Jurnal Komputer dan Elektro Sains, Vol.4 No.1, March 2026, 33-40
https://ejournal.sultanpublisher.com/index.php/komets

Received : March 29, 2026; Revision: June 21, 2026; Accepted: June 21, 2026; Publication: June 22, 2026.

E-ISSN: 3021-8462, DOI: https://doi.org/10.58291/komets.v4i1.581

Lightweight CNN and Grad-CAM Integration for Interpreting Arabic Sign Language Hijaiyyah Letter Classification

Akbar Sidqi[1], Ana Rizkiya[2]
[1]Department of Computer Science, Al-Rifa'ie Modern University Indonesia, Malang, Indonesia
[2]Special Program for Arabic Language Development, Maulana Malik Ibrahim State Islamic University, Malang, Indonesia

Corresponding Author: Akbar Sidqi (e-mail: akbarsidqi@gmail.com)

ABSTRACT

Communication plays an important role in social life as a means of conveying information and building interactions between humans. For individuals who are deaf and mute, sign language serves as a visual communication medium that supports the process of conveying messages. One of the sign language systems is Arabic Sign Language (ArSL), which utilises the Hijaiyyah letters through the fingerspelling method to spell out certain terms or vocabulary. However, the limited understanding of sign language within society still poses a barrier to social interaction. This research develops an Arabic Sign Language (ArSL) Hijaiyyah letter classification system based on a Lightweight Convolutional Neural Network (CNN) with the aim of producing a model that has high performance and low computational requirements. Additionally, the Explainable Artificial Intelligence (XAI) method using Gradient-weighted Class Activation Mapping (Grad-CAM) is applied to enhance the transparency of classification results through the visualisation of important areas in the images. Based on the test results, the developed model achieved an accuracy of 0.84, a precision of 0.84, a recall of 0.84, an F1-score of 0.84 and a an Specificity of 0.99. The Grad-CAM visualisation results show that the model focuses on the relevant hand areas during the prediction process. The findings of this study indicate that the system is capable of performing classification with good and measurable performance, as well as supporting real-time sign language recognition with a short response time, thereby improving communication accessibility for individuals with disabilities.

KEYWORD Arabic Sign Language; Hijaiyyah letters; CNN; Grad-Cam; Sign Language Recognition.

1. INTRODUCTION

Humans need communication as an important part of social life to support interaction and the exchange of information. Through communication, individuals can convey messages, build social relationships, and create understanding in various daily activities [1, 2]. The deaf and mute community uses sign language as a visual communication medium to convey information, express thoughts, and meet communication needs. Sign language also helps improve interaction effectiveness, strengthen social relationships, and reduce communication barriers, thereby supporting the creation of a more inclusive environment [3, 4]. Data from the World Health Organization (WHO) shows that around 430–466 million people worldwide experience hearing impairment [4, 5]. This number is expected to continue increasing due to population growth, demographic changes, and other health factors. This condition demands the development of communication access, increased inclusivity, and the provision of supportive technology for individuals with hearing impairments.

Arabic Sign Language (ArSL) plays an important role in helping the deaf community communicate and build more effective social interactions. The use of ArSL can enhance communication accessibility, expand social participation, and reduce various interaction barriers, thereby contributing to the creation of a more inclusive environment [5, 6]. In its application, one of the basic components used in ArSL is the Hijaiyyah letters, which serve as the foundation of communication. The Hijaiyyah letters are utilized through the fingerspelling method to spell out names, terms, or specific vocabulary that do not yet have a special sign representation [1, 7]. Through this method, ArSL users can convey information more specifically, clearly, and accurately.

However, communication still faces obstacles because the majority of the community does not yet understand sign language [1, 3]. These limitations create communication gaps and hinder interactions in various social environments. To address this issue, various studies have developed computer-based Sign Language Recognition (SLR) systems capable of automatically recognising and interpreting sign language gestures [3, 8]. This system helps bridge communication between sign language users and the general public without fully relying on human interpreters, whose numbers are limited and whose service costs are relatively high.

The development of Artificial Intelligence (AI) and

Computer Vision technologies has brought significant changes to Sign Language Recognition (SLR) systems [3, 8]. Vision-based approaches (using cameras) are now more preferred than wearable sensor devices because they are more practical, cost-effective, and comfortable for users [3, 4, 9]. In this field, Convolutional Neural Networks (CNNs) have become the dominant paradigm due to their ability to automatically extract strong spatial features from hand gesture images [3, 4]. Advanced architectures such as EfficientNet-B7 and ResNet101 have demonstrated strong performance, achieving accuracies exceeding 99% on Arabic Sign Language Alphabet (ArSLA) datasets [1, 5]. The main strength of these models lies in their powerful and hierarchical spatial feature extraction capabilities. However, despite their high accuracy, conventional CNN models often require substantial computational resources and high memory consumption, making them difficult to deploy on mobile devices or edge devices with limited resources [3, 4, 10].

