The object of this study is a transformer-based ASR architecture trained using an Indonesian speech dataset consisting of audio recordings and corresponding transcripts. This study examines the development of an Automatic Speech Recognition (ASR) system for Indonesian, which is still classified as a low-resource language, particularly in terms of dataset availability and model performance. The problem addressed in this study is the limited performance of the standard transformer model in accurately recognizing Indonesian speech. To overcome this limitation, an encoder modification integrating convolutional and vision transformer (ViT) blocks was proposed and compared with the baseline model. The data were preprocessed through 16 kHz mono audio conversion, silence segmentation, pre-emphasis filtering, log-Mel spectrogram extraction, normalization, and subword tokenization using SentencePiece with byte pair encoding (BPE). The dataset was divided into training, validation, and testing sets with a ratio of 80:10:10, comprising 63,952, 7,994, and 7,994 samples, respectively. Model generalization was improved using the SpecAugment data augmentation technique. The experimental results show that the standard model achieves a word error rate (WER) of 0.162 and a character error rate (CER) of 0.121, while the modified model reduces the WER to 0.158 and the CER to 0.118. The significance of this finding lies in the improved feature representation produced by the modified encoder, where the convolutional block captures local acoustic patterns and the ViT block enhances global context modeling on the spectrogram. This complementary mechanism explains the reduction in errors at the word level, which is crucial for a reliable speech-to-text system. Therefore, the proposed model can be applied to real-time two-way communication in service robot applications

Keywords: ASR, modified transformer, SentencePiece, Indonesian dataset, deep learning

UDC 004.8:004.934
DOI: 10.15587/1729-4061.2026.350949

IMPROVING SPEECH-TO-TEXT FOR THE INDONESIAN LANGUAGE USING A MODIFIED TRANSFORMER

Ratna Atika
Doctor of Electrical Engineering, Student
Doctoral Program in Engineering Science**
ORCID: https://orcid.org/0009-0003-5459-0798
Suci Dwijayanti
Doctor in Electrical Engineering, Associate Professor*
ORCID: https://orcid.org/0000-0003-2060-6408
Bhakti Yudho Suprapto
Corresponding author
Doctor of Electrical Engineering, Associate Professor*
E-mail: bhakti@ft.unsri.ac.id
ORCID: https://orcid.org/0000-0002-3995-6347
*Department of Electrical Engineering**
**Universitas Sriwijaya
Palembang-Prabumulih str., 32,
Ogan Ilir, Indonesia 30662

Received 26.11.2025
Received in revised form 22.01.2026
Accepted 09.02.2026
Published 27.02.2026

How to Cite: *Atika, R., Dwijayanti, S., Suprapto, B. Y. (2026). Improving speech-to-text for the Indonesian language using a modified transformer. Eastern-European Journal of Enterprise Technologies, 1 (9 (139)), 78–90. https://doi.org/10.15587/1729-4061.2026.350949*

1. Introduction

Automatic speech recognition (ASR) is an important area of artificial intelligence that aims to transcribe speech signals into text. ASR faces significant challenges because speech signals are highly complex and can vary depending on the speaker, accent, noise, and intonation [1]. The end-to-end sequence-to-sequence approach, particularly the vanilla transformer architecture, is gaining popularity because it uses a self-attention mechanism to capture long-term dependencies in sequential data. This replaces older systems that required separate components such as acoustic models and language models. [2] shows that fine-tuned transformer models, such as bidirectional encoder representations from transformers (BERT), provide superior performance in text normalization tasks for voice applications and reduce errors compared to earlier recurrent neural network (RNN) approaches. Furthermore, [3] the development of efficient transformer variants such as Longformer enables direct processing of speech spectrograms without convolutional downsampling, improving efficiency and accuracy in speech-to-text translation. In the context of two-stage speech recognition systems, [4] the transformer-based deliberation rescoring method replaces traditional long short-term memory (LSTM) rescoring models with more efficient transformer-based models capable of processing tokens in parallel. This approach integrates context from audio encoder features and first-stage decoding hypotheses and trains the model using a combination of cross-entropy and minimum word error rate (MWER) losses, resulting in up to an 18% reduction in word error rate (WER) and improved computational efficiency. To address the modality gap between speech and text representations in speech translation, [5] proposes a new pre-training technique that integrates connectionist temporal classification (CTC) and optimal transport (OT) with Wasserstein distance. This approach jointly trains two encoders with OT- and CTC-based losses, producing more aligned cross-modal representations and significantly improving speech translation performance without the need for external data. Collectively, these advancements highlight the potential of vanilla Transformers and their efficient variants as strong foundations for developing more accurate ASR and speech translation systems capable of handling the complexity of speech signals.

Recent studies over the past five years also demonstrate significant progress in the application of transformer models to Indonesian language processing, from text analysis to speech recognition. The IndoBERT model, a BERT variant specifically developed for Indonesian, has improved the accuracy of senti-

ment analysis and emotion classification in social media data, demonstrating the transformer's ability to capture local linguistic nuances [6]. In machine translation, transformer-based models have also shown strong performance in translating Indonesian to Sundanese, with relatively lightweight configurations using encoder-decoder depths of 2, 4, 6, and 8 layers, demonstrating the feasibility of high-quality translation in low-resource scenarios [7, 8]. Furthermore, transformer models outperform Bidirectional LSTM approaches in Indonesian text topic segmentation, achieving accuracy above 98% by leveraging multi-head attention and positional embeddings to effectively capture inter-sentence relationships [9]. In ASR, a transformer-based system (Kaituoxu SpeechTransformer) outperformed the RNN-based Mozma DeepSpeech model, achieving a lower WER of 22.00% compared to 23.10% [10]. Additionally, the performance of Transformers in Indonesian ASR tasks is comparable to English datasets, using private Indonesian speech data that exhibit nearly similar results to public English datasets, with WER and CER values of 27.34 and 7.96 for Indonesian and 25.28 and 6.02 for English, respectively [11]. These results highlight the strong potential of transformer models for Indonesian speech recognition applications.

