International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-13 Issue-8, August 2026

Development of a Deep Learning Model for Makhraj and Tajwid Classification in Qur'anic Recitation Using CNN-Bigru and CNN-Transformer

Agung Purwoto*, Yunita Sari

Abstract: ***Accurate Qur'anic recitation depends on two complementary skills: producing each Arabic letter from its proper articulation point (makhraj) and applying tajwid rules correctly in context. Most prior audio-based studies model these two aspects as a single task or reuse the same network design for both, even though makhraj is a fine-grained phonetic discrimination problem, whereas tajwid relies on longer contextual cues. We therefore propose two task-specific deep learning pipelines. For makhraj classification, we use a CNN-BiGRU model to capture local spectral patterns and short-range temporal dynamics. For tajwid classification, we use a CNN front end followed by an attention module to model contextual dependencies. We compiled the dataset from Abdul Basit Abdus Samad's public recitations and Ali Al-Hudhaify's YouTube recordings, totalling about ±12 hours of raw audio before segmentation and cleaning. Three certified tajwid experts annotated the segments, with approximately 95% inter-annotator agreement. On the makhraj task, the CNN-BiGRU achieved 72.66% validation accuracy and a character error rate (CER) of 27.34%, reflecting difficulty due to phonetic similarity and class imbalance. On the tajwid task, the CNN-attention model reached 99.68% validation accuracy with a CER of 0.31% and a word error rate (WER) of 0.27%, converging in roughly 10 epochs. Overall, self-attention proved highly effective for context-dependent tajwid recognition, while makhraj remained more challenging even with sequential modelling. The results underscore the value of designing architectures around each task's linguistic properties and support the development of more accurate and pedagogically useful Qur'anic learning systems.***

Keywords: ***Qur'anic Recitation, Makhraj, Tajwid, Deep Learning, CNN-BiGRU, Attention, MFCC, CER, WER.***

Nomenclature:
CER: Character Error Rate
WER: Word Error Rate
MFCCs: Mel-Frequency Cepstral Coefficients
GMMs: Gaussian Mixture Models
CTC: Connectionist Temporal Classification
DCNN: Deep Convolutional Neural Networks
BLSTM: Bidirectional LSTM

Manuscript received on 10 February 2026 | First Revised Manuscript received on 20 July 2026 | Second Revised Manuscript received on 02 August 2026 | Manuscript Accepted on 15 August 2026 | Manuscript published on 30 August 2026

***Correspondence Author(s)**
Agung Purwoto*, Master's Program in Artificial Intelligence, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia. Email ID: agung.purwoto@gmail.com
Yunita Sari, Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia. Email ID: yunita.sari@ugm.ac.id

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open-access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

I. INTRODUCTION

Correct Qur'anic recitation requires both phonetic precision and rule-aware sequencing. Makhraj captures phonetic precision, i.e., producing each Arabic letter from its correct articulation point. At the same time, tajwid governs how letters and words should be recited in sequence.

Because access to qualified teachers is not always evenly distributed, automated assessment tools can help learners practice consistently and receive feedback at scale.

Although modern speech processing has progressed rapidly, models tailored to Qur'anic recitation are still relatively limited. Many existing approaches either merge makhraj and tajwid into a single classification problem or reuse one architecture for both tasks. This choice is often inefficient: makhraj is a fine-grained, short-time phonetic decision, whereas tajwid depends on broader context across the recitation stream.

The literature highlights three recurring issues. First, architectural choices often do not align with the different inductive biases required by makhraj versus tajwid. Second, datasets are often constrained (e.g., relying on generic Arabic speech, non-audio modalities, or limited documentation of expert labelling), which affects reproducibility. Third, evaluation often relies on aggregate metrics (accuracy, CER, WER) without clearly linking them to phonetic precision and contextual correctness, and few studies discuss how model outputs translate into actionable feedback for learners. This work addresses these limitations by using task-specific modelling with expert-validated Qur'anic audio and reporting task-aligned metrics.