Therefore, the use of lightweight CNN architectures becomes a relevant solution [6, 7]. Lightweight models are designed to minimise the number of parameters and computational costs through techniques such as depthwise separable convolutions while still maintaining competitive discriminative performance for real-time applications [3, 6, 7]. In addition to efficiency, another challenge in using deep learning is the nature of the model, which is often considered a black-box, where classification decisions are difficult for humans to understand [5, 7]. In sensitive contexts such as education or healthcare, understanding the reasons behind the model's decisions becomes crucial for building trust and system reliability [6, 7].

On the other hand, interpretability has become an important focus in SLR research through the integration of Explainable AI (XAI). Techniques such as SHAP (Shapley Additive exPlanations) have been used to visualize important features; however, they have limitations related to the assumption of feature independence, which does not always hold for spatial image data [3]. To address these limitations, XAI approaches such as Grad-CAM (Gradient-weighted Class Activation Mapping) have been integrated into the model interpretation process [6,8]. Grad-CAM produces heatmap visualizations that highlight the regions of hand images that contribute most to the model's prediction decisions [6].

Based on an analysis of recent studies, this research proposes the integration of a Lightweight CNN and Grad-CAM for interpreting the classification results of Hijaiyyah letters in Arabic Sign Language, in order to address both computational limitations and interpretability challenges. Unlike previous approaches that primarily focus on improving performance through complex model architectures, this study emphasizes the use of a lightweight CNN architecture to support inference in latency-constrained scenarios. In addition, Grad-CAM is not only used as a visualization mechanism but also as an interpretability tool to evaluate the reliability of the model in representing the visual characteristics of Hijaiyyah letters. Through evaluation on the ArSLA dataset, which contains diverse visual variations, this study produces a classification system with reliable performance while enhancing transparency and trustworthiness in supporting communication for individuals with disabilities.

2. RESEARCH *METHODS*

This section explains the architecture of the Lightweight Convolutional Neural Network (Lightweight CNN) used to classify the Hijaiyyah letters of the Arabic Sign Language. In addition, this section describes the dataset used along with the data preprocessing steps applied before the model training process. This research also integrates the Gradient-weighted Class Activation Mapping (Grad-CAM) method to interpret classification results through the visualisation of image areas that contribute to the model's decision. The series of stages is visualized in Figure 1.

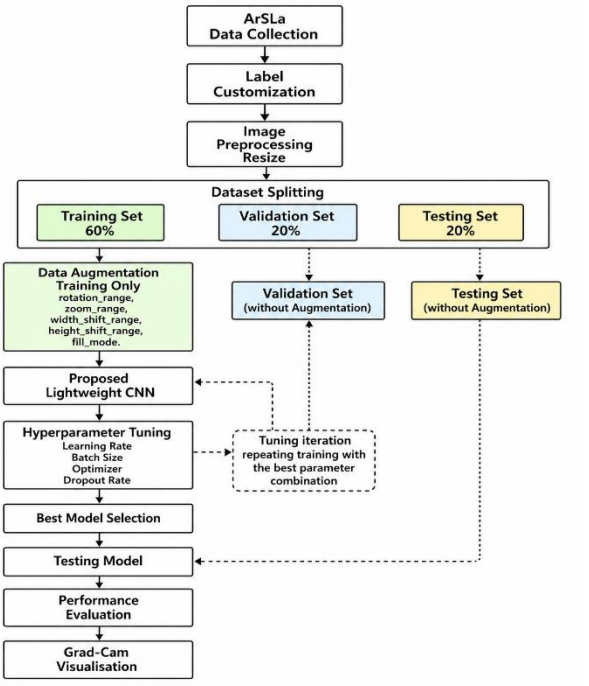

Fig 1. Research Methodology

2.1. DATASET COLLECTION

This research uses the RGB Sign Language dataset [11] developed by Muhammad Albrham (2023) [12]. The dataset was obtained through the Kaggle platform and consists of 7,856 images representing 31 Hijaiyyah letters. Each letter class displays variations in hand shapes from several individuals, thereby increasing the diversity of the data used in the model's training and testing processes. Figure 2 displays the dataset image as a representation of the data used.