The development of a modified basic transformer model is needed to improve the accuracy of speech-to-text recognition in Indonesian in order to overcome the challenges of Indonesian as a language with limited resources and diverse dialect variations. The implementation of this modification can also improve the performance of two-way communication systems in service robots, accelerate the automatic translation process, and support the development of AI-based applications that are more adaptive and responsive to the needs of local users.

2. Literature review and problem statement

Studies related to the transformer method show that this model plays an important role in various natural language processing applications, especially for languages with limited resources such as Indonesian. Paper [8] shows the results of research on the impact of the combination of encoder-decoder layer depth on the transformer model and the type of activation function in the standard transformer feed-forward layer on model performance, in the task of translating Indonesian into Sundanese. However, statistically, the results obtained from variations in layer depth and activation function type did not show significant differences. In paper [9] on Indonesian language comprehension and text segmentation tasks, the transformer model achieved high accuracy and outperformed traditional LSTM-based approaches thanks to its attention mechanism's ability to capture the global context of sentences. However, the Indonesian language dataset used was derived from a collection of texts taken from scientific articles and news, not voice data. Paper [12] explored the influence of the ReLU activation function on the transformer model, which was superior to other activation functions, on the performance of Indonesian to Sundanese translation. However, the dataset used in the development of this translation model was limited to text data only. Paper [13] also successfully applied a transformer-based architecture in Javanese language understanding through pre-training using the Wikipedia corpus and embedding transfer from English, achieving performance equivalent to multilingual models. However, this pre-trained model is limited to documents from the Javanese Wikipedia corpus. Paper [14] discusses the development of Indonesian text-to-text transfer transformer (idT5) to improve model efficiency, namely a reduced and customized multilingual transformer model T5 (mT5) for Indonesian, and successfully reduced the model size by up to 58% for selected vocabularies. This paper shows comparable performance to the original mT5 on NLP tasks such as sentiment analysis, question generation, and question answering with faster inference. However, these models were only retrained for one epoch on all tasks, meaning that the reported performance is not the optimal performance of either model, and the comparison may not fully reflect their best potential. This article [15] discusses automatic grammar error correction (GEC) for Indonesian using transformer-based neural models and develops a large synthetic corpus through a semi-supervised method (by introducing controlled errors). The proposed system, IGEC, uses a transformer model (SAN-GEC) with a copy mechanism and byte pair encoding (BPE). The results show a significant improvement in the accuracy of Indonesian grammar error correction (F1 and BLEU scores). Although the method proposed in this study is capable of generating various training patterns, it still cannot cover the most difficult grammatical errors, such as semantic and syntactic errors. This paper [16] proposes a new approach to image labeling in rooms using Indonesian, utilizing the transformer architecture by modifying the MSCOCO dataset using Indonesian descriptions that are rich in object details (name, color, position, environment). However, this study has limitations in the scope of the dataset, which is limited to only ten objects in a specific room, and the custom dataset created is relatively small, so it is not large enough to optimally train deep learning models to generalize well to various scenarios.

Beyond text and language processing, the transformer-based method in this paper [17] demonstrates performance in Indonesian speech recognition by developing a transformer-based grapheme-to-phoneme (G2P) model with a combination of CNN as an encoder and Bi-LSTM as a decoder. This model was trained using a KBBI dataset formatted similarly to CMUdict. This approach achieved a low word error rate (WER) of 6.7% and an accuracy of 93.3% on the KBBI dataset. However, this study has limitations in the size of the KBBI dataset used, which is smaller than CMUDict, potentially limiting model generalization and data format dependency. This paper [18] develops an end-to-end transformer-based speech recognition model that significantly reduces the word error rate (WER) compared to conventional DeepSpeech and Speech-transformer systems. The evaluation shows that the developed model achieves a word error rate of 14.172%. However, this study is still limited to the use of Indonesian language datasets in text form. This article [19] develops a named entity recognition (NER) model for Indonesian called TWCAM (transformer-Word2Vec-CNN-Attention). This model combines Word2Vec and CNN for embedding, transformer with multi-head attention for context, and CRF for sequence tagging. The main goal is to improve NER accuracy, especially on limited datasets and "long-tail" problems, by utilizing better word representations and context understanding. However, this study is limited to the use of relatively small Indonesian language datasets with high computational complexity. Additionally, paper [20] trains a transformer-based model for Indonesian hoax news classification using pre-trained multilingual transformer models (XLM-R and mBERT) combined with BERTopic. However, this study is still a pre-training model that has limitations in terms of relatively small data and is still in the form of text data.

According to the previous studies described above, studies on transformer models generally focus on optimizing accuracy for various tasks such as text translation [8, 12, 15, 18, 19], image tagging [16], text segmentation [9], pre-training models [13, 14], Indonesian speech recognition [17], and separate language processing [15–20]. However, in these studies, all research is limited to the simulation stage and does not consider real-time system integration. In addition, previous studies used text data rather than voice data because of the limited availability of Indonesian speech datasets with extensive vocabularies and natural speech variations.

Although the performance of transformer-based models has been extensively tested on various individual tasks, there are still challenges in their application to real-time two-way communication systems for Indonesian-speaking service robots. Therefore, it is necessary to develop a modified transformer architecture and provide an Indonesian speech dataset that supports real-time two-way interaction in order to bridge the gap between simulation and real-world implementation.