To address these issues, this work makes three contributions:

A. A curated Qur'anic audio dataset with expert-validated annotations for both makhraj and tajwid.
B. A CNN-BiGRU pipeline optimised for fine-grained phonetic discrimination in makhraj.
C. A CNN-based model with an attention module to capture contextual dependencies for tajwid.

II. RELATED WORKS

Research on Qur'anic recitation has advanced alongside developments in speech processing. Early systems typically combined handcrafted acoustic features with conventional classifiers to support pronunciation

© Copyright: All rights reserved.

assessment and learning. For example, Putra et al. (2012) [4] used Mel-frequency cepstral coefficients (MFCCs) with Gaussian Mixture Models (GMMs) to detect recitation errors and provide corrective feedback, showing that MFCC-based pipelines are practical even though they are less expressive than modern deep models.

As end-to-end deep learning became common, researchers adopted CNN–RNN hybrids to learn richer representations directly from audio. Al Harere and Al

Jallad (2018) [5] combined CNN and GRU layers with a Connectionist Temporal Classification (CTC) decoder and reported WER of 8.34% and CER of 2.42%, indicating that convolutional feature extraction paired with temporal modelling is effective for Qur'anic audio.

For phoneme-level correctness, several studies focus on Arabic pronunciation detection. Ziafat et al. (2021) [1] [3] compared Deep Convolutional Neural Networks (DCNN) and Bidirectional LSTM (BLSTM) for Arabic alphabet pronunciation and obtained higher accuracy with DCNN (95.95%) than BLSTM (88.32%) [2]. These findings are directly relevant to makhraj recognition, which requires distinguishing subtle acoustic differences among closely related phonemes.

Researchers have also studied tajwid using modalities beyond speech. Nuraeni and Pambudi (2024) used CNN-based image classification on Qur'anic text images to recognise tajwid rules and achieved 95.10% validation accuracy after data augmentation. While this approach is not audio-based, it reinforces that tajwid is inherently rule- and context-driven.

Other Qur'an-related audio tasks, such as reciter identification, provide useful guidance on feature representations and temporal modelling. Qayyum et al. (2018) [7] reported up to 99.89% accuracy with a two-layer BLSTM using MFCC features, outperforming classical classifiers. Khan et al. (2019) [9] combined MFCC and pitch features with spectrogram-based representations and achieved 88% accuracy with Random Forest and Naïve Bayes across multiple qaris.

Beyond speech, researchers have applied deep learning to Qur'anic text authentication and contextual modelling. Touati-Hamad et al. (2022) [6] compared CNN and CNN–LSTM for Arabic verse authentication using word embeddings; their CNN–LSTM reached 98.33% accuracy. Although the approach is text-based, the results highlight the value of modelling context in rule-governed structures, which is also central to tajwid recognition.

More recently, researchers have introduced attention-based models to Qur'an-focused automatic speech recognition and related tasks. Hadwan et al. (2023) [8] reported strong performance with end-to-end Transformer-based ASR for Qur'anic reciters, reflecting the broader success of self-attention in capturing long-range dependencies.

Taken together, prior studies show progress in Qur'anic speech recognition, pronunciation detection, and tajwid learning, but they rarely treat makhraj and tajwid as two distinct problems with different modelling needs. This gap motivates our task-specific design: CNN-BiGRU for makhraj and a CNN-based attention model for tajwid.

III. DATASETS

A. Data Sources

Two sources were used:

i. A public Qur'anic audio repository containing Abdul Basit Abdus Samad's recordings.

ii. A dataset collected from Ali Al-Hudhaify's official YouTube channel, extracted with FFmpeg and then processed with volume normalisation, noise reduction, and rule-based segmentation. The raw recordings total approximately ±12 hours before segmentation.

We selected these sources to balance consistency and real-world variability. The public repository provides relatively clean recordings suitable as a reference corpus. In contrast, the YouTube recordings introduce realistic variations (compression artefacts, background noise, and differing environments) that reduce reciter- or source-specific bias. Both sources used the same preprocessing pipeline, so observed performance differences reflect the inherent difficulty of makhraj and tajwid rather than differences in audio format or quality.

