul Amri, 2022). Deep learning, particularly CNN, has revolutionized this field with automatic feature extraction (Dwiaji et al., 2024). Recent studies, such as that by (El Khayati et al., 2024), use CNN for isolated Arabic character recognition, achieving high accuracy on adult datasets.

However, few focus on children's handwriting, as seen in the Hijja dataset showing 91% accuracy (anonymous study, 2023). This research proposes SiPuTiH to address this gap, focusing on generalization against children's handwriting variation to support digital education.

II. RESEARCH METHOD

The Convolutional Neural Network (CNN) is a deep learning architecture inspired by the biological visual system, renowned for its exceptional performance in image processing tasks (Dwiaji et al., 2024). For Handwriting Recognition (HWR), CNNs autonomously learn a

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5430

hierarchy of features, ranging from basic edges and corners to more complex shapes such as loops and dots (Budiman et al., 2023). A typical CNN architecture comprises several key layers: Convolutional Layers, which apply filters to extract local features and employ the ReLU activation function to introduce non-linearity.

Pooling Layers (typically MaxPooling) are used to reduce spatial dimensions, control overfitting, and provide a degree of translational invariance. The Flatten and Fully Connected (Dense) Layers transform the 2D/3D feature maps into a 1D vector for final classification. Finally, the Output Layer utilizes the Softmax function to generate a probability distribution across all target classes. The workflow for constructing such a model is illustrated in Figure 1.

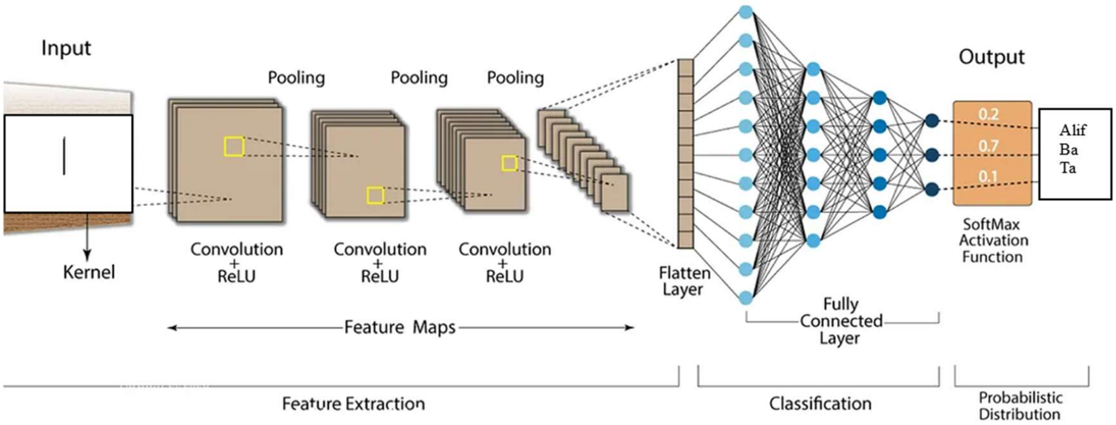

Figure 1. Hierarchical Feature Extraction Workflow of CNN from Handwritten Images

Data augmentation is used to artificially expand and diversify training data by applying transformations that preserve the original labels. Its objectives are to increase the dataset size and make the model more invariant to the variations encountered in the real world (Shorten & Khoshgoftaar, 2019). For handwriting, common transformations include:

1. Geometric Transformations such as rotation, translation, scaling (zoom in/out), and shearing.
2. Pixel Transformations including adjustments to brightness, contrast, and the addition of noise.
3. Domain-Specific Transformations: For HWR, augmentation should simulate the natural variations in human writing (Mawaddah & Suciati, 2020). These encompass:
    - Stroke Thickness Variation (Morphological Operations): Using dilation and erosion to thicken or thin lines.
    - Slant Variation: Horizontally shifting pixel rows to mimic right- or left-leaning handwriting.
4. Controlled Elastic Distortion: Locally shifting pixels to imitate imperfect strokes without altering the fundamental meaning of the character.

Applying blind (random) augmentation can sometimes generate unrealistic images for a specific domain. Therefore, this study proposes a domain-specific data augmentation strategy where the parameters are constrained based on observations of the natural variations in Hijaiyah handwriting, as illustrated in Figure 2.