3. The aim and objectives of the study

This study develops a modified transformer model to improve speech-to-text accuracy in Indonesian, focusing on architectural enhancements to the vanilla transformer for two-way communication in service robots.

To achieve this aim, the following objectives were accomplished:

– to optimize a modified vanilla transformer's performance in addressing limited resources and diverse Indonesian dialects, enabling a more accurate automatic speech recognition system that supports two-way interaction with service robots;

– to systematically evaluate the performance of the modified transformer architecture compared to the original vanilla transformer architecture in automatic speech recognition tasks using the NSS-ID dataset.

4. Materials and methods

4. 1. The object and hypothesis of the study

The object of this study is a transformer-based ASR architecture trained using the NSS-ID (Nusantara Speech Sample Indonesian Dataset). NSS-ID is an Indonesian audio recording dataset that contains a variety of dialects from several provinces across the Indonesian archipelago. This study aims to evaluate the performance of the transformer model and its modifications using an Indonesian dataset. Therefore, it is hypothesized that the proposed modifications to the transformer architecture may improve ASR accuracy compared to the original model. The simplifications adopted in this study involve the use of a multilayer perceptron (MLP), which is expected to enhance the training efficiency of the modified transformer.

4. 2. NSS-ID (Nusantara Speech Sample Indonesian Dataset)

NSS-ID is a speech corpus comprising recorded audio samples that capture diverse Indonesian dialects. The data were collected in a controlled recording environment equipped with sound-absorbing materials to reduce acoustic noise and ensure high-quality audio acquisition. The recording process was conducted using Audacity software version 3.7.4 and a supercardioid microphone. All voice data processing was performed using the Python programming language with Visual Studio Code version 1.109 and Google Colab. The training process was carried out on a laptop with the following specifications: GPU: RTX 4060 (Laptop) with 8 GB VRAM; CPU: Intel i7-13650HX; RAM: 44 GB DDR5 (4800 MT/s); SSD NVMe: 2.5 TB; Environment: Conda running on WSL2 (Ubuntu) and Google Colab with NVIDIA-SMI 580.82.07 and CUDA version 13.0. The resulting dataset includes 15 Indonesian dialects collected from 20 respondents. Each respondent recited 11 types of sentences, resulting in 220 sentences per speaker. Each sentence was repeated 25 times with variations in intonation, loudness, and speaking rate. This dataset was specifically designed to support automatic speech recognition systems intended for real-time two-way communication in service robots. The dataset is provided in WAV format, with a total size of 16 GB and an overall audio duration of 120 hours. It includes recordings from 20 unique speakers aged between 20 and 37 years. The dataset is dominated by male speakers, with 60% of the recordings produced by men and 40% by women. The word distribution in the NSS-ID dataset is shown in Fig. 1.

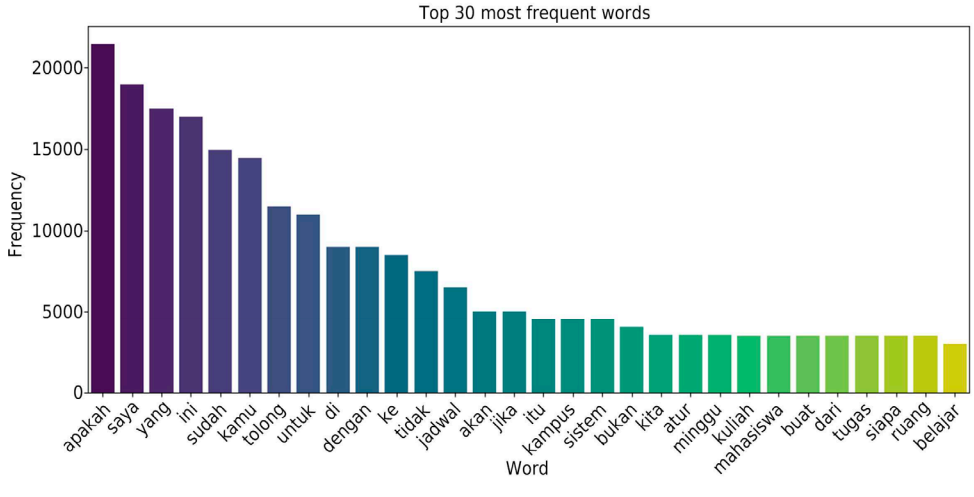


Fig. 1. Word distribution in the NSS-ID dataset

Fig. 1 is a bar chart showing the 30 most frequently occurring words in the NSS-ID text collection, with the x-axis (Words) representing Indonesian words and the y-axis (Frequency) representing their number of occurrences. Most of the high-frequency words are dominated by functional Indonesian words, such as "apakah" with a frequency of approximately 23,000 occurrences, followed by "saya" with around 18,000 occurrences, and then "yang," "ini," and "sudah." These five most frequent words "apakah," "saya," "yang," "ini," and "sudah" indicate that the dataset contains many utterances in the form of questions, personal statements, and explanatory sentence structures. The word "apakah" suggests a high proportion of interrogative sentences, while "saya" reflects conversational or dialogue-based patterns involving personal references. The words "yang" and

"ini" are functional words commonly used in Indonesian to connect clauses and refer to specific objects, while "sudah" frequently appears in contexts indicating completion or confirmation of an action. The dominance of these words suggests that the corpus largely consists of question-answer exchanges and descriptive conversational interactions. Overall, the distribution displays a long-tail pattern in which a small number of words appear at very high frequencies, while the majority appear much less frequently. This reflects the natural characteristics of Indonesian, where functional and structural words dominate everyday language use.