B. Annotation Protocol

Annotations followed a structured multi-stage procedure to ensure linguistic validity and consistency. Three certified tajwid experts independently labelled the audio based on established makhraj articulation points and tajwid rules. For makhraj, we labelled at the phoneme level by marking start and end timestamps for each articulated letter using acoustic cues and phonetic boundaries. For tajwid, we assigned labels at the word or phrase level, covering the time span where a rule occurs and its subcategory, when applicable.

To assess reliability, we measured inter-annotator agreement by comparing the initial labels and conducting cross-validation, yielding about 92% agreement for makhraj and 95% for tajwid. We flagged disagreements and resolved them through consensus discussion among the annotators. We clarified ambiguous makhraj boundaries by inspecting waveforms and spectrograms, and we verified uncertain tajwid cases using surrounding linguistic context and standard tajwid references. We retained only consensus samples in the final dataset.

The final datasets contain 3,024 makhraj samples across 28 classes and 3,002 tajwid samples across 20 classes. We split each dataset into 80% training and 20% testing, as summarised in Table I. The makhraj dataset is notably imbalanced, with a maximum class ratio of about 15:1 between the most and least frequent classes.

Table I Dataset Summary

Task	Total Sample	Classes	Train	Test	Audio Quality
Makhraj	3024	28	2,419 (80%)	605 (20%)	16 kHz, mono
Tajwid	3002	20	2,402 (80%)	600 (20%)	16 kHz, mono

Makhraj data shows a documented maximum imbalance ratio of 15:1 between majority and minority classes.

© Copyright: All rights reserved.

IV. PREPROCESSING AND FEATURE EXTRACTION

Feature design strongly affects speech classification performance because raw waveforms contain large variability and substantial redundant information. We therefore transform each audio segment into a compact time–frequency representation that captures perceptually meaningful spectral cues while remaining robust to recording differences.

We use Mel Frequency Cepstral Coefficients (MFCC) as the primary representation. The Mel-frequency mapping is defined as:

$$\text{Mel(f)} = 2595 \times log_{10}\left(1 + \frac{\text{f}}{700}\right)$$

After mapping frequencies to the Mel scale, we apply standard conditioning steps that improve discriminability for consonantal and articulation-related cues. Because high-frequency components often have lower energy, we use a pre-emphasis filter to boost high-frequency information and balance the spectrum.

Pre-emphasis is applied as:

$$y(t) = x(t) - ax(t-1), a \approx 0.95 \; to \; 0.97$$

Next, we frame the signal into short, overlapping windows to reflect speech's quasi-stationary assumption. To reduce discontinuities at frame boundaries and limit spectral leakage, we apply a Hamming window to each frame before computing spectral features.

Windowing uses a Hamming function:

