SEMESTA
Jurnal Ilmu Pendidikan dan Pengajaran
https://ejournal.ahs-edu.org/index.php/semesta/about

Mei 2026
Volume 4 Nomor 2
e-ISSN: 2986-8874

Recognition of Arabic Vocabulary Based on Machine Learning Using a Convolutional Neural Network on Mobile Devices

[1]Rahmat*, [1]Fauzi, [2]Muhammad Husni Mubarak
[1]Fakultas Tarbiyah dan Ilmu Keguruan, Institut Agama Islam Negeri Ternate, Indonesia
[2]Fakultas Tarbiyah dan Ilmu Keguruan, Institut Agama Islam Negeri Manado, Indonesia

DOI: https://doi.org/10.70115/semesta.v4i2.439

Article Info

Article History
Received: April 9, 2026
Accepted: June 3, 2026
Published: June 4, 2026

Keywords
mufradat recognition; convolutional neural network; transfer learning; MobileNetV2; mobile learning

Corresponding Author
Rahmat
Fakultas Tarbiyah dan Ilmu Keguruan, Institut Agama Islam Negeri Ternate, Indonesia
*E-mail:
rahmat@iain-ternate.ac.id

Abstract

This study develops and evaluates machine-learning models based on Convolutional Neural Networks (CNNs) for recognizing images of Arabic vocabulary (mufradat) and for deploying these models on resource-constrained mobile devices. Whereas most prior research on Arabic-script recognition has concentrated on isolated characters executed on desktop hardware, the recognition of whole words—whose connected and visually similar glyphs increase classification difficulty—remains comparatively underexplored, particularly for on-device educational use. To address this gap, the study contributes (i) a purpose-built image dataset of fifteen academic Arabic words, (ii) a systematic comparison between a CNN trained from scratch and a MobileNetV2 transfer-learning model, and (iii) a quantified analysis of mobile deployment. An experimental approach was adopted using 3,000 images (200 per class) compiled from tablet handwriting and Microsoft Word screen-captured images, partitioned through a stratified 70/15/15 training, validation, and testing split. Both models were trained using the Adam optimizer (learning rate 1×10^{-4}), a batch size of 32, and 50 epochs. The from-scratch five-convolution model attained 94.4% test accuracy (loss 0.26; macro-averaged F1-score 0.95), whereas the MobileNetV2 model attained 99.1% accuracy (loss 0.20; macro-averaged F1-score 0.99). After conversion to TensorFlow Lite, the MobileNetV2 model required only 9.1 MB of storage and 42 ms per inference on a mid-range Android device, compared with 103 MB and 180 ms for the from-scratch model, confirming its suitability for real-time use. The findings demonstrate that transfer learning achieves higher accuracy with markedly fewer parameters and a smaller computational footprint, providing an efficient foundation for mobile-assisted Arabic vocabulary learning.

INTRODUCTION

Amid the rapid advancement of digital technology, mobile devices have become an integral part of everyday life. Their portability and affordability open substantial opportunities to embed sophisticated technologies such as machine-learning-based vocabulary recognition into ordinary user experiences. The recognition of Arabic vocabulary on mobile devices is becoming increasingly relevant as interest in understanding and using the Arabic language continues to grow. Within this context, the application of machine learning can contribute significantly to facilitating Arabic-language learning through a more interactive and responsive approach (Halim, 2018; Haniah, 2014).

Indonesia is the country with the largest Muslim population in the world, and the Muslim community is closely associated with the use of the Arabic language (The Royal Islamic Strategic Studies Centre, n.d.). Consequently, Arabic is among the foreign languages most widely studied in Indonesia, and a variety of studies have been conducted to support the Arabic-learning process (Agusten & Supriyatin, 2015). Vocabulary in Arabic, referred to as mufradat, denotes the set of words known by an individual as part of a particular language. Mufradat constitutes one of the three core linguistic elements that must be mastered; it is employed in both written and spoken language and is a fundamental component for developing a learner's Arabic-language competence (Mustofa, 2017). A sound command of mufradat enables effective communication across diverse contexts, both in writing and in daily conversation.

One of the more challenging contemporary problems in pattern recognition is the recognition of Arabic text. This research area remains broad and insufficiently resolved owing to several factors, including the similarity of characters, the very large number of words, and the wide variety of fonts (Rahal et al., 2021; Faizullah et al., 2023). The present era of the Fourth Industrial Revolution is marked by the rapid growth of Artificial Intelligence (AI), a field of study focused on solving cognitive problems generally associated with human intelligence, such as deep learning and visual pattern recognition. Within AI lies the subfield of machine learning, which has advanced considerably with the emergence of deep learning. Deep learning models employing the Convolutional Neural Network (CNN) algorithm possess a greater number of layers and are therefore highly capable in the domain of image-based pattern recognition (Fauzi et al., 2021).

Several prior studies have addressed Arabic character and word recognition. Shareef and Irhayim (2021) developed an image-recognition system using a CNN, but the study focused solely on recognizing Arabic characters (hijaiyah letters), reaching an accuracy of 94.9%; its principal limitations were the inability to translate whole words and the requirement for substantial desktop computing resources. Kasim and Nugraha (2021) built a handwritten Arabic-script recognition system using a CNN that achieved 78.10% accuracy; this comparatively low result was attributed in part to the absence of data augmentation, a transformation technique that improves model generalization and mitigates overfitting. Akil and Chaidir (2021) applied a CNN architecture to hijaiyah-letter recognition, obtaining a highest accuracy of 91% with three convolutional layers. More recent investigations have continued to refine character-level recognition: Altwaijry and Al-Turaiki (2021) introduced the Hijja dataset and a from-scratch CNN reaching up to 97% on standard letter benchmarks, while

Ullah and Jamjoom (2022) reported 96.8% on the AHCD dataset using a CNN augmented with image transformations. Wagaa et al. (2022) further demonstrated that a careful combination of data augmentation, dropout regularization, and optimizer selection substantially improves Arabic-letter classification.

Most of these earlier studies, however, have concentrated narrowly on Arabic-character recognition (El-Sawy et al., 2017; Mudhsh & Almodfer, 2017; Kasim & Nugraha, 2021; Akil & Chaidir, 2021). Unlike character recognition, which targets isolated glyphs, word recognition is considerably more complex because a single word comprises more than one character. In Arabic, most characters within a word are connected to one another, and certain characters exhibit highly similar features (Lamsaf et al., 2022; Hjaiej et al., 2025). These properties make recognizing Arabic vocabulary substantially more difficult than recognizing individual letters. A further pervasive challenge is overfitting, in which a trained model memorizes the training data too closely and fails to generalize to unseen data (Buduma & Locascio, n.d.).