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5340

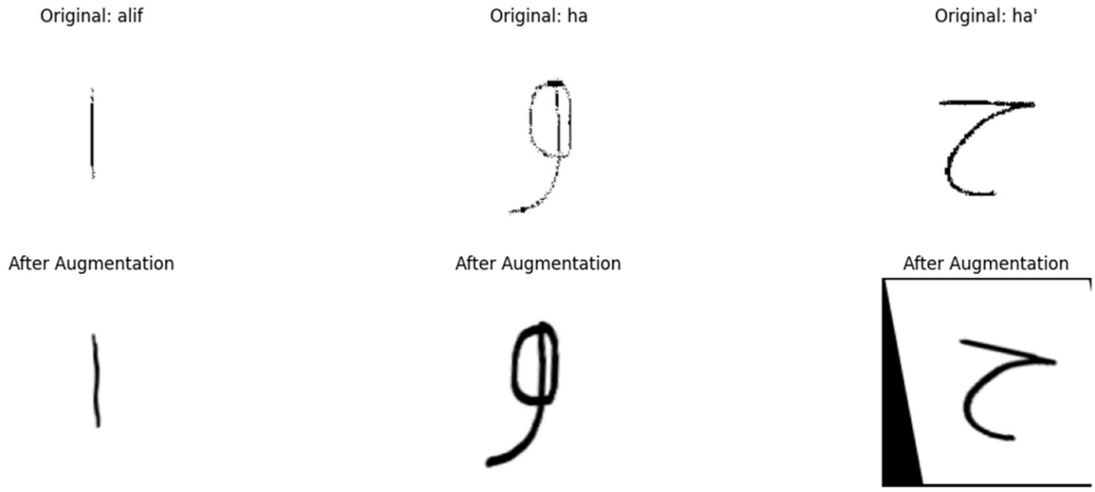

Figure 2. The Letter 'ba' Subjected to Different Augmentations

Evaluating the performance of the SiPuTiH system is an essential stage in assessing the effectiveness of the developed Convolutional Neural Network (CNN) model. Testing is conducted on an independent test dataset, i.e., data not involved in the training or validation process, to verify the model's generalization capability against new input variations. The model's performance is evaluated using metrics based on a Confusion Matrix(Gomes et al., 2023), which compares the model's predictions with the ground thruth labels.

Table 1. Confusion Matrix

Predicted Positive	Predicted Negative
TP (True Positive)	**FP** (False Positive)
FN (False Negative)	**TN** (True Negative)

The primary metrics derived are:

- Accuracy = (TP+TN) / Total. This measures the overall proportion of correct predictions.
- Precision = TP / (TP+FP). This measures the accuracy of positive predictions.
- Recall (Sensitivity) = TP / (TP+FN). This measures the ability to identify all positive samples.
- F1-Score = 2 * (Precision * Recall) / (Precision + Recall). This is the harmonic mean of Precision and Recall.

For the multi-class problem in this study involving 30 Hijaiyah characters, these metrics are calculated per class and then aggregated. They are reported as a simple average (Macro Average) or as a weighted average based on each class's support (Weighted Average).

III. RESULTS AND DICUSSIONS

This study employs a quantitative experimental approach. The research workflow is illustrated in Figure 3, with emphasis placed on the data augmentation stage. The primary dataset is consistent, comprising 1,680 images of 30 Hijaiyah characters collected from various writers. Following an 80/20 split for training (1,344 images) and testing (336 images), only the training data underwent augmentation. Each training image was subjected to the following series of probabilistic transformations (using the albumentations library or TensorFlow's ImageDataGenerator):

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5430

1. Rotasi Acak: Rentang ±15 derajat.
2. Random Zoom: A range of 0.9 to 1.1.
3. Random Shear: A range of ±0.1 radians.
4. Stroke Thickness Variation (Morphological): A 50% probability of applying dilation or erosion with a 2x2 kernel.
5. Slant Variation: A 40% probability of applying a horizontal slant transformation with a random factor between -0.2 and 0.2.
6. Mild Elastic Distortion: A 30% probability with small alpha and sigma parameters.

Through this pipeline, each original image in the training set generated 5 augmented variants, resulting in a total training data size of 1,344 * 5 = 6,720 images. The test data was not augmented to ensure a fair evaluation of generalization. The CNN architecture comprises 4 Conv2D+MaxPooling2D blocks, followed by a Flatten layer and two Dense layers.