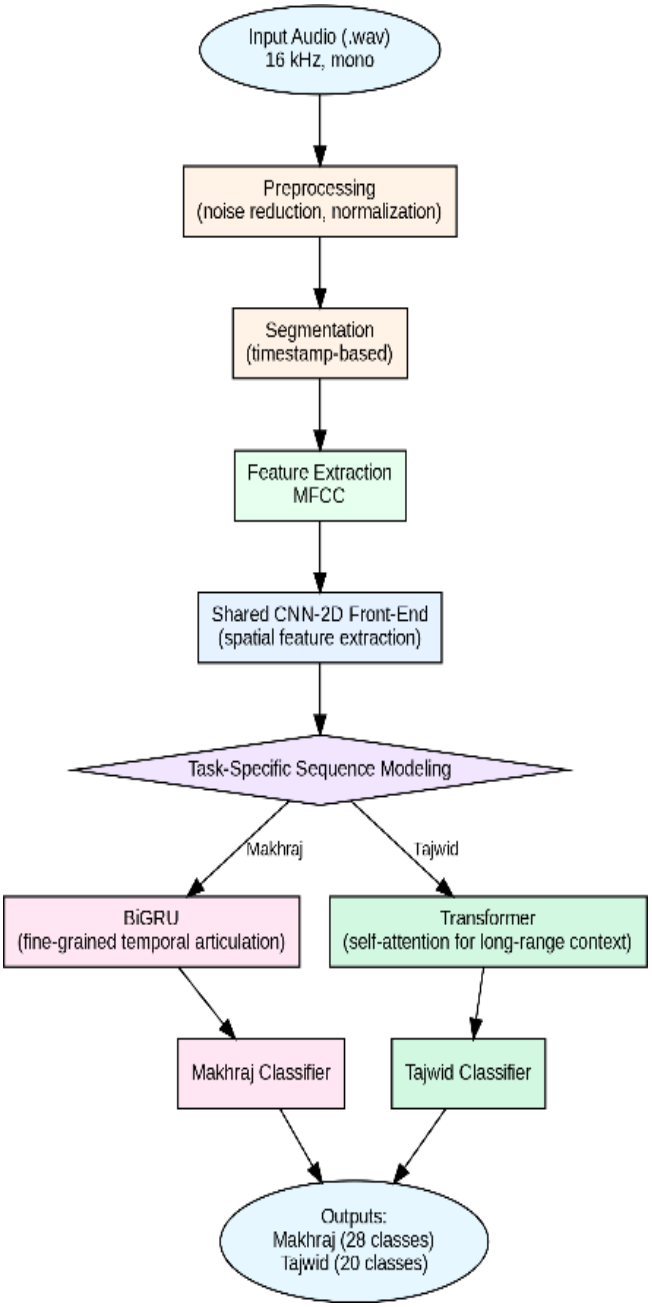
$$\omega(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right)$$

In summary, Mel-scale mapping, pre-emphasis, framing, and windowing together produce stable MFCC feature maps for learning. This pipeline emphasises perceptually relevant frequency content and reduces artefacts, providing a consistent input representation for the downstream deep models.

V. PROPOSED MODELS

Because makhraj and tajwid pose different learning problems, we use two dedicated architectures that share a common acoustic front end but differ in sequence modelling. Both pipelines begin with the same 2D CNN feature extractor operating on MFCC maps; the makhraj head uses BiGRU to model short temporal patterns, while the tajwid head uses an attention mechanism to capture broader contextual dependencies.

Figure 1 summarises the workflow. Segmented recitations are converted to MFCC maps and passed through a shared CNN front-end to learn compact time–frequency features. The network then branches: for makhraj, the features are reshaped into a sequence and processed by BiGRU before classification over 28 classes; for tajwid, the features are aggregated and processed by an attention module before classification over 20 rule classes. This design keeps low-level feature learning comparable across tasks while tailoring higher-level modelling to phonetic versus contextual requirements.



[Fig.1: Proposed Model Architecture]

A. Shared CNN-2D Feature Extractor

Both pipelines use the same CNN-2D backbone as a shared acoustic feature extractor. The input MFCC map has shape (141, 174, 1) and passes through two Conv2D layers with 32 and 64 filters (3×3 kernel, same padding), each followed by batch normalisation and 2×2 max pooling. With this configuration (Tables II and IV), the shared CNN front-end contains 19,200 trainable parameters and outputs a feature map of shape (35, 43, 64).

B. CNN-BiGRU for Makhraj

For makhraj recognition, we employ a CNN-BiGRU architecture. The CNN front end extracts local spectral patterns from MFCC inputs; we then reshape the feature map into a sequence and process it with a bidirectional GRU to capture temporal

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

dynamics in both directions. This helps when neighbouring context influences how a phonetic event is realised. We include batch normalisation and dropout to improve training stability and generalisation. Table II lists the full configuration.