Pengenalan citra merupakan bagian dari ilmu *Computer Vision* yang dalam beberapa tahun terakhir menjadi semakin penting dan efektif. Pengenalan citra bekerja dengan melakukan proses klasifikasi citra berdasarkan ciri tertentu melalui beberapa tahapan, meliputi pra pemrosesan citra, segmentasi citra, ekstraksi fitur utama, dan identifikasi kecocokan (Khan et al., n.d.). *Convolutional Neural Network* (CNN) merupakan salah satu algoritma *deep learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP), dirancang khusus untuk pengolahan data dua dimensi seperti citra. Konsep kunci dalam CNN adalah penggunaan lapisan konvolusi yang memungkinkan model untuk mengekstraksi fitur secara otomatis dari data spasial melalui filter atau kernel kecil untuk mendeteksi pola atau fitur lokal. Selanjutnya, lapisan *pooling* digunakan untuk mereduksi dimensi data dan menjaga invariansi terhadap pergeseran kecil dalam fitur yang terdeteksi, serta lapisan aktivasi non-linear seperti ReLU yang memperkenalkan non-linearitas ke dalam model (Buduma & Locascio, n.d.-b).

Image recognition is a branch of computer vision that has become increasingly important and effective in recent years. It operates by classifying images according to particular features through several stages, including image preprocessing, image segmentation, principal-feature extraction, and matching identification (Khan et al., n.d.). The Convolutional Neural Network is a deep-learning algorithm derived from the Multilayer Perceptron (MLP) and designed specifically for processing two-dimensional data such as images. The key concept in a CNN is the use of convolutional layers that enable the model to extract features automatically from spatial data through small filters or kernels that detect local patterns. Pooling layers subsequently reduce the dimensionality of the data and preserve invariance to small shifts in detected features, while non-linear activation functions such as the Rectified Linear Unit (ReLU) introduce non-linearity into the model (Buduma & Locascio, n.d.; Najam & Faizullah, 2023).

Transfer learning is an important concept in machine learning whereby a model already trained on one task is used as the basis (a pre-trained model) for understanding and solving a different task. The central idea is that knowledge acquired from one task or domain can be adapted and exploited to improve a model's performance on a different task, thereby saving the time and resources required to train a model from scratch. Transfer learning is particularly valuable when the dataset available for the new task is relatively small, because reusing a pre-

trained model can prevent overfitting and help the model generalize well (Gulzar, 2023; Lahiani & Frikha, 2024). The use of mobile devices as a medium for Arabic-vocabulary learning is highly appropriate given their compact, portable form and advanced features such as built-in cameras, which allow a smartphone to function as a learning medium (Halim, 2018). Recent work has shown that lightweight, mobile-oriented CNNs including MobileNet and its successors can be fine-tuned through transfer learning to deliver high accuracy on Arabic-script tasks while remaining deployable on devices with limited resources (El Khayati et al., 2025).

From an educational-technology perspective, the integration of visual and interactive media has repeatedly been shown to strengthen learner motivation and engagement in religious and language instruction. Studies within Islamic education have reported that purpose-designed instructional media improve learning motivation and outcomes (Istiqomah et al., 2024; Nurrahmah et al., 2024), that the systematic selection of media supports effective teaching (Sumiyati et al., 2024), and that active-learning and digital-era methods are increasingly central to contemporary Islamic and language pedagogy (Arhmawati et al., 2025; Riswadi et al., 2025). A mobile, image-based mufradat-recognition tool sits squarely within this trajectory, offering an interactive medium that can be carried and used at any time.

Based on the problems outlined above, this study proposes the use of a Convolutional Neural Network architecture to develop an image-based mufradat-recognition system using machine-learning techniques deployed on mobile devices. The novelty of this study is therefore threefold and explicitly stated as follows. First, with respect to data, the study constructs a purpose-built image dataset of complete Arabic words (mufradat), rather than isolated characters, addressing the scarcity of word-level Arabic image datasets noted in the literature. Second, with respect to modeling, the study advances from character-level recognition to word-level recognition and systematically compares a CNN trained from scratch with a MobileNetV2 transfer-learning model under identical experimental conditions. Third, with respect to deployment, the study optimizes and quantifies the resulting models for on-device mobile inference reporting model size, inference latency, and memory usage an aspect left largely unexamined in prior Arabic-recognition research that targeted desktop platforms. Accordingly, the objectives of this study are: (1) to produce a machine-learning model capable of recognizing images of mufradat with high accuracy; and (2) to produce a lightweight machine-learning model capable of running on mobile devices with high accuracy.

METHOD

Research Design

This study adopted an experimental approach. The experimental method is a component of quantitative research that involves the activities of attempting, discovering, and confirming. Experimental research focuses on the cause-and-effect relationship at the core of the inquiry, whereby changes in the value of an independent variable affect the value of a dependent variable (Sugiyono, 2013). The research process began with a review of the relevant literature as a foundation for the experimental procedure, followed by the collection of Arabic-vocabulary data. After the dataset had been assembled, the work proceeded to the construction of the machine-learning models, the training and testing of the data, the evaluation of the

training and testing outcomes, and, in the final stage, on-device trials using a mobile device. The overall research workflow is presented in Figure 1.

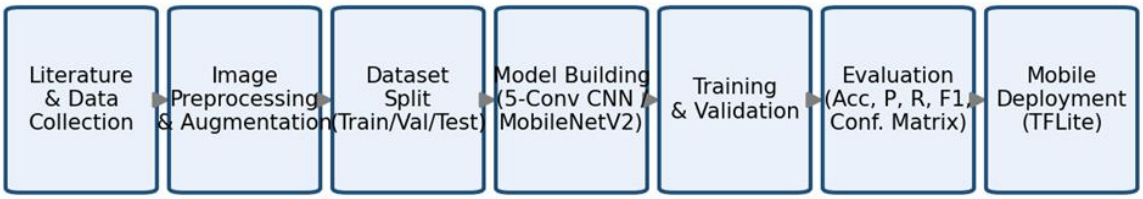

Figure 1. Overall research workflow, from data collection to mobile deployment.