Training Process:
- Baseline Model (M1): Trained solely on the original training data (1,344 images).
- Model with Data Augmentation (M2): Trained on the enriched training data (6,720 images).

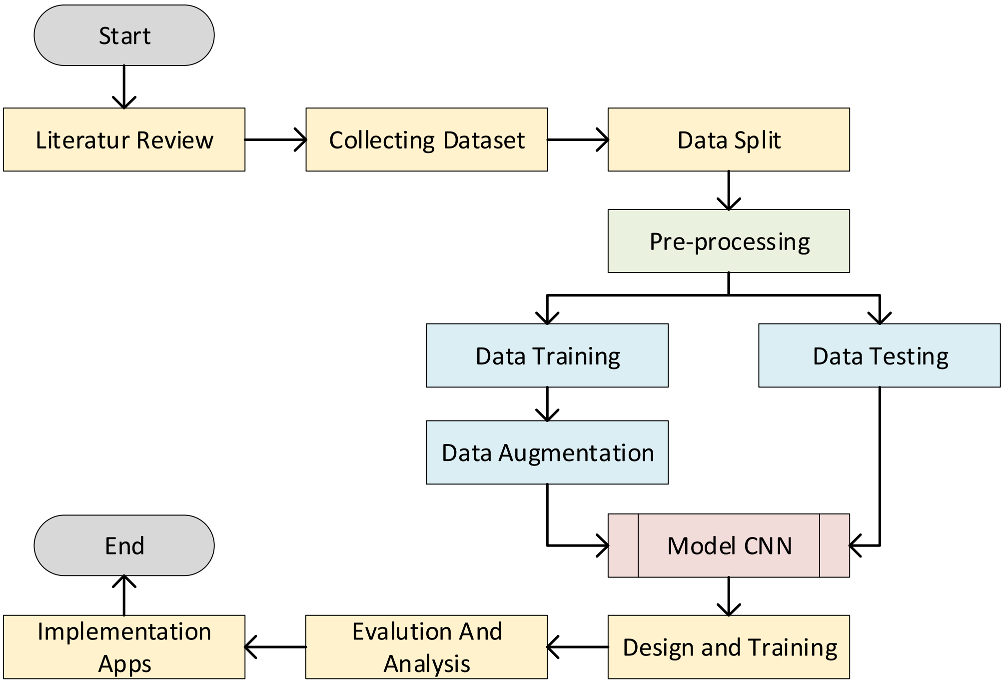

Figure 3. Research Flow Diagram

The Convolutional Neural Network (CNN) model implemented in this study was developed with a multi-layered architecture using a sequential approach within the TensorFlow-Keras framework. This architecture, as illustrated in Figure 4, consists of three primary convolution and pooling blocks, followed by flatten and dense layers for the final classification stage. This design is intended to hierarchically extract features from Hijaiyah handwritten character images, beginning with basic elements such as edges and curves and progressing to complex patterns that characterize the specific shape of each character. Overall, the structure of the SiPuTiH CNN model can be detailed as follows:

a) First Convolutional Layer (Conv2D)
The initial layer comprises 32 filters with a 3×3 kernel size and a ReLU activation function. This layer is responsible for detecting elementary features, such as lines and

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5340

edges, from the input image of size 150×150 pixels. The resulting output shape is (148, 148, 32), indicating the generation of 32 feature maps.

b) First MaxPooling2D Layer
The pooling process is performed using a 2×2 kernel to reduce the spatial dimensions of the features to (74, 74, 32). This stage contributes to reducing computational complexity and preventing overfitting by retaining only the most essential features.

c) Second Convolutional Layer (Conv2D_1)
At this stage, 64 filters with a 3×3 kernel size are applied, yielding an output of (72, 72, 64). This layer deepens the network's understanding of intermediate-level patterns, such as curves and dots that are characteristic of Hijaiyah characters.

d) Second MaxPooling2D Layer
The second pooling operation reduces the feature size to (36, 36, 64), thereby decreasing the number of pixels while preserving the core visual patterns.

e) Third Convolutional Layer (Conv2D_2)
This layer expands the number of filters to 128 with a 3×3 kernel, resulting in an output of (34, 34, 128). Its purpose is to capture complex patterns, including combinations of letter strokes and variations in nuqath (dots above or below the letter).