The word-length distribution in the corpus used in this study is shown in Fig. 2. In general, the distribution exhibits a unimodal pattern, with most words falling within the range of 4 to 7 characters. This range contains the highest number of words, with the peak occurring at approximately 5 characters. Words longer than 10 characters show a sharp decline, indicating that very long words are not commonly used in this dataset. This distribution aligns with the natural characteristics of the Indonesian language, in which most words are of moderate length and have relatively consistent morphological structure. In this study, only 8 types of sentences as the initial sample dataset were used, namely: interrogative, imperative, declarative, clarifying, confirming, scheduling, negative, and rhetorical sentences.

Fig. 3 shows a random sample spectrogram from the dataset.

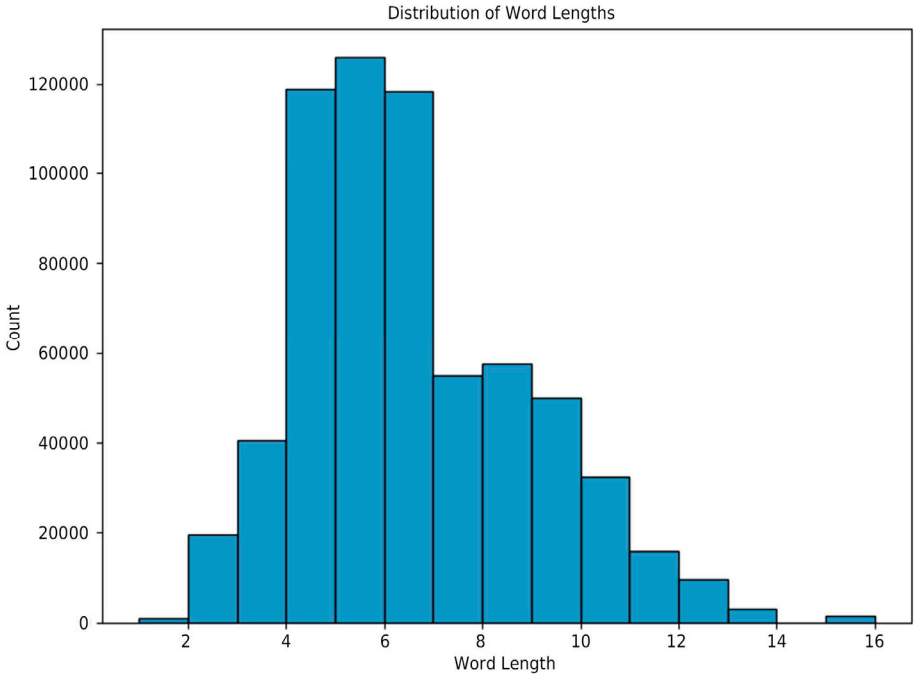


Fig. 2. Distribution of word lengths

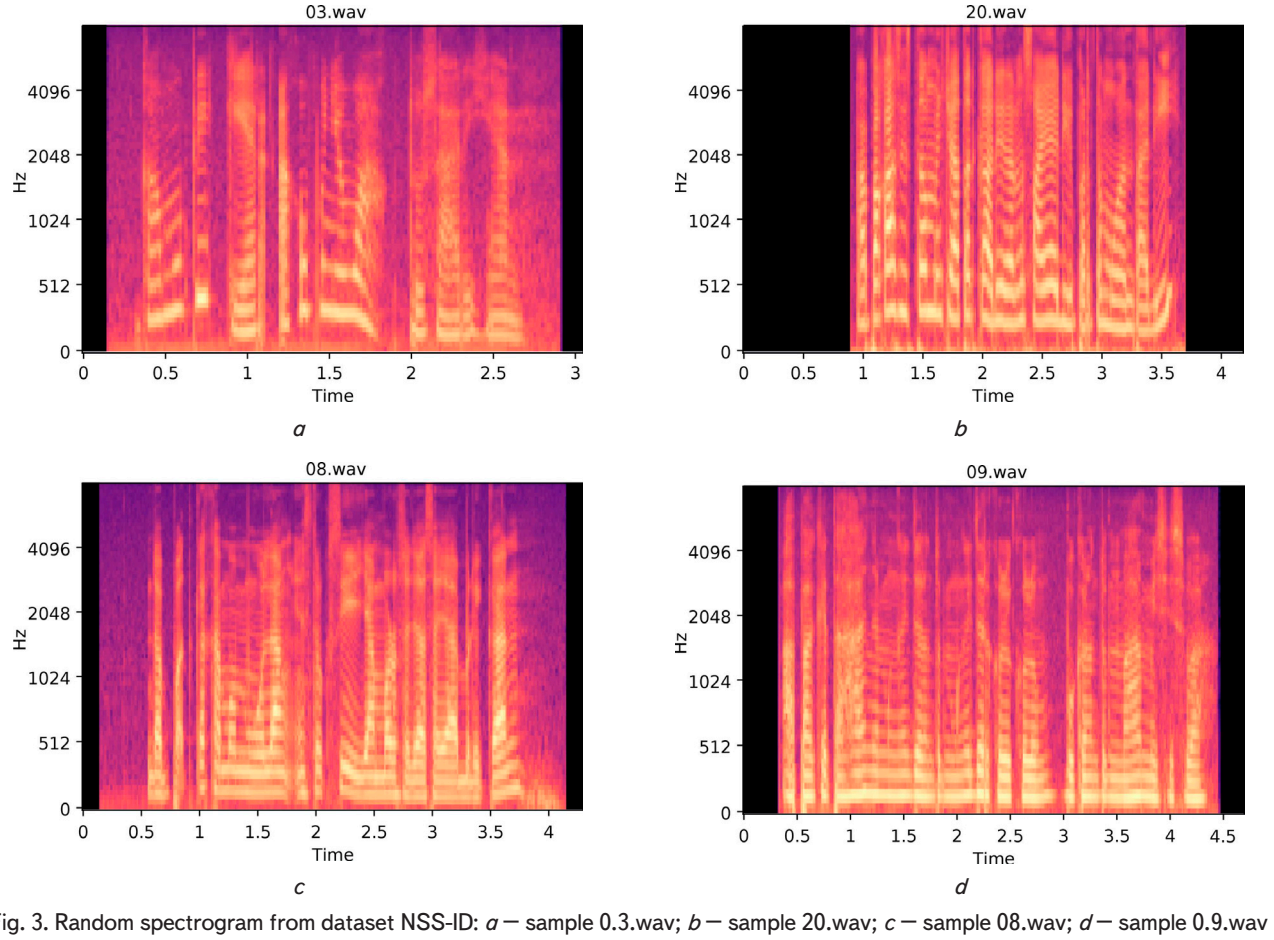


Fig. 3. Random spectrogram from dataset NSS-ID: *a* — sample 0.3.wav; *b* — sample 20.wav; *c* — sample 08.wav; *d* — sample 0.9.wav

Fig. 3 shows several spectrogram samples labeled 03.wav, 20.wav, 08.wav, and 09.wav, which are used for input in speech recognition. Spectrograms are used to represent audio signals in the time-frequency domain so that speech acoustic features can be extracted and recognized effectively by the ASR model.

4. 3. Modified vanilla transformer

The vanilla transformer is the original architecture introduced by [11] for sequence modeling and transduction tasks, such as language modeling and machine translation. It features six identical encoder-decoder layers and an attention mechanism designed to overcome the limitations of traditional recurrent models and encoder-decoder frameworks [8, 21]. The vanilla transformer model consists of an encoder and a decoder; using this encoder-decoder structure ensures that the sequence of speech generated by the decoder follows the input sequence from the encoder, thereby preserving the meaning of the utterance in a sentence [11]. Furthermore, to normalize the statistical distribution of dataset utterances, normalization layers are used in both the encoder and decoder so that the training process becomes faster and the resulting training and validation gradients become smoother.

One of the defining characteristics of this model is the use of multi-head attention. Multi-head attention enables the model to capture relationships between words more effectively, allowing transformer predictions to form sentences with correct semantic meaning. In addition, scaled dot-product attention one of the core components of the architecture computes "attention" between elements in the input, allowing the model to focus on relevant parts of the sequence based on their relationships. This mechanism is highly effective for tasks such as language translation, text comprehension, and other sequence-processing applications. To further improve model performance, the attention mechanism can be parallelized, known as multi-head attention.