Table II: CNN-BiGRU for Makhraj

No	Layer (Type)	Output Shape	Parameters
1	Input Layer (InputLayer)	(None, 141, 174, 1)	0
2	Convolution Layer (Conv2D)	(None, 141, 174, 32)	320
3	Batch Normalization	(None, 141, 174, 32)	128
4	Max Pooling Layer (MaxPooling2D)	(None, 70, 87, 32)	0
5	Convolution Layer (Conv2D)	(None, 70, 87, 64)	18,496
6	Batch Normalization	(None, 70, 87, 64)	256
7	Max Pooling Layer (MaxPooling2D)	(None, 35, 43, 64)	0
8	Reshape Layer	(None, 1505, 64)	0
9	Bidirectional GRU	(None, 256)	148,992
10	Dense Layer	(None, 128)	32,896
11	Batch Normalization	(None, 128)	512
12	Dropout	(None, 128)	0
13	Output Layer (Dense)	(None, 28)	3,612

Beyond the layer-by-layer specification, reporting parameter counts helps clarify model capacity and computational cost. Table III summarises the total, trainable, and non-trainable parameters of the CNN-BiGRU model and the number of output classes used for makhraj classification.

Table III: CNN-BiGRU Components for Makhraj

Component	Value
Total Parameters	2,05,212
Trainable Parameters	204,804
Non-trainable Parameters	408
Output Classes	28 (Makhraj)

As shown in Table II, two Conv2D blocks (with batch normalisation and max pooling) process the MFCC inputs to learn a compact time–frequency feature map. The map is reshaped from (35, 43, 64) to a sequence of length 1505 with 64 features per step, which is then modelled with a bidirectional GRU. The resulting representation is refined with a dense layer, batch normalisation, and dropout, and finally projected to 28 makhraj classes. Table III reports 205,212 total parameters, with the BiGRU block accounting for 148,992 parameters.

C. CNN-Transformer for Tajwid

For tajwid classification, we use a CNN-based model augmented with self-attention to capture contextual dependencies. The CNN front end learns discriminative spectral patterns from MFCC maps; we then project these features into an embedding space and process them with a multi-head attention layer to model interactions that help identify rules that depend on surrounding context rather than isolated sounds. Layer normalisation and dropout stabilise training and reduce overfitting. The detailed configuration is presented in Table IV.

Table IV: CNN-Transformer for Tajwid

No	Layer (Type)	Output Shape	Parameters
1	Input Layer (InputLayer)	(None, 141, 174, 1)	0
2	Convolution Layer (Conv2D)	(None, 141, 174, 32)	320
3	Batch Normalization	(None, 141, 174, 32)	128
4	Max Pooling Layer (MaxPooling2D)	(None, 70, 87, 32)	0
5	Convolution Layer (Conv2D)	(None, 70, 87, 64)	18,496
6	Batch Normalization	(None, 70, 87, 64)	256
7	Max Pooling Layer (MaxPooling2D)	(None, 35, 43, 64)	0
8	Flatten Layer	(None, 96,320)	0
9	Dense Layer	(None, 128)	1,23,29,888
10	Reshape Layer	(None, 1, 128)	0
11	Multi-Head Attention	(None, 1, 128)	2,63,808
12	Layer Normalization	(None, 1, 128)	256
13	Flatten Layer	(None, 128)	0
14	Dense Layer	(None, 128)	16,512
15	Dropout	(None, 128)	0
16	Output Layer (Dense)	(None, 20)	2,580

was done in Python 3.10 with TensorFlow; Scikit-learn utilities were used for evaluation. We split both datasets into 80% training and 20% testing using fixed random seeds. For the tajwid task, we applied an additional context-aware constraint during splitting to preserve contextual associations better.

We report accuracy, precision, recall, and F1-score, supported by confusion matrices. To better reflect sequence-level errors, we also report Character Error Rate (CER) and Word Error Rate (WER), computed as:

$$\text{CER} = \frac{S + D + I}{N} \times 100\%, WER = \frac{S + D + I}{N} \times 100\%,$$

Table V provides an overview of the tajwid model's parameter counts, including total, trainable, and non-trainable parameters, along with the number of output classes. This summary helps characterise model capacity and computational complexity.