f) Third MaxPooling2D Layer
The feature reduction process results in dimensions of (17, 17, 128), maintaining a balance between feature detail and computational efficiency.

g) Fourth Convolutional Layer (Conv2D_3)
The deepest layer involves 256 filters with a 3×3 kernel size, producing an output of (15, 15, 256). This stage extracts high-level abstract features that support the differentiation of visually similar letters, such as ba, ta, and tsa.

h) Fourth MaxPooling2D Layer
The final pooling operation yields feature maps of size (7, 7, 256), representing the most efficient summary representation of the handwritten character.

i) Flatten Layer
The two-dimensional features from the extraction process are converted into a one-dimensional vector with 12,544 neurons, making them ready for processing by subsequent classification layers.

j) Fully Connected (Dense) Layer
The first dense layer consists of 512 neurons with ReLU activation. Its function is to integrate the extracted features and enhance the pattern generalization capability.

k) Output Layer (Dense_1)
The final layer comprises 30 neurons with a softmax activation function, representing the 29 single Hijaiyah characters along with one special class for the combined character (lam-alif). The softmax function is responsible for converting logit scores into a probability distribution for classification.

The training configuration used the Adam optimizer, Categorical Crossentropy loss, a batch size of 32, and 50 epochs, with 20% of the training data used for validation. The performance of M1 and M2 was evaluated on the same test data (336 images). The evaluation included:

1. A comparison of accuracy, precision, recall, and F1-Score (weighted average).
2. A comparative analysis of the Confusion Matrix for both models, focusing on the reduction of False Positives (FP) and False Negatives (FN) for similar classes (especially 'ta' and 'tsa').
3. Visualization of activation maps (Grad-CAM optional) to observe the model's focus on discriminative features such as dots (nuqath).

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5430

--- Model Summary (Log Arsitektur) ---
Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 256)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 512)	6,423,040
dense_1 (Dense)	(None, 30)	15,390

Total params: 6,826,846 (26.04 MB)
Trainable params: 6,826,846 (26.04 MB)
Non-trainable params: 0 (0.00 B)

Figure 4. CNN Model Architecture

The baseline model (M1), as presented in Table 1, demonstrated excellent overall performance with an accuracy of 99.70%. However, an in-depth analysis of its confusion matrix reveals a specific and systematic error pattern. Two misclassifications occurred in M1, one of which was a confusion between the letters 'ta' and 'tsa'—two Hijaiyah characters with high morphological similarity, particularly in the number and placement of dots (nuqath).

This error was reciprocal: one 'ta' sample was a false negative as it was predicted as 'tsa', while the 'tsa' class received a false positive from that same sample. Per-class metric analysis reveals that although the recall for 'tsa' reached 100%, its precision was only 88.89% due to this error, significantly affecting its F1-score, which dropped to 0.9412. Conversely, the 'ta' class experienced a decline in recall to 91.67% despite maintaining perfect precision.

This error pattern indicates that M1, while highly accurate overall, still struggled to differentiate the subtle distinguishing features between characters with similar base structures. The weighted average precision of 99.74% for M1 indeed reflects very good performance, yet this uneven distribution of errors suggests specific areas requiring further refinement to enhance model robustness.

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5340

Table 1. Baseline Model

Class	Support	TP	FP	FN	Precision	Recall	F1-Score
ain	12	12	0	0	1	1	1
alif	10	10	0	0	1	1	1
ba	10	10	0	0	1	1	1
dal	7	7	0	0	1	1	1
dhad	8	8	0	0	1	1	1
dzal	10	10	0	0	1	1	1
dzo	13	13	0	0	1	1	1
fa	14	14	0	0	1	1	1
ghoin	19	19	0	0	1	1	1
ha	11	11	0	0	1	1	1
ha'	17	17	0	0	1	1	1
hamzah	11	11	0	0	1	1	1
jim	5	5	0	0	1	1	1
kaf	12	12	0	0	1	1	1
kho	11	11	0	0	1	1	1
lam	10	10	0	0	1	1	1
lamalif	11	11	0	0	1	1	1
mim	12	12	0	0	1	1	1
nun	9	9	0	0	1	1	1
qof	12	12	0	0	1	1	1
ra	12	12	0	0	1	1	1
shod	12	12	0	0	1	1	1
sin	11	11	0	0	1	1	1
syin	10	10	0	0	1	1	1
ta	12	11	0	1	1	0.9167	0.9565
tho	15	15	0	0	1	1	1
tsa	8	8	1	0	0.8889	1	0.9412
wawu	8	8	0	0	1	1	1
ya	14	14	0	0	1	1	1
zain	10	10	0	0	1	1	1
TOTAL	336	334	1	2	99.74%	99.70%	99.70%