The popularity of the vanilla transformer has led to numerous architectural modifications and variations, including X-formers, which have been widely used in computer vision, audio processing, and natural language processing [21]. The development of other transformer-based models [22] has expanded into multimodal architectures capable of handling various modalities simultaneously through tokenization, embedding, and flexible attention mechanisms. Study [8] has explored the impact of encoder-decoder depth and activation functions on the vanilla transformer, showing that although certain configurations (e.g., depth = 6 with ReLU activation) result in the lowest training and validation losses, the performance differences across depths and activation functions are not statistically significant. Nevertheless, the vanilla transformer remains the foundational baseline for subsequent architectural innovations [21]. The architectural design in this study develops a modified model derived from the vanilla transformer architecture, as shown in Fig. 4. In this modification, the encoder and decoder are still used because they can produce more accurate outputs by leveraging complex relationships between input elements. The encoder optimally captures global relationships among inputs, while the decoder combines information from the encoder with previous predictions to generate outputs incrementally. However, each feed-forward network layer is replaced with a multi-layer perceptron (MLP), as MLPs offer more flexibility in layer structure compared to standard feed-forward layers. The number of layers and their sizes are adjusted according to system requirements. The use of MLPs in this modified architecture allows backpropagation to minimize loss more effectively, potentially yielding more optimal results than using conventional feed-forward layers.

Modifications to the activation function are also implemented, using SiLU, ReLU, or Leaky ReLU depending on which yields the best performance. In the input embedding stage, the embedding layer converts input text into a fixed-dimensional vector representation. Positional encoding is then added to the embedding to provide positional information for each word in the sequence. In the multi-head attention component, the encoder computes self-attention to understand the relationships between tokens in the input sequence and to capture diverse relational patterns.