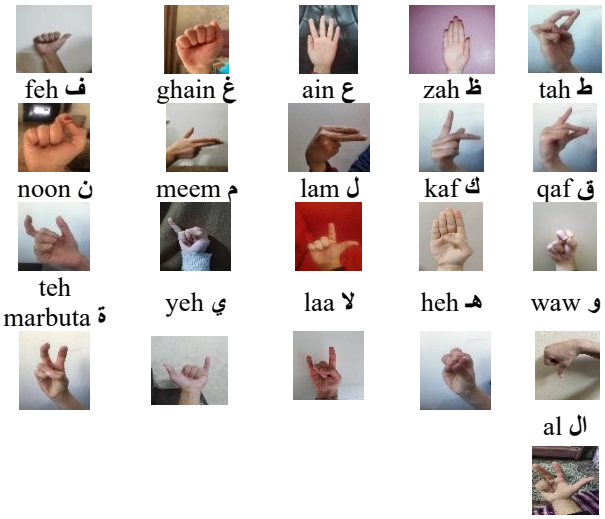

Fig 2. Image of Hijaiyah Letters

2.2. HIJAIYYAH LETTER CUSTOMISATION

This research adjusts labels to standardise dataset naming into a common Hijaiyyah letter format, as shown in Table 1. This process aims to make the research results easier to understand and improve the consistency of data naming during the processing and analysis stages.

Table 1. Hijaiyyah Letter Customisation

Original Label	New Label	Huruf Hijaiyyah
alef	alif	ا
beh	ba	ب
teh	ta	ت
theh	tsa	ث
jeem	jim	ج
hah	ha	ح
khah	kha	خ
dal	dal	د
thal	dzal	ذ
reh	ra	ر
zain	zay	ز
seen	sin	س
sheen	syin	ش
sad	shad	ص
dad	dhad	ض
tah	tha	ط
zah	zha	ظ
ain	ain	ع
ghain	ghain	غ
feh	fa	ف
qaf	qaf	ق
kaf	kaf	ك
lam	lam	ل
meem	mim	م
noon	nun	ن
waw	wau	و
heh	ha	ه
laa	lam alif	لا
yeh	ya	ي
teh marbuta	ta marbuta	ة
al	alif lam	ال

2.3. SPLIT DATASET AND AUGMENTATION

Dataset Splitting [13] and data augmentation [14] are essential stages in the development of a sign language recognition model, as they enhance the model's learning capability and generalization performance. In this study, the dataset was divided using a 60:20:20 ratio, consisting of 60% training data, 20% validation data, and 20% testing data. This ratio was selected to provide sufficient data for the training process while enabling objective validation and performance evaluation of the model. Furthermore, the distribution of data across the 31 Hijaiyyah letter classes was relatively balanced, thereby minimizing the potential impact of class imbalance and allowing the model to learn the characteristics of each class proportionally.

To increase the diversity of the training data, data augmentation was applied using 5° rotation, 5% zoom, 5% horizontal shift, and 5% vertical shift. The nearest-neighbor interpolation method was employed to fill the empty regions generated by these image transformations. The implementation of these augmentation techniques produced a more diverse training dataset, thereby improving the model's generalization capability and reducing the risk of overfitting.

2.4. PROPOSED LIGHTWEIGHT CNN

This study proposes a Lightweight Convolutional Neural Network (CNN) architecture for the classification of Hijaiyyah letters in Arabic Sign Language (ArSL). The architecture is designed to progressively extract visual features from hand gesture images while maintaining relatively low computational complexity. Furthermore, the model design emphasizes the ability to learn spatial characteristics that distinguish each Hijaiyyah letter, thereby enabling the extraction of more discriminative feature representations for the classification process. The proposed Lightweight CNN architecture employed in this study is illustrated in Figure 3.