Table V: CNN-Transformer Components for Tajwid

Component	Value
Total Parameters	1,26,31,444
Trainable Parameters	1,26,31,252
Non-trainable Parameters	192
Output Classes	20 (Tajwid)

Table IV details the CNN–attention architecture used for tajwid classification. The MFCC input passes through the same two Conv2D blocks used in the makhraj pipeline,

© Copyright: All rights reserved.

producing a (35, 43, 64) feature map. We flatten this map, project it to a 128-dimensional embedding, reshape it into a single-step sequence, and apply multi-head attention followed by layer normalisation. We then flatten the output, refine it with a dense layer and dropout, and map it to 20 tajwid classes. As shown in Table V, the model has 12,631,444 parameters in total; the dense projection layer and the multi-head attention layer contribute the most.

VI. EXPERIMENTAL SETUP

We trained and evaluated the model on Google Colab using an A100 GPU. In this implementation, *S*, *D*, and *I* are the counts of substitutions, deletions, and insertions, and *N* is the number of reference units.

VII. RESULTS

A. Model Configuration Summary

Table VI compares the two models by their shared CNN front end, task-specific sequence module, parameter counts, and output classes. This side-by-side view highlights how the architectures align with the different requirements of makhraj and tajwid.

Table VI: Model Configuration Summary

Model	CNN	Seq.	Seq. P	Total	Cls
BiGRU	19.2k	BiGRU	149k	205k	28
Attn	19.2k	MHA	12.61M	12.63M	20

Table VI summarises the parameter breakdown of the two architectures. Both models share the same CNN front end (19,200 parameters), keeping low-level time–frequency feature extraction consistent across tasks. For makhraj, the bidirectional GRU adds 148,992 parameters, and the full model totals 205,212 parameters across 28 classes. For tajwid, the attention-based sequence module (including the embedding projection and multi-head attention) dominates the parameter count, resulting in 12,631,444 total parameters over 20 classes.

B. Main Performance

Table VII reports validation performance for each task. We use validation accuracy as a broad measure of correctness, and CER/WER to capture finer-grained errors that are useful for recitation analysis. We also note training behaviour (e.g., convergence speed) to contextualise the results.

Table VII: Model Configuration Summary

Task	Model	Validation Accuracy	CER	WER	Training Notes
Makhraj	CNN-BiGRU	72.66%	27.34%	—	75 epochs
Tajwid	CNN-Transformer	99.68%	0.31%	0.27%	Converges ≈10 epochs

As shown in Table VII, makhraj classification is more challenging: the CNN-BiGRU achieved 72.66% validation accuracy with CER of 27.34% after 75 epochs. In contrast, the tajwid model reached 99.68% validation accuracy with very low CER (0.31%) and WER (0.27%) and converged in about 10 epochs. These results suggest that attention is especially effective when correct labelling depends on contextual cues, whereas makhraj requires finer acoustic separation among closely related phonemes.

VIII. DISCUSSIONS

Makhraj recognition remains intrinsically difficult because it requires discriminating fine phonetic differences and because the dataset is substantially imbalanced. The maximum imbalance ratio reaches about 15:1, which can bias the model toward frequent classes and increase confusion among letters with similar articulation zones.

The CNN-BiGRU architecture provides a reasonable baseline by combining local spectral feature learning with short-range temporal modelling, yet the moderate accuracy indicates room for improvement. Future refinements may include stronger class-balancing strategies, phoneme-aware loss functions, or multi-task learning that shares representations with related objectives.

Tajwid classification achieves near-perfect scores in this setup. However, the tajwid dataset is reported to be dominated by Mad Thabii (over 50% of samples), so aggregate accuracy may partially reflect majority-class prevalence. For a fairer picture, future work should report per-class metrics and consider rebalancing or stratified sampling.

Even with this caveat, the very low CER/WER and fast convergence indicate that the attention mechanism captures contextual dependencies effectively across recitation segments.

From an application standpoint, separating the pipelines is practical: learners benefit from phonetic-sensitive makhraj feedback, while tajwid correction benefits from context-aware rule tracking.

IX. CONCLUSIONS AND FUTURE WORK

This work proposes a dual-pipeline framework for automated assessment of Qur'anic recitation, combining a shared CNN front end with two specialised heads: CNN-BiGRU for makhraj and a CNN-based attention model for tajwid. On expert-annotated data, the approach achieved 72.66% validation accuracy for makhraj and 99.68% for tajwid with low CER/WER.