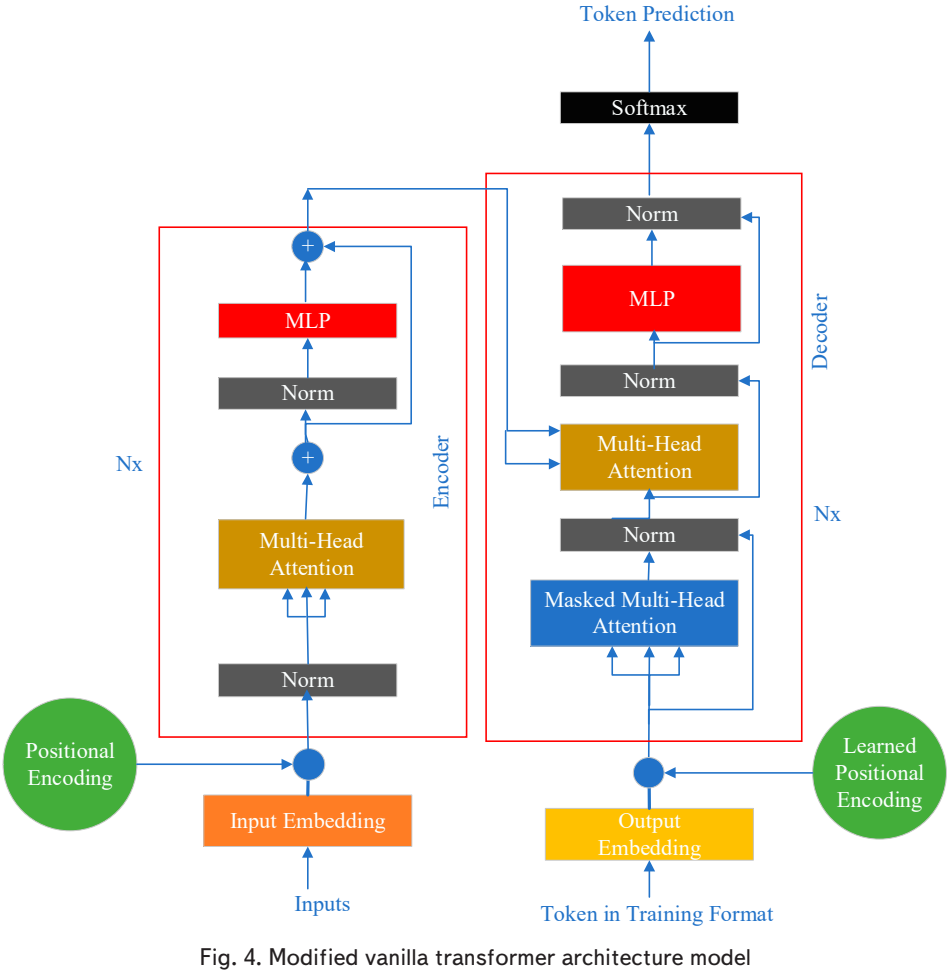


Fig. 4. Modified vanilla transformer architecture model

Next, normalization is applied to stabilize training and accelerate convergence before proceeding to the MLP block. This encoder layer is repeated N times to enrich the learned representations. The encoder output is the final representation of all input tokens, which is then forwarded to the decoder. In the decoder, this modified architecture still adopts the whisper transformer decoder, which uses tokens and learned positional encoding to ensure that the relative positions of tokens remain detectable. During the decoder process, masking is applied to ensure that predictions depend only on previous tokens, thereby maintaining causality. In the multi-head attention module, attention is computed using the encoder output to understand the input context. Afterward, normalization is performed to maintain training stability before entering the MLP block.

Finally, in the output stage, the probability distribution over the target classes is computed using the Softmax activation function to determine the next token. The training process for this modified architecture is shown in Fig. 5, *a*. This training procedure uses the vision transformer (ViT), which begins by transforming the audio signal input using the fast Fourier transform (FFT) to convert each frame of the signal into the frequency domain, producing a frequency spectrum. The FFT output is then converted to the Mel scale using a Mel filter bank, making the frequency representation more aligned with human auditory perception. With ViT, the spectrogram image is divided into several small parts called "patches," and each patch is treated as a token in the transformer model. Each patch is converted into a vector using an embedding layer, producing lower-dimensional representations that can be processed by the transformer. Every patch is represented as an embedding similar to tokens in NLP. Positional encoding is added to each patch embedding (patch + position embedding) so that the transformer model can recognize the relative position of each patch within the image.

Afterward, the encoded patches with positional information are processed by the transformer encoder, where self-attention is applied to capture cross-patch relationships and understand how different parts of the image are connected. In this ViT model, a special token called the classification token (CLS) is used at the beginning of the input for classification purposes. After the self-attention process, this CLS token is extracted to generate class predictions from the image. Furthermore, modifications to the MLP section are shown in Fig. 5, *b*, where the activation function (GeLU) is replaced and adjusted according to the activation function that yields the best performance (e.g., ReLU, SiLU, Leaky ReLU).