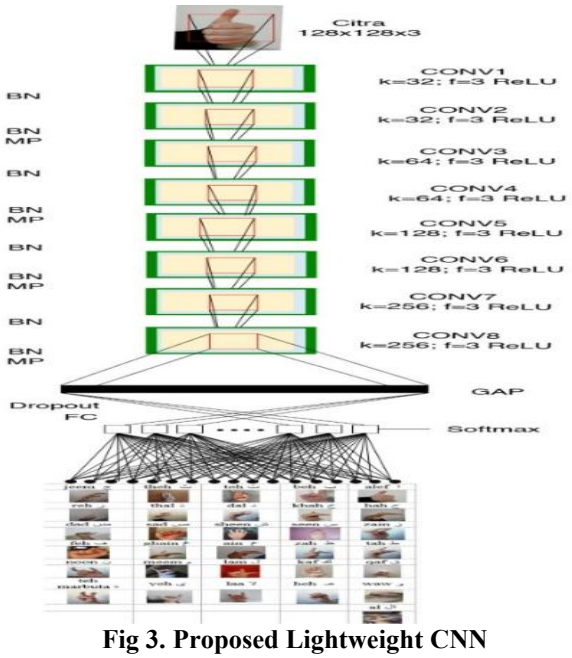

Fig 3. Proposed Lightweight CNN

In this study, all images were resized to 128 × 128 pixels to ensure a uniform input dimension [15, 16]. After the preprocessing stage, the images were fed into the proposed lightweight CNN model. The proposed architecture consists of eight convolutional layers with the number of filters gradually increasing from 32, 64, 128, and 256 filters. This progressive increase in the number of filters enables the model to learn hierarchical feature representations, ranging from simple visual patterns such as edges and hand contours to more complex patterns that represent the shapes of Hijaiyyah letter gestures. Each convolutional layer employs a 3 × 3 kernel and the ReLU activation function, as this configuration effectively extracts spatial features while maintaining a relatively low number of parameters. Batch normalization is applied to maintain the stability of feature distributions during training, while Max Pooling is used to reduce feature dimensions and preserve the most significant characteristics of hand gestures. Following the feature extraction process, Global Average Pooling (GAP) is utilized to reduce the number of parameters and model complexity compared to larger fully connected layers. In addition, a dropout layer is incorporated to mitigate the risk of overfitting caused by similarities among the patterns of different Hijaiyyah letter classes. The model was trained using the Adam optimizer with a learning rate of 0.0001, a batch size of 32, and 50 training epochs. These hyperparameter settings were selected to maintain optimization stability while enabling the model to achieve effective convergence during training.

a. Convolutional Layer

Each Hijaiyyah letter image entering the neuron layer undergoes a convolution process using various filters to extract important characteristics from the image. This convolution process generates feature maps that represent the specific patterns and distinctive characteristics of the Hijaiyyah letter images. Mathematically, the convolution process can be expressed as follows.

$$Z_i = f(W_i X + b_i), i = 1,2,3,4,5,6,7,8, \quad (1)$$

b. Activation Layer

The activation layer plays a role in enhancing the nonlinear properties of the model's decision function. This layer operates by applying an activation function, enabling the model to learn more complex patterns during the learning process. In this study, the Rectified Linear Unit (ReLU) activation function was applied in each convolution process. Mathematically, the ReLU function can be expressed as follows.

$$\tilde{Z}(Z_i) = \begin{cases} Z_i Jika Z_i \geq 0 \\ 0 Jika Z_i < 0 \end{cases}, i = 1,2,3,4,5,6,7,8 \quad (2)$$

c. Batch Normalization

Batch normalization is used to normalize the output of each mini batch to ensure a more stable training process. In Hijaiyyah letter classification, this method is applied after the convolution process to maintain the consistency of the extracted feature distribution. Batch normalization helps the model learn letter patterns more effectively and reduces shifts in data distribution during training. It can be mathematically expressed as follows, where y_i represents the normalized output, μ_B denotes the mini batch mean, σ_B^2 represents the variance, γ is the scaling parameter, β is the shifting parameter, and $\in$ is a small constant used to prevent division by zero.

d. Pooling Layer

The pooling layer is used to reduce the spatial dimensions of the convolution output, thereby decreasing computational complexity and reducing the risk of overfitting. In this study, the max-pooling method with a 2×2 patch size was employed to select the maximum value from each feature region. Mathematically, the max-pooling process can be expressed as follows.

$$g_i(\tilde{Z}_i) = Max\{\tilde{Z}_{ij}\} j, i = 1,2,3,4,5,6,7,8 \quad (3)$$

e. Global Average Pooling

In Hijaiyyah letter classification, Global Average Pooling (GAP) reduces feature dimensions by calculating the average value of each feature map. This method decreases the number of parameters, reduces overfitting, and improves the model's efficiency in recognizing Hijaiyyah letter patterns. It can be mathematically expressed as follows.

$$\text{GAP (x)}_c = \frac{1}{\text{H x W}} \sum_{1=1}^{H} \sum_{j=1}^{W} X_{i,j,c} \quad (4)$$

f. Output (Classification) Layer

Before entering the classification layer, the fully connected layer connects each neuron to the neurons in the previous layer to combine the extracted features and perform Hijaiyyah letter image classification. The number of neurons in the output layer is adjusted according to the number of classes in the training dataset. Subsequently, the softmax activation function is applied to the output layer to calculate the probability distribution across the 31 Hijaiyyah letter classes based on the output of the previous layer. Mathematically, the softmax function can be expressed as follows.

$$y_k(\hat{h}) = \frac{exp(\hat{h}_k)}{\sum_{j=1}^{4} exp(\hat{h}_j)}, k = 31 \quad (5)$$

2.5. GRAD-CAM

The Gradient-weighted Class Activation Mapping (Grad-CAM) method is used to visualize the image regions that the model focuses on during the classification of Hijaiyyah letters. This method identifies image areas that contribute to the model's prediction results. Grad-CAM utilizes gradients to determine the regions that provide the greatest contribution to the prediction of a specific class. In Hijaiyyah letter

classification, Grad-CAM uses feature maps from the last convolutional layer to highlight the image regions that most strongly influence the prediction results. Subsequently, the method calculates the weight of each feature map based on the average gradient values with respect to the predicted class. The feature map weights are computed using the following equation

$$\alpha_k^c = \frac{1}{z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (6)$$

where α_k^c represents the weight of the k-th feature map, y^c denotes the score of class c, and A_{ij}^k represents the activation value in the last convolutional layer. Subsequently, Grad-CAM generates a heatmap using the following equation.