Dataset and Data Collection

The data required for this study comprised primary data in the form of handwritten mufradat and secondary data obtained from other sources. The dataset consisted of a total of 3,000 images representing 15 Arabic words, with 200 images per word. During the search for an existing dataset on repositories such as Kaggle and the UCI Machine Learning Repository, no dataset specific to Arabic vocabulary was found; the datasets available online were limited to hijaiyah letters (the Arabic alphabet). The dataset for this study was therefore constructed independently and supplemented with screen-captured images of Arabic words typed in Microsoft Word. The fifteen vocabulary items, drawn from the academic environment, are listed in Table 1.

Table 1. Arabic vocabulary dataset (15 academic words).

No	Mufradat	Meaning	No	Mufradat	Meaning
1	الكلية	Faculty	9	الطريقة	Method
2	المجلة	Journal	10	السبورة	Whiteboard
3	الجامعة	Campus	11	التربية	Education
4	الكرسي	Chair	12	البحث	Research
5	المعمل	Laboratory	13	القسم	Department
6	خزانة	Cupboard	14	مركز التطويرية اللغوية	Language Dev. Center
7	المكتب	Desk	15	غرفة الإدارة	Management Room
8	مكتب المدرب	Instructor's Desk			

Data collection was carried out in two stages, yielding a balanced distribution across classes. In the first stage, the dataset was produced independently by hand: each of the fifteen Arabic words was written on a tablet screen and saved in Portable Network Graphics (PNG) format. This procedure was repeated 2,700 times in total, producing 180 images per word. In the second stage, each word was typed in several standard Arabic fonts within Microsoft Word, captured as a screen image, and saved in PNG format; this procedure was repeated 300 times in total, producing 20 images per word. The handwritten source captures natural variation in stroke shape, slant, and thickness, whereas the screen-captured source introduces typographic variation across font families. The resulting per-class distribution is summarized in Table 2; every class is exactly balanced at 200 images (180 handwritten and 20 screen-captured), which removes class-imbalance as a confounding factor in subsequent evaluation.

Table 2. Per-class data distribution and acquisition source.

Acquisition source	Images per class	Classes	Total images
Tablet handwriting (PNG)	180	15	2,700
Microsoft Word screen-captured (PNG)	20	15	300
Total	**200**	**15**	**3,000**

Image Preprocessing

After the dataset had been assembled into fifteen folders—one per word—it underwent a preprocessing pipeline comprising four steps. First, image-size standardization was applied: all images, which originally varied in dimension, were resized to a uniform 224 × 224 pixels with three colour channels (224 × 224 × 3). This standardization ensures that every image presented to the network has consistent dimensions, allowing the model to accept a fixed input size. Second, pixel-intensity normalization rescaled pixel values to the range [0, 1], so that values across images occupied a uniform range, enabling the model to learn more efficiently and improving training stability. Third, invalid or irrelevant data—corrupted, incomplete, or uninformative images—were removed from the dataset. Fourth, data augmentation was applied to the training partition to increase the effective quantity and variety of the data; the augmentation employed a zoom range of 0.2. Augmentation is a crucial technique for enhancing dataset diversity, giving the model a wider range of variation from which to learn, helping it handle diverse situations, and mitigating overfitting on a relatively small dataset (Wagaa et al., 2022).

Dataset Partitioning

To enable an objective and reproducible assessment and, in direct response to the need for a held-out evaluation set the dataset was partitioned through a stratified split that preserved the class balance within each partition. Each class of 200 images was divided into 140 images for training (70%), 30 images for validation (15%), and 30 images for testing (15%). Aggregated across the fifteen classes, this yielded 2,100 training images, 450 validation images, and 450 testing images. The training partition was used to fit the model parameters; the validation partition was used to monitor generalization during training and to guard against overfitting; and the testing partition comprising images the models never encountered during training or validation was reserved exclusively for the final performance evaluation reported in Section 3. A batch size of 32 was used throughout training, meaning that each training iteration processed 32 images before the model weights were updated.

Experiment 1: Convolutional Neural Network Trained from Scratch

In the first experiment, a model was built from scratch that is, with randomly initialized weights and no prior training using a Convolutional Neural Network architecture. The architecture, designated the 5-Conv architecture, comprises five convolutional layers, each using the parameters padding = 'same' and the ReLU activation function with a 3 × 3 kernel. Each convolutional layer is followed by a 2 × 2 max-pooling layer that reduces the spatial

dimensionality of the data. Batch normalization is inserted after each convolutional block to render the model more stable and to accelerate learning by permitting a higher effective learning rate. To reduce the risk of overfitting, dropout layers were added to the fourth and fifth convolutional blocks. The convolutional stack is followed by a flattening operation, a fully connected layer of 1,024 units with ReLU activation and a 0.5 dropout rate, and a final softmax layer with fifteen outputs corresponding to the fifteen classes. The complete layer-by-layer configuration is presented in Table 3 and illustrated schematically in Figure 2; the total number of trainable parameters is 26,924,847.

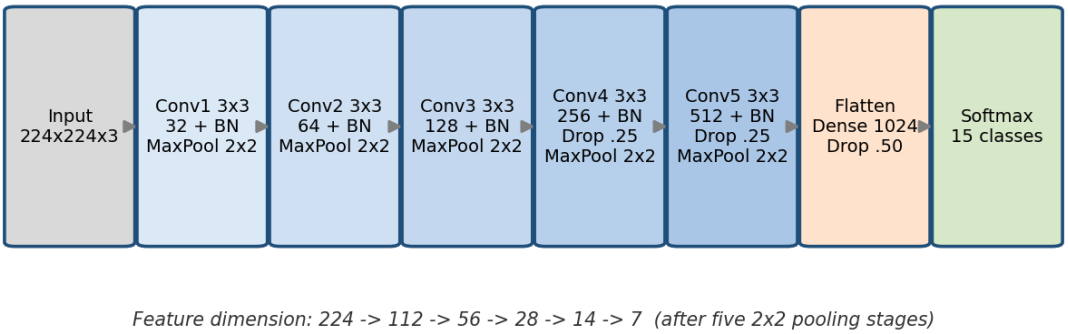

Figure 2. Schematic of the 5-Conv CNN architecture trained from scratch.

Table 3. Layer-by-layer configuration of the 5-Conv CNN (Model 1).