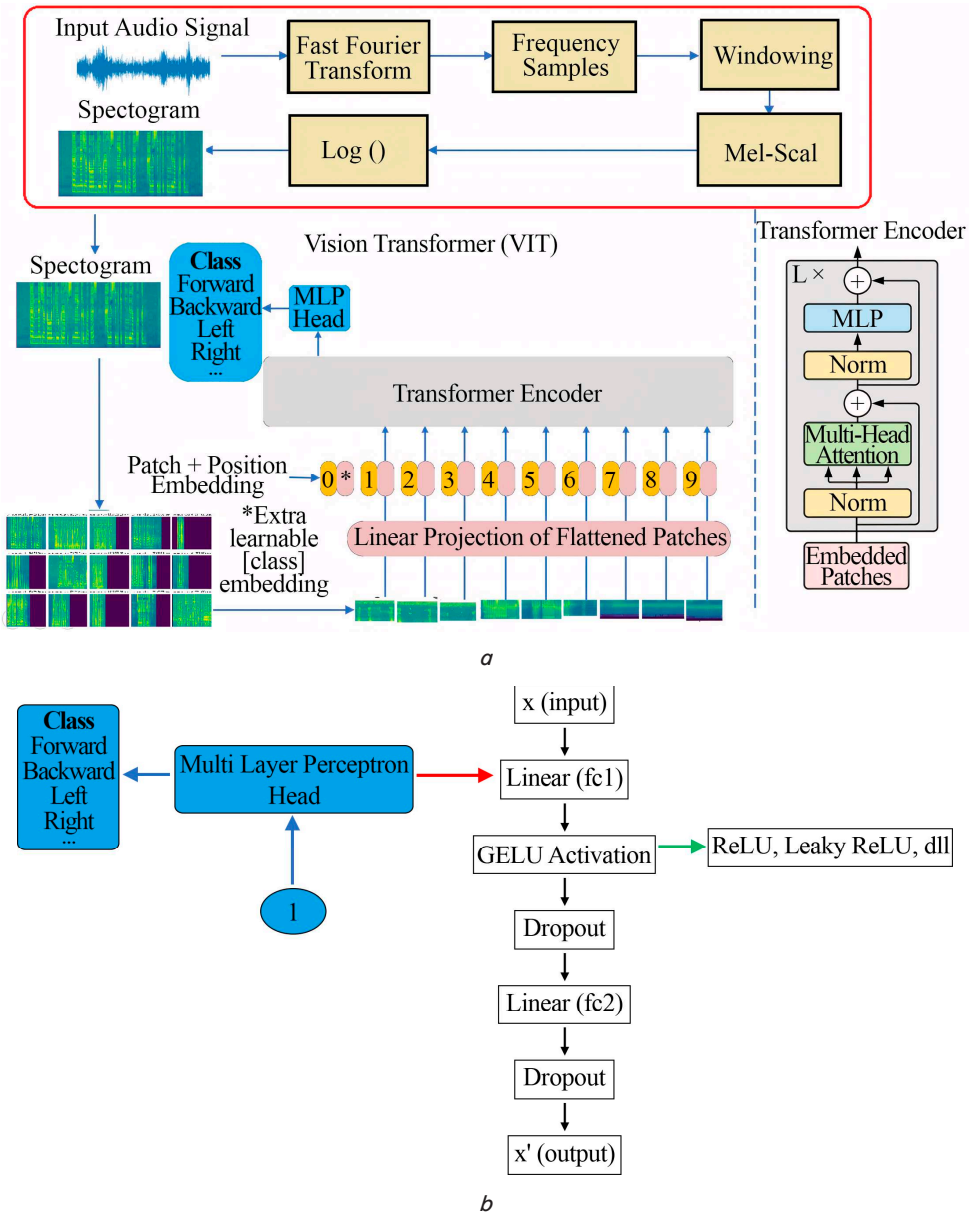


Fig. 5. Modified transformer: *a* — model training procedure; *b* — architecture using multi-layer perceptron (MLP)

5. Results of the modified vanilla transformer

5. 1. Optimization of a modified vanilla transformer model

Fig. 6 illustrates the initial stage of the audio dataset preparation. As shown in Fig. 6, *a*, the process begins with the compilation of the NSS-ID dataset, which is the primary audio dataset recorded by the researcher and involves 20 respondents. Once collected, the availability of text tran-

scripts is verified. If transcripts are missing, manual or semi-automatic transcription is performed to ensure that each audio file has a corresponding text pair. All audio files are then standardized to .wav format to meet digital signal processing requirements. After formatting, the dataset is divided into training, validation, and test sets with an 80:10:10 ratio, corresponding to 63,952; 7,994; and 7,994 samples, respectively.

The next step is training the tokenization model using SentencePiece with the byte pair encoding (BPE) algorithm, as shown in Fig. 6, *b*. Transcript files in .csv or .txt format are first cleaned by removing irregular characters, converting text to lowercase, and eliminating unnecessary punctuation to maintain consistency. The tokenization model is then trained using key parameters, including model_type = bpe, vocab_size = 80, and character_coverage = 1.0, with predefined special token indices (pad_id = 0, unk_id = 1, bos_id = 2, eos_id = 3). Training is performed using the SentencePieceTrainer.Train() function.

The output consists of spm_asr.model, representing the trained tokenization model, and spm_asr.vocab, containing the generated vocabulary list. These files are used to tokenize both input and output text data. The complete SentencePiece BPE training parameters are presented in Table 1.

Table 1

SentencePiece Bpe model training parameters

Parameter	Value	Function
model_type	bpe	Determining the type of tokenization algorithm used, namely byte pair encoding (BPE)
vocab_size	80 (adjustable)	The size of the vocabulary to be generated; can be adjusted according to the needs of the dataset
character_coverage	1.0	Determining the character coverage in the data; 1.0 means that all characters are covered
pad_id	0	A special ID for padding tokens used when adjusting sequence length
unk_id	1	Special ID for unknown tokens (characters/words not in the vocabulary)
bos_id	2	Special ID for begin of sentence tokens (sentence start markers)
eos_id	3	Special ID for end of sentence tokens (sentence end markers)