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right) \quad (7)$$

Where A^k represents the feature map obtained from the last convolutional layer. The resulting heatmap highlights the regions of the Hijaiyyah letter that become the model's focus during the classification process.

2.6. EVALUATION

Model performance was evaluated using accuracy, precision, sensitivity (recall), specificity, and F1-score metrics to measure the classification results of Hijaiyyah letters. These metrics were used to assess the level of prediction accuracy achieved by the model for each class. Mathematically, these evaluation metrics can be expressed as follows (8–12).

$$Akurasi = \frac{(tp + tn)}{(tp + fp + tn + fn)} \quad (8)$$

$$Presisi = \frac{tp}{(tp + fp)} \quad (9)$$

$$Recall\ (Sensitivitas) = \frac{tp}{(tp + fn)} \quad (10)$$

$$Spesifitas = \frac{tn}{(tn+fp)} \quad (11)$$

$$F - Score = \frac{2x\ svt\ x\ prs}{(svt + prs)} \quad (12)$$

3. RESULTS AND DISCUSSION

This section discusses the testing results and evaluation of the Lightweight CNN model integrated with Grad-CAM for Hijaiyyah letter classification. The analysis is conducted based on experimental scenarios and evaluation parameters to assess the model's performance and examine the impact of Grad-CAM integration on the classification results.

3.1. TRAINING AND VALIDATION RESULTS

The training process was conducted using the lightweight CNN architecture. Figure 4 presents the training and validation curves to illustrate the model's performance progression across each epoch. These curves show the changes in accuracy and loss values throughout the training process, giving insight into the model's ability to learn image patterns and maintain performance on validation data. The difference between the training and validation results indicates that the model maintains consistent performance throughout the learning process.

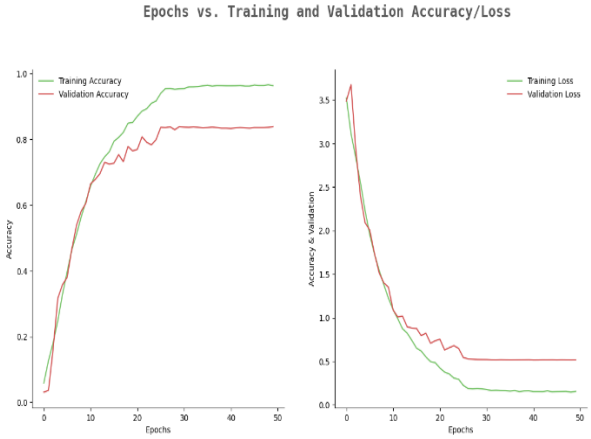

Fig 3. Lightweight CNN Training and Validation

Based on Figure 4, the training and validation accuracy curves show a consistent improvement as the number of epochs increases. The training accuracy gradually rises and reaches approximately 96%, while the validation accuracy achieves around 84%. The accuracy improvement occurs relatively rapidly during the early stages of training and begins to converge after approximately the 25th epoch. These results indicate that the model successfully learns the underlying characteristics and patterns present in the training dataset.

The training and validation loss curves also exhibit a stable decreasing trend throughout the learning process. The training loss decreases to approximately 0.15, whereas the validation loss reaches about 0.51 at the end of training. The reduction in loss accompanied by the increase in accuracy suggests that the model becomes progressively more effective at minimizing classification errors during the optimization process.

Nevertheless, a performance gap exists between the training and validation datasets, as evidenced by the approximately 12% difference between training accuracy and validation accuracy. Furthermore, after the 25th epoch, the validation accuracy curve tends to plateau, while the training accuracy continues to improve. This condition indicates a tendency toward overfitting, where the model becomes increasingly tailored to the specific characteristics of the training data, thereby limiting its ability to generalize to previously unseen data. However, both validation accuracy and validation loss remain relatively stable until the end of training without any significant degradation in performance. Therefore, the observed overfitting can be classified as mild to moderate, suggesting that the model still retains a reasonably

satisfactory generalization capability for image classification on the validation dataset.

3.2. TEST RESULTS

The testing process was conducted using test data that was not involved in the model training phase. This evaluation aims to assess the model's generalization capability on unseen images. The evaluation was performed using several metrics, including accuracy, precision, recall, specificity, and F1-score. Table 2 presents the classification results of the 31 Hijaiyyah letters in Arabic Sign Language based on the applied evaluation metrics.