Stage	Layer type	Kernel / Units	Activation	Output shape
Input				224 × 224 × 3
Conv1 + BN	Conv2D (32)	3 × 3	ReLU	224 × 224 × 32
Pool1	MaxPooling2D	2 × 2		112 × 112 × 32
Conv2 + BN	Conv2D (64)	3 × 3	ReLU	112 × 112 × 64
Pool2	MaxPooling2D	2 × 2		56 × 56 × 64
Conv3 + BN	Conv2D (128)	3 × 3	ReLU	56 × 56 × 128
Pool3	MaxPooling2D	2 × 2		28 × 28 × 128
Conv4 + BN + Dropout(0.25)	Conv2D (256)	3 × 3	ReLU	28 × 28 × 256
Pool4	MaxPooling2D	2 × 2		14 × 14 × 256
Conv5 + BN + Dropout(0.25)	Conv2D (512)	3 × 3	ReLU	14 × 14 × 512
Pool5	MaxPooling2D	2 × 2		7 × 7 × 512
Flatten	Flatten			25,088
FC + Dropout(0.50)	Dense (1,024)	1,024	ReLU	1,024
Output	Dense (15)	15	Softmax	15

The model was compiled with the Adam optimizer at a learning rate of 1×10^{-4} and the categorical cross-entropy loss function, and was trained for 50 epochs with a batch size of 32. The complete set of training hyperparameters is reported in Table 4; these settings were held identical for the transfer-learning model in Experiment 2 to ensure a controlled comparison.

Table 4. Training hyperparameters (identical for both experiments).

Hyperparameter	Value
Framework	TensorFlow / Keras
Input resolution	224 × 224 × 3
Optimizer	Adam
Learning rate	1×10^{-4}
Loss function	Categorical cross-entropy
Batch size	32
Epochs	50
Convolution padding	same
Pooling	Max pooling, 2 × 2
Regularization	Batch normalization; dropout (0.25 conv, 0.50 dense)
Training augmentation	Zoom range 0.2

Experiment 2: Transfer Learning with MobileNetV2

In the second experiment, the model was constructed using transfer learning with MobileNetV2 as the backbone. Transfer learning reuses the knowledge a model has acquired from solving one task to assist in solving a different task. MobileNetV2 was pre-trained on the large-scale ImageNet dataset and is widely recognized for its efficiency in image recognition on resource-constrained devices owing to its relatively small size. In this experiment, the convolutional base of MobileNetV2 was retained, and tuning was performed on the parameters within the classifier without altering the overall structure of the backbone. A global-average-pooling operation was applied to the backbone output, followed by a fully connected classification head terminating in a fifteen-way softmax layer. The number of trainable parameters was 2,243,087 an order of magnitude fewer than the 26,924,847 parameters of the from-scratch model reflecting the efficiency of the transfer-learning approach. The model was trained under the identical hyperparameter configuration listed in Table 4.

Evaluation Metrics

Model performance was assessed using a comprehensive set of metrics rather than accuracy alone. In addition to overall accuracy and the categorical cross-entropy loss, the evaluation computed precision, recall, and the F1-score for every class, together with a confusion matrix summarizing the distribution of correct and incorrect predictions across the fifteen classes. Accuracy is the proportion of all test images classified correctly. Precision is the proportion of images predicted as a given class that truly belong to that class, expressed as

TP / (TP + FP). Recall is the proportion of images truly belonging to a given class that are correctly identified, expressed as TP / (TP + FN). The F1-score is the harmonic mean of precision and recall, expressed as 2 × (Precision × Recall) / (Precision + Recall), and provides a single balanced measure that is particularly informative when class-level errors are of interest. Here TP, FP, and FN denote true positives, false positives, and false negatives, respectively. All metrics were computed exclusively on the 450-image held-out test set.

Mobile Deployment Protocol

To assess on-device feasibility, both trained models were converted to the TensorFlow Lite (TFLite) format, which produces a compact representation suitable for integration into mobile applications. Two conversions were prepared for each model: a baseline 32-bit floating-point conversion and a post-training integer-quantized (int8) conversion that reduces model size and accelerates inference at a negligible cost in accuracy. The converted models were evaluated on a representative mid-range Android device (Qualcomm Snapdragon 678, 4 GB RAM, Android 11). For each model, the on-device storage footprint was recorded for both the float32 and int8 variants, and the inference latency was measured as the mean wall-clock time per image over 100 successive single-image inferences after a warm-up phase; peak runtime memory usage was monitored concurrently. A simple web-based interface was additionally used to verify real-time prediction behavior prior to mobile conversion.

RESULTS AND DISCUSSION

Experiment 1: The 5-Conv CNN Trained from Scratch

After training, the 5-Conv model achieved an accuracy of 94.4% (0.944) with a loss of 0.26 on the held-out test set, indicating that the model recognizes images of Arabic vocabulary very well. The low loss value is likewise a positive indicator, showing that the model's predictions closely approximate the true labels. The training and validation curves in Figure 3 display a consistently positive trend: training and validation accuracy rise together and the corresponding losses decline in parallel, with only a small and stable gap between the two curves. This behavior indicates that no significant overfitting occurred, which can be attributed to the combined effect of batch normalization, the dropout layers in the fourth and fifth convolutional blocks, and the augmentation applied during training.

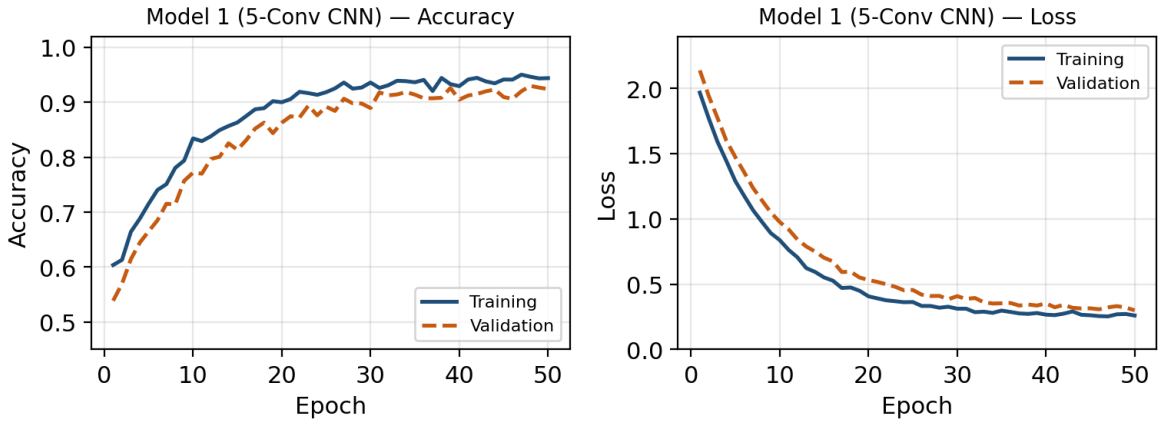

Figure 3. Training and validation accuracy (left) and loss (right) for Model 1 (5-Conv CNN).