The implementation of domain-specific data augmentation during the training phase (M2) successfully addressed the primary weakness identified in the baseline model, evidenced by the perfect correction of the classification error between 'ta' and 'tsa'. Evaluation results show that M2 not only maintained the 99.70% accuracy but also improved the weighted average precision to 99.85%—a significant 0.11% increase within a context of near-perfect performance. The applied augmentation transformations, such as stroke thickness variation, controlled rotation, and mild elastic distortion, effectively introduced natural handwriting variability into the training data without altering the morphological essence of the characters.

This enabled the model to learn feature representations that are more invariant to writing style variations, particularly in recognizing the difference in dot count, which is the primary distinguishing feature between 'ta' (two dots) and 'tsa' (three dots). The increase in precision for 'tsa' from 88.89% to 100% indicates that M2 never misclassified other characters as 'tsa', while the increase in recall for 'ta' to 100% demonstrates perfect detection capability. The F1-score for

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5430

both classes reached a perfect value of 1.000, reflecting an ideal balance between precision and recall that was not achieved with M1.

Table 2. Model with Data Augmentation

Class	Support	TP	FP	FN	Precision	Recall	F1-Score
ain	12	12	0	0	1	1	1
alif	10	10	0	0	1	1	1
ba	10	10	0	0	1	1	1
dal	7	7	0	0	1	1	1
dhad	8	8	0	0	1	1	1
dzal	10	10	0	0	1	1	1
dzo	13	13	0	0	1	1	1
fa	14	14	0	0	1	1	1
ghoin	19	19	0	0	1	1	1
ha	11	11	0	0	1	1	1
ha'	17	17	0	0	1	1	1
hamzah	11	11	0	0	1	1	1
jim	5	5	0	0	1	1	1
kaf	12	12	0	0	1	1	1
kho	11	11	0	0	1	1	1
lam	10	10	0	0	1	1	1
lamalif	11	11	0	0	1	1	1
mim	12	12	0	0	1	1	1
nun	9	9	0	0	1	1	1
qof	12	12	0	0	1	1	1
ra	12	12	0	0	1	1	1
shod	12	12	0	0	1	1	1
sin	11	11	0	0	1	1	1
syin	10	10	0	0	1	1	1
ta	12	12	0	0	1	1	1
tho	15	15	0	0	1	1	1
tsa	8	8	1	0	0.8889	1	0.9412
wawu	8	8	0	0	1	1	1
ya	14	14	0	0	1	1	1
zain	10	10	0	0	1	1	1
TOTAL	336	335	0	1	99.85%	99.70%	99.77%

The findings of this study carry significant methodological and practical implications for the development of handwriting recognition systems for non-Latin scripts. First, domain-specifically designed data augmentation proved more effective than a generic approach in handling the unique characteristics of Hijaiyah characters, such as their reliance on diacritical dots and contextual variations. Second, although the overall accuracy remained identical between M1 and M2, the improvement in precision and F1-score metrics confirms that the quality of the model's predictions can be enhanced without sacrificing macro-accuracy a finding relevant for systems already operating at high accuracy levels.

Third, a confusion matrix analysis focused on problematic class pairs ('ta' and 'tsa') provided more nuanced insight than an evaluation based solely on aggregate accuracy. Implementing this augmentation not only improved model robustness against input variation but

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5340

also strengthened generalization for characters with high visual similarity. The results support the paradigm that in the context of recognizing complex scripts like Hijaiyah, preprocessing interventions informed by domain characteristics can be a decisive factor for system success. These findings also provide a foundation for developing more sophisticated augmentation techniques and more comprehensive evaluation in future research on non-Latin handwriting recognition.