Table 2. Testing Result

Class	Acc	Pre	Rec	Spe	F1-s
alif	0.99	0.82	0.92	0.99	0.87
ba	1	0.93	0.96	1	0.94
ta	0.99	0.9	0.93	1	0.91
tsa	0.99	0.83	0.89	0.99	0.86
jim	0.99	0.89	0.96	1	0.92
ha	0.99	0.82	0.79	0.99	0.8
kha	0.98	0.71	0.57	0.99	0.63
dal	0.99	0.92	0.78	1	0.84
dzal	0.98	0.69	0.88	0.99	0.77
ra	0.98	0.76	0.76	0.99	0.76
zay	0.98	0.7	0.76	0.99	0.73
sin	0.99	0.91	0.79	1	0.85
syin	0.99	0.85	0.7	1	0.77
shad	0.99	0.9	0.96	1	0.93
dhad	0.99	0.92	0.92	1	0.92
tha	0.98	0.74	0.76	0.99	0.75
zha	0.99	0.92	0.72	1	0.81
ain	0.98	0.71	0.61	0.99	0.66
ghain	0.99	0.87	0.76	1	0.81
fa	0.99	0.8	0.91	0.99	0.85
qaf	1	0.9	0.98	1	0.94
kaf	1	1	0.96	1	0.98
lam	0.99	0.83	0.84	0.99	0.83
mim	0.99	0.91	0.85	1	0.88
nun	0.99	0.81	0.82	0.99	0.81
wau	0.99	0.76	0.7	0.99	0.73
ha	0.99	0.87	0.89	0.99	0.88
lam alif	0.99	0.84	0.96	0.99	0.9
ya	0.99	0.82	0.84	0.99	0.83
ta marbuta	0.99	0.84	0.78	1	0.81
alif lam	0.99	0.73	0.82	0.99	0.77

Table 2 presents the overall classification results based on global metrics, with an accuracy of 0.84, precision of 0.84, recall of 0.84, Specificity 0.99 and F1-score of 0.84. The obtained results indicate the model's capability to consistently classify the 31 Hijaiyyah letters in Arabic Sign Language.

3.3. GRAD-CAM RESULTS

The Grad-CAM results provide a visualization of the image regions that became the model's focus during the hijaiyyah letter classification process. This visualisation helps identify the image areas that contribute to the model's decisions and can be used to analyze the model's attention patterns in recognizing the characteristics of each letter. The Grad-CAM visualisation results are presented in Figure 5.

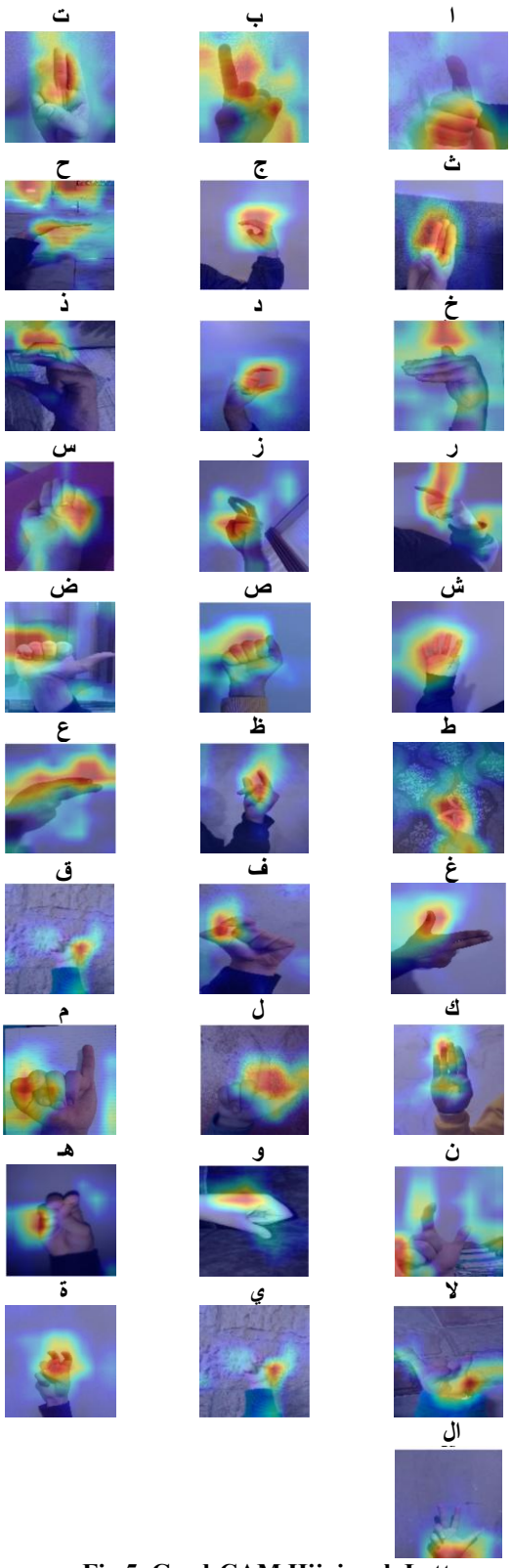

Fig 5. Grad-CAM Hijaiyyah Letters

To gain a deeper understanding of the image regions that influence the model's classification decisions, this study presents the original image,

heatmap, and Grad-CAM visualizations. These three representations provide insights into the image areas that receive the greatest attention from the model during the classification process. As an illustration, the Grad-CAM visualization of the Hijaiyah letter Ta Marbutah is shown in Figure 6.