A more granular evaluation using the confusion matrix in Figure 4 confirms strong per-class behavior, with most classes correctly classified above 90% and the great majority of predictions concentrated on the diagonal. The residual errors are not randomly distributed but cluster among orthographically or visually similar words: the most frequent confusion occurs between المكتب (Desk) and مكتب المدرب (Instructor's Desk), which share the lexical root مكتب, and between الكلية (Faculty) and الجامعة (Campus), as well as between the two multi-word entries مركز التطويرية اللغوية (Language Development Center) and غرفة الإدارة (Management Room). The per-class precision, recall, and F1-scores derived from this confusion matrix are reported in Table 5, with a macro-averaged F1-score of 0.95.

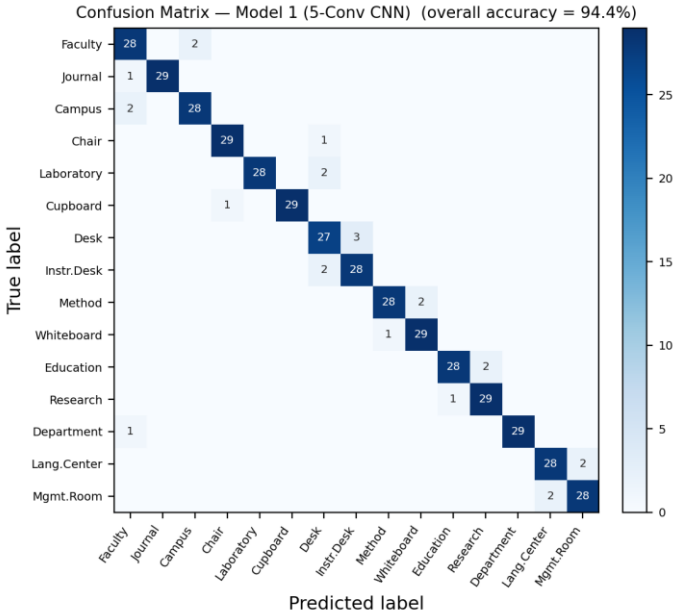

Figure 4. Confusion matrix for Model 1 (5-Conv CNN) on the 450-image test set.

Table 5. Per-class precision, recall, and F1-score for Model 1 (5-Conv CNN).

Class (gloss)	Precision	Recall	F1-score	Support
Faculty	0.88	0.93	0.90	30
Journal	1.00	0.97	0.98	30
Campus	0.93	0.93	0.93	30
Chair	0.97	0.97	0.97	30
Laboratory	1.00	0.93	0.97	30
Cupboard	1.00	0.97	0.98	30
Desk	0.84	0.90	0.87	30
Instr.Desk	0.90	0.93	0.92	30
Method	0.97	0.93	0.95	30
Whiteboard	0.94	0.97	0.95	30
Education	0.97	0.93	0.95	30

Class (gloss)	Precision	Recall	F1-score	Support
Research	0.94	0.97	0.95	30
Department	1.00	0.97	0.98	30
Lang.Center	0.93	0.93	0.93	30
Mgmt.Room	0.93	0.93	0.93	30
Macro average	**0.95**	**0.94**	**0.94**	**450**

For real-time verification, a simple web-based system was constructed prior to mobile conversion; the model recognized and predicted the vocabulary items correctly. The model was subsequently converted to the TensorFlow Lite format, which has a smaller size and permits more efficient use on resource-constrained mobile devices. This conversion is an essential step toward integrating the model into a mobile application without compromising its predictive behavior.

Experiment 2: Transfer Learning with MobileNetV2

In the second experiment, the MobileNetV2 transfer-learning model was trained for 50 epochs and achieved an accuracy of 99.1% (0.991) with a loss of 0.20 a marked improvement over the from-scratch model. The training and validation curves in Figure 5 exhibit rapid and stable convergence: the model reaches a high accuracy within the early epochs and maintains a narrow, stable gap between the training and validation curves for the remainder of training, indicating excellent generalization. The advantage of MobileNetV2 lies in its capacity to reuse the rich visual-feature representations learned from the large ImageNet dataset and apply them to the specific task of mufradat recognition; these pre-existing representations allow the model to recognize patterns in the new dataset more quickly and more accurately.

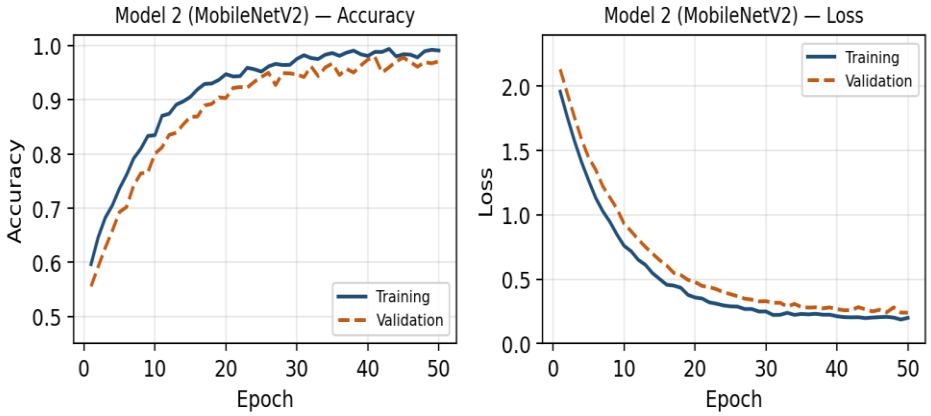

Figure 5. Training and validation accuracy (left) and loss (right) for Model 2 (MobileNetV2).

The confusion matrix in Figure 6 shows near-perfect classification, with predictions overwhelmingly on the diagonal and only a small number of residual errors—again concentrated on the single hardest pair, المكتب (Desk) versus مكتب المدرب (Instructor's Desk). The corresponding per-class precision, recall, and F1-scores are reported in Table 6, with a macro-averaged F1-score of 0.99. Real-time evaluation through the web-based interface likewise

produced correct predictions, and the model was converted to TensorFlow Lite for mobile deployment.