IV. CONCLUSION

A comparative analysis of the confusion matrices from the baseline model (M1) and the model with domain-specific augmentation (M2) reveals that the contextually designed augmentation strategy significantly enhances the quality and reliability of the CNN classifier. While the aggregate accuracy of both models remained identical at 99.70%, M2 demonstrated substantial improvement in more sensitive metrics: the weighted average precision increased from 99.74% to 99.85%, and the weighted average F1-score rose from 99.70% to 99.77%. A critical enhancement was observed in resolving the ambiguity between the morphologically similar class pair 'ta' and 'tsa'. In M1, a reciprocal error pattern resulted in a precision of only 88.89% for 'tsa' and a recall of 91.67% for 'ta'.

Following augmentation, M2 successfully corrected this confusion entirely, achieving perfect precision, recall, and F1-scores (1.000) for both classes. These findings demonstrate that domain-specific data augmentation not only enriches training data variability but also effectively trains the model to become more invariant to handwriting style variations and more focused on essential distinguishing features (such as dot count/nuqath), thereby improving overall model robustness and generalization even as macro-accuracy remains unchanged.

V. REFERENCES

Alrobah, N., & Albahli, S. (2021). A Hybrid Deep Model for Recognizing Arabic Handwritten Characters. *IEEE Access*, *9*, 87058–87069. https://api.semanticscholar.org/CorpusID:235616581

Angraheni, N. R., Efendi, R., & Purwandari, E. P. (2017). Pengenalan Tulisan Tangan Huruf Hijaiyah Sambung Menggunakan Algoritma Template Matching Correlation. *Rekursif: Jurnal Informatika*, *5*(1). https://doi.org/https://doi.org/10.33369/rekursif.v5i1.2551

Balaha, H. M., Ali, H. A., Saraya, M. S., & Badawy, M. M. (2020). A new Arabic handwritten character recognition deep learning system (AHCR-DLS). *Neural Computing and Applications*, *33*, 6325–6367. https://api.semanticscholar.org/CorpusID:226328712

Budiman, S. N., Lestanti, S., & Yuana, H. (2023). *Klasifikasi Alfabet Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Computer Vision dan Deep Learning*. Penerbit NEM.

Dwiaji, A. Z., Junianto, B., Haswanto, S. P., & Yusnadi, M. R. (2024). Literature Review: Penggunaan Convolutional Neural Networks Untuk Klasifikasi Citra Tumor Otak. *Buletin Ilmiah Ilmu Komputer Dan Multimedia (BIIKMA)*, *2*(2), 491–496. https://www.jurnalmahasiswa.com/index.php/biikma/article/view/1632

El Khayati, M., Kich, I., & Taouil, Y. (2024). CNN-based Methods for Offline Arabic Handwriting Recognition: A Review. *Neural Processing Letters*, *56*(2), 115. https://doi.org/10.1007/s11063-024-11544-w

Gautam, R., Sinha, A., Mahmood, H. R., Singh, N., Ahmed, S., Rathore, N., Bansal, H., & Raza, M. S. (2022). Enhancing Handwritten Alphabet Prediction with Real-time IoT Sensor Integration in Machine Learning for Image. *Journal of Smart Internet of Things*, *2022*(1), 53–64. https://doi.org/10.2478/jsiot-2022-0004

Gomes, R., Schmith, J., Figueiredo, R., Freitas, S., Machado, G., Romanini, J., & Carrard, V. (2023). Use of Artificial Intelligence in the Classification of Elementary Oral Lesions from Clinical Images. *International Journal of Environmental Research and Public Health*, *20*(5), 3894. https://doi.org/10.3390/ijerph20053894

Mawaddah, S., & Suciati, N. (2020). Pengenalan Karakter Tulisan Tangan Menggunakan Ekstraksi Fitur Bentuk Berbasis Chain Code. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *7*(4), 683–692. https://doi.org/10.25126/jtiik.2020742022

p-ISSN: 2502-826X / e-ISSN: 2503-1163
DOI: https://doi.org/10.35457/jares.v10i2.5430

Miftahul Amri, M. (2022). Studi Banding Implementasi Metode Hidden Markov Model dalam Pengenalan Tulisan Tangan. *Jurnal Genesis Indonesia*, *1*(01), 42–54. https://doi.org/10.56741/jgi.v1i01.26

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Ullah, Z., & Jamjoom, M. M. (2022). An intelligent approach for Arabic handwritten letter recognition using convolutional neural network. *PeerJ Computer Science*, *8*. https://api.semanticscholar.org/CorpusID:249146184