Fig 6. Ta Marbuta Original

The original image depicts a human hand against a simple background without significant visual noise. The hand object is clearly visible, which facilitates feature extraction by the CNN model, particularly in terms of finger shape and structure. The homogeneous background also helps the model focus on the region of interest without interference from irrelevant features.

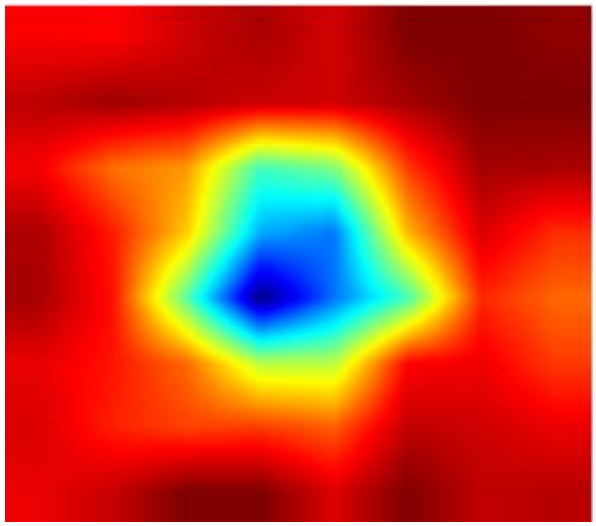

Fig 7. Ta Marbuta Heatmap

The heatmap shows the distribution of intensity values across the observation area. Blue represents low values, while red represents high values. A gradient pattern is observed from the center toward the edges, indicating a gradual increase in intensity.

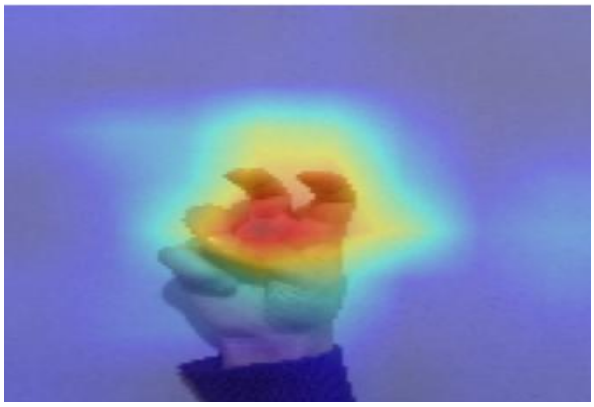

Fig 6. Ta Marbuta Grad-Cam

The Grad-CAM results show that the CNN focuses its attention on the hand region, particularly the fingers, which are highlighted in red–yellow as areas with the highest activation. This indicates that features in this region serve as the primary factors used by the model in performing classification, whereas the background areas contribute relatively little.

4. CONCLUSION

This study applies a Lightweight CNN model for the classification of 31 Arabic Hijaiyyah letters in Arabic Sign Language. The experimental results show that the model achieves an accuracy of 0.84, precision of 0.84, recall of 0.84, specificity of 0.99, and an F1-score of 0.84. These results indicate that the model has a relatively good ability to recognize and distinguish the characteristics of each Hijaiyyah letter in the test dataset. The training and validation curves show an increase in accuracy and a decrease in loss throughout the training process. This indicates that the model is able to learn image patterns effectively and maintains relatively stable performance on the validation data. Although there is a difference between training and validation results, the model still demonstrates a reasonably good generalization capability until the end of the training process. In addition, the integration of Grad-CAM improves interpretability by highlighting the image regions that the model focuses on during the decision-making process. These visualizations show that the model tends to attend to regions relevant to the shape of Hijaiyyah letters, thereby helping to explain the classification process.

This study does not include comparisons with other methods (baseline or state-of-the-art); therefore, the evaluation is solely based on the testing results of the proposed model. Future research is encouraged to conduct comparative studies using various CNN architectures or other classification methods to obtain a more comprehensive assessment of performance. Further development may include increasing the size and diversity of the dataset to improve model generalization, applying more complex architectures or transfer learning to enhance performance, and incorporating additional interpretability methods beyond Grad-CAM for a deeper analysis of the model's decision-making process.