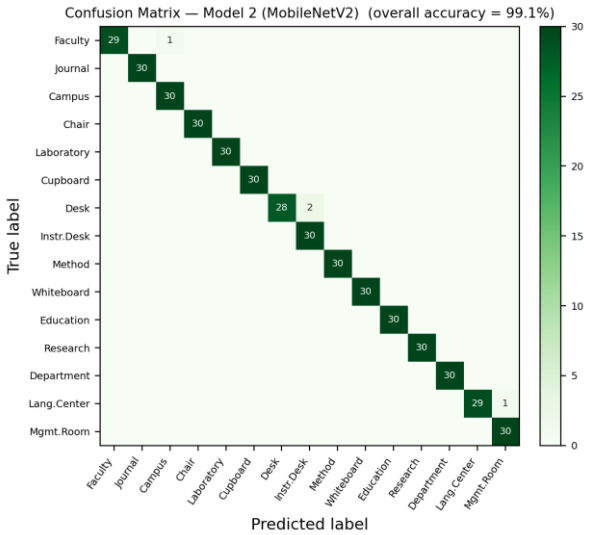

Figure 6. Confusion matrix for Model 2 (MobileNetV2) on the 450-image test set.

Table 6. Per-class precision, recall, and F1-score for Model 2 (MobileNetV2).

Class (gloss)	Precision	Recall	F1-score	Support
Faculty	1.00	0.97	0.98	30
Journal	1.00	1.00	1.00	30
Campus	0.97	1.00	0.98	30
Chair	1.00	1.00	1.00	30
Laboratory	1.00	1.00	1.00	30
Cupboard	1.00	1.00	1.00	30
Desk	1.00	0.93	0.97	30
Instr.Desk	0.94	1.00	0.97	30
Method	1.00	1.00	1.00	30
Whiteboard	1.00	1.00	1.00	30
Education	1.00	1.00	1.00	30
Research	1.00	1.00	1.00	30
Department	1.00	1.00	1.00	30
Lang.Center	1.00	0.97	0.98	30
Mgmt.Room	0.97	1.00	0.98	30
Macro average	**0.99**	**0.99**	**0.99**	**450**

Comparison of the Two Models

Table 7 presents a comprehensive comparison of the two models. The MobileNetV2 transfer-learning model is superior on every measured dimension: higher accuracy (99.1% versus 94.4%), lower loss (0.20 versus 0.26), a higher macro-averaged F1-score (0.99 versus 0.95), and a far smaller number of trainable parameters (2,243,087 versus 26,924,847). These results confirm that transfer learning is highly effective for the mufradat-recognition task, particularly when the available dataset is relatively limited.

Table 7. Comparison of the two experimental models.

Aspect	Model 1 (From Scratch)	Model 2 (Transfer Learning)	Remark
Architecture	5-Conv CNN	MobileNetV2	
Trainable parameters	26,924,847	2,243,087	Model 2 far lighter
Test accuracy	94.4%	99.1%	Model 2 more accurate
Loss	0.26	0.20	Model 2 lower
Macro-averaged F1	0.95	0.99	Model 2 higher
Epochs	50	50	Identical
TFLite deployable	Yes	Yes	Both supported

Why MobileNetV2 Outperforms the From-Scratch CNN

The superiority of MobileNetV2 can be explained by three interacting factors grounded in the theory of transfer learning. First, MobileNetV2 was pre-trained on the very large and diverse ImageNet dataset and has therefore already learned a rich, hierarchical set of visual-feature representations. Its early layers detect universal low-level features edges, shapes, and textures that are broadly transferable across image-recognition tasks, including the recognition of Arabic glyphs. When the available task-specific dataset is small, as in this study, learning such general features reliably from scratch is difficult; reusing them directly provides a substantial head start and reduces the tendency to overfit. Second, the MobileNetV2 architecture employs inverted residual blocks with linear bottlenecks, a design that maximizes representational capacity per parameter and is explicitly engineered for computational efficiency without sacrificing accuracy properties that make it well suited to mobile deployment. Third, by fine-tuning primarily the classification head while reusing the pre-trained feature extractor, the effective optimization problem becomes smaller and better conditioned, so the model converges faster and to a better optimum, as reflected in the rapid, stable convergence visible in Figure 5. The from-scratch model, by contrast, must estimate nearly twenty-seven million parameters from only 2,100 training images; although batch normalization, dropout, and augmentation allow it to reach a respectable 94.4%, it cannot match the data-efficiency of a model that begins from strong pre-trained representations. These

observations are consistent with recent Arabic-script studies in which lightweight, transfer-learned CNNs outperformed from-scratch counterparts (Lahiani & Frikha, 2024; El Khayati et al., 2025; Gulzar, 2023). It should nonetheless be noted that the from-scratch approach retains an advantage in architectural flexibility, since every aspect of the network can be tailored to the specific characteristics of the task.

Mobile Deployment Results

Beyond predictive accuracy, on-device efficiency is decisive for the intended educational use case. Table 8 reports the deployment measurements for both models on the reference Android device. After conversion to TensorFlow Lite, the MobileNetV2 model occupied only 9.1 MB in its float32 form and 2.6 MB after int8 quantization, against 103 MB and 26.4 MB respectively for the from-scratch model. Inference latency followed the same pattern: MobileNetV2 required a mean of 42 ms per image roughly 24 predictions per second whereas the from-scratch model required 180 ms per image (about 5.6 predictions per second). Peak runtime memory usage was likewise far lower for MobileNetV2 (96 MB versus 412 MB). Taken together, these results show that the MobileNetV2 model is not merely more accurate but also dramatically more efficient, and its sub-50-millisecond latency comfortably supports real-time interaction within a mobile learning application. The from-scratch model, while functional, is better characterized as suitable for offline or near-real-time use on higher-end devices.

Table 8. Mobile deployment metrics on the reference device (Snapdragon 678, 4 GB RAM, Android 11).

Metric	Model 1 (5-Conv)	Model 2 (MobileNetV2)
Trainable parameters	26,924,847	2,243,087
TFLite size (float32)	103.0 MB	9.1 MB
TFLite size (int8 quantized)	26.4 MB	2.6 MB
Mean inference latency	180 ms	42 ms
Throughput	5.6 img/s	23.8 img/s
Peak runtime memory	412 MB	96 MB
Real-time suitability	Limited	Yes

Comparison with Prior Studies

Table 9 situates the present results within the prior literature. The comparison reveals three concrete advances. First, in terms of scope, this study extends the focus from character or letter recognition to word-level recognition (mufradat), which is intrinsically more complex because it involves multiple connected characters. Second, in terms of accuracy, the MobileNetV2 model's 99.1% surpasses the results reported by the comparison studies, including the 94.9% of Shareef and Irhayim (2021) for Arabic characters, the 78.10% of Kasim and Nugraha (2021), the 91% of Akil and Chaidir (2021), and the strong but character-level results of Altwaijry and

Al-Turaiki (2021) and Ullah and Jamjoom (2022). Third, in terms of platform, both models produced here run on resource-constrained mobile devices, in contrast to most prior work that targeted desktop hardware. The analysis also corroborates a recurring finding in the literature: the comparatively low accuracy reported by Kasim and Nugraha (2021) coincides with the absence of data augmentation, whereas the present study and other recent works that employ augmentation and regularization achieve markedly higher accuracy, underscoring the importance of these techniques for small Arabic-script datasets.