Future work should address class imbalance more aggressively, expand coverage across additional juz and reciters, and explore joint modelling of makhraj and tajwid to improve shared representation learning. Robustness tests under more diverse recording conditions would further strengthen readiness for deployment in learning tools.

ACKNOWLEDGMENTS

The author sincerely thanks his supervisor, Yunita Sari, S.Kom., M.Sc., PhD, for her guidance and support throughout this research, and the thesis examiners, Afiahayati, S.Kom., M.Cs., PhD, and Prof. Dr Azhari, M.T., for their valuable feedback and suggestions. The author also thanks colleagues who assisted with data collection and annotation, as well as mufassir and tajwid experts

© Copyright: All rights reserved.

who contributed makhraj and tajwid corrections and validation. Finally, heartfelt appreciation is extended to the author's family for their continuous

DECLARATION STATEMENT

Some of the cited references are older and are noted explicitly as **[1], [2] and [4]**. However, these works remain significant for the current study, as they are pioneering in their fields.

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of interest:** Based on my understanding, this article has no conflicts of interest.
- **Funding/Grants/Financial Support:** This article has not been sponsored or funded by any organisation or agency.
- **Ethical Approval and Consent to Participate:** The data provided in this article are exempt from ethical approval or participant consent requirements.
- **Data Access Statement and Material Availability:** The data in this article are publicly accessible.
- **Author's Contribution:** Agung Purwoto contributed to conceptualisation, methodology, software implementation, analysis, data curation, and drafting of the manuscript under the supervision of Yunita Sari.

REFERENCES

1. O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying Convolutional Neural Networks Concepts to Hybrid NN-HMM Model for Speech Recognition," IEEE, 2012. DOI: 10.1109/ICASSP.2012.6288864. Works remain significant; see the declaration.
2. O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," IEEE Transactions on Audio, Speech and Language Processing, vol. 22, no. 10, pp. 1533–1545, 2014, DOI: 10.1109/TASLP.2014.2339736, works remain significant; see the declaration.
3. N. Ziafat, H. F. Ahmad, I. Fatima, M. Zia, A. Alhumam, and K. Rajpoot, "Correct pronunciation detection of the Arabic alphabet using deep learning," Applied Sciences (Switzerland), vol. 11, no. 6, 2021. DOI: 10.3390/app11062508.
4. Putra, B. T. Atmaja, and D. Prananto, "Developing Speech Recognition System for Quranic Verse Recitation Learning Software," vol. 1, 2012. DOI: 10.14421/ijid.2012.01203. Works remain significant; see the declaration
5. Al Harere and K. Al Jallad, "Quran Recitation Recognition using End-to-End Deep Learning," 2018. DOI: 10.48550/arXiv.2305.07034
6. Z. Touati-Hamad, M. R. Laouar, I. Bendib, and S. Hakak, "Arabic Quran Verses Authentication Using Deep Learning and Word Embeddings," International Arab Journal of Information Technology, vol. 19, no. 4, pp. 681–688, 2022, DOI: 10.34028/iajit/19/4/13.
7. Qayyum, S. Latif, and J. Qadir, "Quran Reciter Identification: A Deep Learning Approach," IEEE, 2018. DOI: 10.1109/ICCCE.2018.8539336
8. M. Hadwan, H. A. Alsayadi, and S. Al-Hagree, "An End-to-End Transformer-Based Automatic Speech Recognition for Qur'an Reciters," Computers, Materials and Continua, vol. 74, no. 2, pp. 3471–3487, 2023, DOI: 10.32604/cmc.2023.033457.
9. R. U. Khan, A. M. Qamar, and M. Hadwan, "Quranic reciter recognition: A machine learning approach," Advances in Science, Technology and Engineering Systems, vol. 4, no. 6, pp. 173–176, 2019, DOI: 10.25046/aj040621

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Publisher /Journal and/or the editor(s). The Publisher /Journal and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.