5. REFERENCES

[1] R. M. Mohammed and S. M. Kadhem, "Automatic translation from Iraqi sign language to Arabic text or speech using CNN," Iraqi Journal of Computer Communication Control and System Engineering, pp. 112–124, Jun. 2023, doi: 10.33103/uot.ijccce.23.2.9.

[2] N. Bhagyawant, G. Tamondkar, S. Yadav, S. Kenche, and S. Sall, "Sign Language Detection and Recognition using Image Processing for Improved Communication," International Journal of Soft Computing and Engineering, vol. 15, no. 2, pp. 16–23, May 2025, doi: 10.35940/ijsce.b3668.15020525.

[3] A. Kasapbaşı and H. Canbolat, "A multi-stage convolutional and self-attention architecture for high-precision sign language gesture recognition," The Journal of Supercomputing, Mar. 2026, [Online]. Available: https://doi.org/10.1007/s11227-026-08481-x.

[4] A. A. Alani and G. Cosma, "ArSL-CNN: A convolutional neural network for arabic sign language gesture recognition," Zenodo (CERN European Organization for Nuclear Research), May 2021, doi: 10.11591/ijeecs.v22i2.pp1096-1107.

[5] N. Alasmari and S. Asiri, "ASLDetect: Arabic sign language detection using ResNet and U-Net like component," Scientific Reports, vol. 15, no. 1, p. 18012, May 2025, doi: 10.1038/s41598-025-01588-w.

[6] M. Balat et al., "Revolutionizing Communication with Deep Learning and XAI for Enhanced Arabic Sign Language Recognition," arXiv, Jan. 2025, [Online]. Available: https://arxiv.org/abs/2501.08169v1.

[7] H. M. L. S. Kumari and C. K. Walgampaya, "Explainability evaluation of transfer learning models utilized in multimodal translation of sign language," Engineer Journal of the Institution of Engineers Sri Lanka, vol. 59, no. 2, pp. 99–110, May 2026, doi: 10.4038/engineer.v59i2.7752.

[8] A. T. Elgohr, M. S. Elhadidy, M. El-Geneedy, S. Akram, and M. a. A. Mousa, "Advancing Sign Language Recognition: A YOLO V.11-Based deep learning framework for alphabet and transactional hand gesture detection," Proceedings of the AAAI Symposium Series, vol. 6, no. 1, pp. 209–217, Aug. 2025, doi: 10.1609/aaaiss.v6i1.36055.

[9] D. Mhnaa, Y. Dayoub, and J. Salman, "Development of an Intelligent System for Recognizing Islamic Religious Visual Signs in the Arabic Language," Computer Vision Foundation, p. 4933–4941, Oct. 2025, doi: 10.1109/iccvw69036.2025.00511.

[10] L. Zholshiyeva, T. Zhukabayeba, A. Serek, R. Duisenbek, M. Berdieva, and N. Shapay, "Deep Learning-Based Continuous Sign Language recognition," Journal of Robotics and Control (JRC), vol. 6, no. 3, pp. 1106–1119, May 2025, doi: 10.18196/jrc.v6i3.25881.

[11] W. Ismaiel, L. Kechiche, Y. Aribi, O. S. D. Omer, and W. Merghani, "Leveraging edge detection techniques to enhance Arabic sign language static-gesture recognition using deep learning," Journal of Engineering Research, vol. 14, no. 1, pp. 896–915, Sep. 2025, doi: 10.1016/j.jer.2025.09.011.

[12]https://www.kaggle.com/datasets/muhammadalbrham/rgb-arabic-alphabets-sign-language-dataset.

[13] Mosab. A. Hassan, Alaa. H. Ali, and A. A. Sabri, "Recent progress in Arabic sign language recognition: utilizing convolutional neural networks (CNN)," BIO Web of Conferences, vol. 97, p. 00073, Jan. 2024, doi: 10.1051/bioconf/20249700073

[14] M. Al-Hammadi, G. Muhammad, W. Abdul, M. Alsulaiman, M. A. Bencherif, and M. A. Mekhtiche, "Hand gesture recognition for sign language using 3DCNN," IEEE Access, vol. 8, pp. 79491–79509, Jan. 2020, doi: 10.1109/access.2020.2990434.

[15] S. Alyami and H. Luqman, "A Comparative Study of RGB-based Continuous Sign Language Recognition Techniques," Computer Vision Foundation, pp. 4923–4932, Oct. 2025, doi: 10.1109/iccvw69036.2025.00510.

[16] M. a. A. Mosleh and A. H. Gumaei, "An Efficient Bidirectional Android Translation Prototype for Yemeni Sign Language Using Fuzzy logic and CNN Transfer Learning Models," IEEE Access, p. 1, Jan. 2024, doi: 10.1109/access.2024.3512455