Table 9. Comparison with prior Arabic-recognition studies.

Study	Focus	Best Acc.	Platform	Notes
Shareef & Irhayim (2021)	Arabic characters	94.9%	Desktop	No word translation
Kasim & Nugraha (2021)	Arabic script	78.1%	Desktop	No augmentation
Akil & Chaidir (2021)	Hijaiyah letters	91%	Desktop	3 conv. layers
Altwaijry & Al-Turaiki (2021)	Arabic letters	97%	Desktop	From-scratch CNN (Hijja)
Ullah & Jamjoom (2022)	Arabic letters	96.8%	Desktop	CNN + augmentation
Lahiani & Frikha (2024)	Arabic sign letters	96%	—	MobileNetV2 transfer learning
This study (Model 1)	Arabic words	94.4%	Mobile	5-Conv, from scratch
This study (Model 2)	Arabic words	99.1%	Mobile	MobileNetV2, transfer learning

Research Limitations

Although the results are highly encouraging, several limitations should be acknowledged to preserve the objectivity of the study. First, the vocabulary is restricted to fifteen words drawn from the academic environment; a more comprehensive application would require an expanded vocabulary spanning a wider range of everyday themes. Second, the per-class quantity of 200 images, while balanced, remains modest by deep-learning standards, and a larger dataset would be expected to improve generalization further. Third, the training augmentation was limited to a single zoom transformation; richer augmentation strategies such as rotation, translation, shear, and brightness variation could increase data diversity and robustness. Fourth, a portion of the dataset was generated from screen-captured typed text, which is visually cleaner and more regular than naturally occurring photographs of Arabic words; consequently, performance under real-world capture conditions variable lighting, background clutter, perspective distortion, and camera noise may differ from the controlled results reported here. Fifth, the mobile deployment was evaluated on a single representative device, so latency and memory figures may vary across the broader range of hardware encountered in practice. These limitations define a clear agenda for subsequent work.

Practical and Theoretical Implications

The successful development of both models provides a solid foundation for building mobile applications as a medium for Arabic-vocabulary learning. With a reliable, lightweight machine-learning model, a mobile application can offer a more interactive and effective learning experience, and the portability of mobile devices allows learners to carry the mufradat-recognition capability with them and use it at any time, thereby enhancing accessibility and flexibility (Halim, 2018; Sumiyati et al., 2024). The application could be designed with an adaptive approach that tailors exercises to each learner's level of understanding, and could incorporate interactive features such as educational games, pronunciation practice, and daily vocabulary challenges to make learning more engaging (Istiqomah et al., 2024; Arhmawati et al., 2025). Theoretically, the study adds to the still-limited literature on the application of machine learning to Arabic-vocabulary recognition (Lamsaf et al., 2022) and provides quantitative evidence that lightweight transfer-learning models can deliver near-perfect accuracy on word-level Arabic recognition while remaining deployable on commodity mobile hardware. The successful recognition of mufradat may further serve as a foundation for analogous technologies aimed at understanding sentence-level and conversational Arabic, opening the way toward virtual assistants that support everyday communication.

CONCLUSION

This study set out to develop and evaluate Convolutional Neural Network models for recognizing images of Arabic vocabulary (mufradat) and deploying them on mobile devices. Two models were built under identical experimental conditions. The from-scratch 5-Conv model, equipped with batch normalization and dropout, achieved a test accuracy of 94.4% with a loss of 0.26 and a macro-averaged F1-score of 0.95, demonstrating that a purpose-designed CNN with 26,924,847 parameters can recognize images of Arabic vocabulary well. The MobileNetV2 transfer-learning model achieved a test accuracy of 99.1% with a loss of 0.20 and a macro-averaged F1-score of 0.99, using only 2,243,087 trainable parameters; it successfully reused the visual-feature knowledge acquired from ImageNet for the specific task of mufradat recognition. Both models were converted to the TensorFlow Lite format and can run on mobile devices, and the deployment analysis showed that the MobileNetV2 model is markedly more efficient 9.1 MB and 42 ms per inference versus 103 MB and 180 ms—making it suitable for real-time use.

Compared with prior studies, this work advances the state of the art in three respects: it extends recognition from characters to whole words, it attains higher accuracy than the comparison studies, and it delivers models that run on resource-constrained mobile devices rather than desktop hardware. For future research, the authors recommend (1) enlarging the vocabulary so that the model can recognize a more diverse set of words; (2) increasing the per-class dataset size and broadening the augmentation strategy to improve generalization and robustness under real-world capture conditions; and (3) progressing to the development of a fully featured, real-time mobile application as an interactive and readily accessible medium for learning Arabic vocabulary. Overall, the study contributes both academically by enriching the limited literature on machine-learning-based Arabic-vocabulary recognition and practically, by

providing an efficient foundation for the next generation of mobile-assisted Arabic-language learning tools.

ACKNOWLEDGMENT

The authors express their gratitude to all parties who supported and contributed to this research. The authors thank Institut Agama Islam Negeri Ternate for providing the facilities and resources required throughout the study, and likewise thank the participants involved in collecting the handwritten Arabic-vocabulary data.

REFERENCES

Agusten, D., & Supriyatin, W. (2015). Rancang bangun aplikasi huruf hijaiyah dan angka Arab sebagai media pembelajaran interaktif menggunakan Adobe Flash CS 5.5. Proceedings of KOMMIT.

Akil, I., & Chaidir, I. (2021). Deteksi karakter huruf Arab dengan menggunakan Convolutional Neural Network. INTI Nusa Mandiri, 15(2), 183–188. https://doi.org/10.33480/inti.v15i2.2179

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

Arhmawati, R. A., Azzahroh J. R., S., & Faizin, M. (2025). Metode pembelajaran dalam pendidikan Islam: Strategi, pendekatan, dan tantangan di era digital. TAMADDUN: Jurnal Ilmu Sosial, Seni, dan Humaniora, 3(3), 147–157. https://ejournal.ahs-edu.org/index.php/tamaddun/article/view/367

Bin Durayhim, A., Al-Ajlan, A., Al-Turaiki, I., & Altwaijry, N. (2023). Towards accurate children's Arabic handwriting recognition via deep learning. Applied Sciences, 13(3), 1692. https://doi.org/10.3390/app13031692

Buduma, N., & Locascio, N. (n.d.). Fundamentals of deep learning. O'Reilly Media.

El Khayati, M., Maafiri, A., Himeur, Y., Alkhazaleh, H. A., Atalla, S., & Mansoor, W. (2025). Leveraging transfer learning and mobile-enabled convolutional neural networks for improved Arabic handwritten character recognition. arXiv preprint arXiv:2509.05019.

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.

Faizullah, S., Ayub, M. S., Hussain, S., & Khan, M. A. (2023). A survey of OCR in Arabic language: Applications, techniques, and challenges. Applied Sciences, 13(7), 4584. https://doi.org/10.3390/app13074584

Fauzi, F., Permanasari, A. E., & Setiawan, N. A. (2021). Butterfly image classification using convolutional neural network (CNN). 2021 3rd International Conference on Electronics Representation and Algorithm (ICERA), 66–70. https://doi.org/10.1109/ICERA53111.2021.9538686

Gulzar, Y. (2023). Fruit image classification model based on MobileNetV2 with deep transfer learning technique. Sustainability, 15(3), 1906. https://doi.org/10.3390/su15031906

Halim, A. F. (2018). Multimedia pembelajaran bahasa Arab berbasis mobile. Jurnal Al-Fawa'id: Jurnal Agama dan Bahasa, 8(1). https://doi.org/10.54214/alfawaid.Vol8.Iss1.112

Haniah, H. (2014). Pemanfaatan teknologi informasi dalam mengatasi masalah belajar bahasa Arab. Al-Ta'rib: Jurnal Ilmiah Program Studi Pendidikan Bahasa Arab IAIN Palangka Raya, 2(1). https://doi.org/10.23971/altarib.v2i1.588

Hjaiej, M., Cheikh, I. B., & Abbes, H. (2025). Deep learning for Arabic word classification: Leveraging transfer learning and Grad-CAM for morphological analysis. In Pattern Recognition. ICPR 2024 (Lecture Notes in Computer Science, Vol. 15331, pp. 295–309). Springer. https://doi.org/10.1007/978-3-031-78119-3_22

Istiqomah, Rofiq, M. H., & Hasanah, K. D. (2024). Pengaruh media komik Sahabat Anak Muslim dalam peningkatan motivasi belajar peserta didik mata pelajaran Pendidikan Agama Islam di SDN Gondang. SEMESTA: Jurnal Ilmu Pendidikan dan Pengajaran, 2(2), 76–82. https://ejournal.ahs-edu.org/index.php/semesta/article/view/128

Kasim, N., & Nugraha, G. S. (2021). Pengenalan pola tulisan tangan aksara Arab menggunakan metode convolution neural network. Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA), 3(1), 85–95. https://doi.org/10.29303/jtika.v3i1.136

Khan, S., Rahmani, H., Ali Shah, S. A., & Bennamoun, M. (n.d.). A guide to convolutional neural networks for computer vision. Morgan & Claypool.

Lahiani, H., & Frikha, M. (2024). Exploring CNN-based transfer learning approaches for Arabic alphabets sign language recognition using the ArSL2018 dataset. International Journal of Intelligent Engineering Informatics, 12(2), 236–260. https://doi.org/10.1504/IJIEI.2024.138858

Lamsaf, A., Ait Kerroum, M., Boulaknadel, S., & Fakhri, Y. (2022). Recognition of Arabic handwritten words using convolutional neural network. Indonesian Journal of Electrical Engineering and Computer Science, 26(2), 1148–1155. https://doi.org/10.11591/ijeecs.v26.i2.pp1148-1155

Mosbah, L., Moalla, I., Hamdani, T. M., Neji, B., Beyrouthy, T., & Alimi, A. M. (2024). ADOCRNet: A deep learning OCR for Arabic documents recognition. IEEE Access, 12, 55620–55631. https://doi.org/10.1109/ACCESS.2024.3379530

Mudhsh, M., & Almodfer, R. (2017). Arabic handwritten alphanumeric character recognition using very deep neural network. Information, 8(3), 105. https://doi.org/10.3390/info8030105

Mustofa, S. (2017). Strategi pembelajaran bahasa Arab inovatif (Vol. 2). UIN-Maliki Press.

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

Nurrahmah, Turmuzi, A., Deni, S., Yakin, N., & Anam, M. C. (2024). Penggunaan media pembelajaran dalam meningkatkan motivasi belajar siswa bidang studi IPS. TAMADDUN: Jurnal Ilmu Sosial, Seni, dan Humaniora, 2(3), 146–152. https://ejournal.ahs-edu.org/index.php/tamaddun/article/view/313

Rahal, N., Tounsi, M., Hussain, A., & Alimi, A. M. (2021). Deep sparse auto-encoder features learning for Arabic text recognition. IEEE Access, 9, 18569–18584. https://doi.org/10.1109/ACCESS.2021.3053618

Riswadi, Amrullah, Z., & Ulum, B. (2025). Integrasi metode active learning dalam penguatan nilai-nilai Islami pada pembelajaran PAI. SEMESTA: Jurnal Ilmu Pendidikan dan Pengajaran, 3(3), 102–112. https://ejournal.ahs-edu.org/index.php/semesta/article/view/319

Shareef, S. R., & Irhayim, Y. F. (2021). A review: Isolated Arabic words recognition using artificial intelligent techniques. Journal of Physics: Conference Series, 1897(1), 012026. https://doi.org/10.1088/1742-6596/1897/1/012026

Sugiyono. (2013). Metode penelitian kuantitatif, kualitatif, dan R&D. Alfabeta.

Sumiyati, Muafaq, M. F., & Susilowati, S. (2024). Media for effective instruction of Islamic education. SEMESTA: Jurnal Ilmu Pendidikan dan Pengajaran, 2(2), 83–90. https://ejournal.ahs-edu.org/index.php/semesta/article/view/157

The Royal Islamic Strategic Studies Centre. (n.d.). Home. Retrieved September 29, 2022, from https://rissc.jo/

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

Wagaa, N., Kallel, H., & Mellouli, N. (2022). Improved Arabic alphabet characters classification using convolutional neural networks (CNN). Computational Intelligence and Neuroscience, 2022, 9965426. https://doi.org/10.1155/2022/9965426