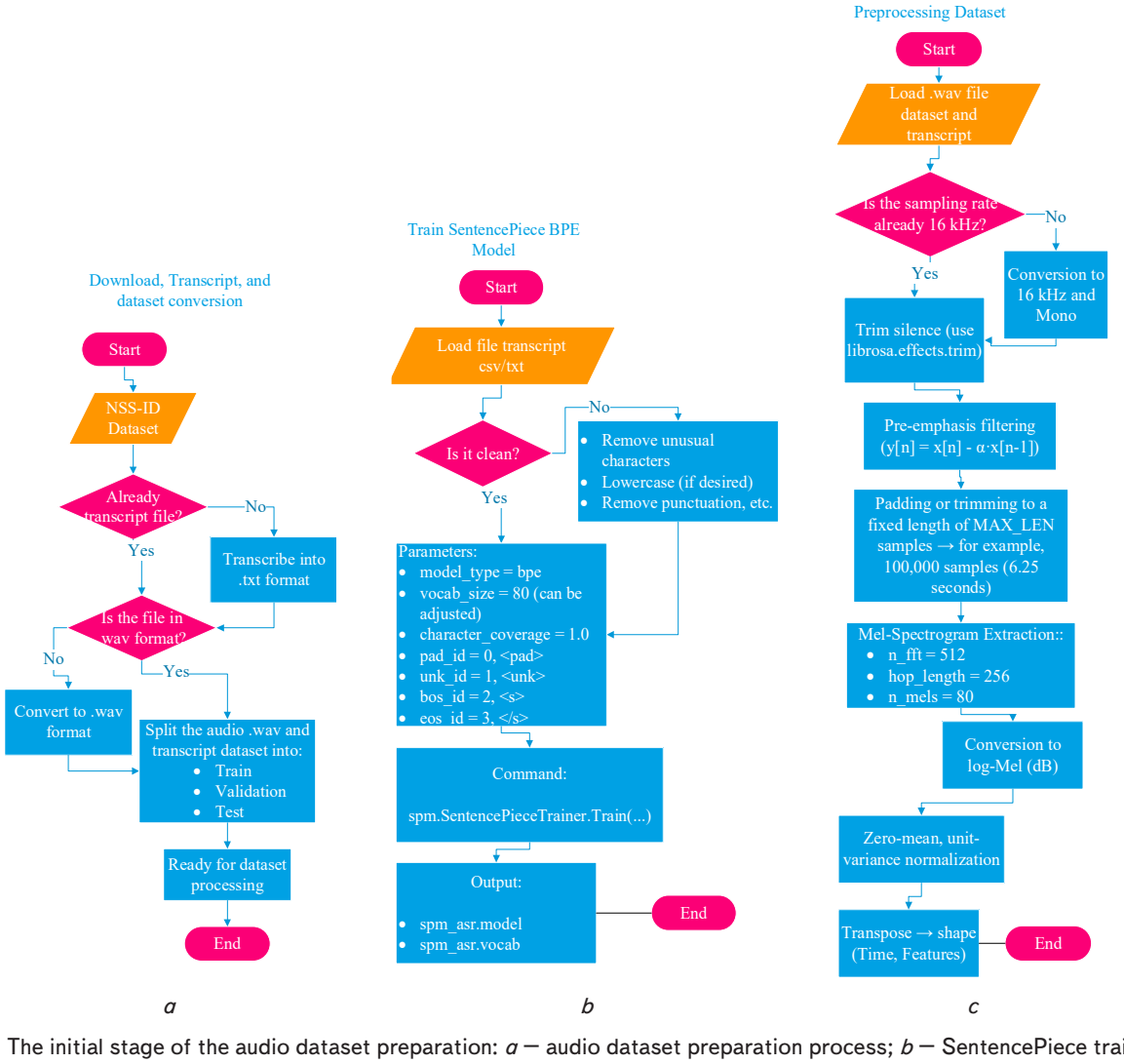


Fig. 6. The initial stage of the audio dataset preparation: *a* — audio dataset preparation process; *b* — SentencePiece training process with the BPE (byte pair encoding) model; *c* — preprocessing of audio datasets (.wav) and transcripts

Fig. 6, *c* shows the audio dataset preprocessing stage, which is carried out to ensure the consistency and quality of the input before it is used for training the speech-to-text model. The process begins by loading audio files in .wav format along with their corresponding text transcripts. Next, the audio sampling rate is checked. If it does not meet the 16 kHz standard, the audio is converted to 16 kHz and mono for uniformity.

Silence at the beginning and end of the signal is removed using the librosa.effects.trim function. The next step is pre-emphasis filtering with a coefficient of 0.97, which aims to enhance high-frequency components and improve the signal-to-noise ratio (SNR). To ensure uniform signal length, padding or trimming is applied so that all audio signals have a fixed duration for example, 100,000 samples (approximately 6.25 seconds). The results before and after pre-emphasizing can be seen in Fig. 7, *a*.

The main feature used is the Mel-spectrogram, extracted with the parameters n_fft = 512, hop_length = 256, and n_mels = 80. The resulting spectrogram is then converted to a logarithmic (dB) scale to better match human auditory perception. Zero-mean and unit-variance normalization are subsequently applied to ensure a stable feature distribution during training. Finally, the spectrogram is transposed so that the data format becomes (time, features), which matches the input format required by the transformer model. An example of the spectrogram used as the input feature can be seen in Fig. 7, *b*.

The mel-spectrogram extraction parameters are presented in Table 2, and the tokenization process using SentencePiece is shown in Table 3.

Table 2

Mel-Spectrogram extraction parameters

Parameters	Value
Sampling rate	16 kHz
n_fft	512
hop_length	256
n_mels	80
Window function	Hann
Scale	log-Mel (dB)
Normalization	Zero-mean, unit-variance

Table 3

Tokenization process with SentencePiece

Process	Example	Description
Original sentence (input)	hello world	Raw text sentence from the transcript
Normalization	hello world	Already in lowercase, without additional punctuation
Subword tokenization	[“hello”, “_world”]	Sentence split into subword units (subwords)
Add special tokens	[“<BOS>”, “hello”, “_world”, “<EOS>”]	BOS at the beginning of the sentence, EOS at the end of the sentence
Token ID (output)	[2, 121, 543, 3]	The result of mapping subwords to numeric IDs according to the model vocabulary

The next step in Fig. 8, *a* is tokenizing the text into subwords using the pre-trained SentencePiece model. This process begins by loading the tokenization model (.model) to ensure consistency in mapping words or phrases into numerical representations. Each sentence in the transcript is then encoded into a sequence of token IDs that correspond to the training vocabulary.

To maintain the data structure during the training process, special tokens BOS (Begin of Sentence, id = 2) and EOS (End of Sentence, id = 3) are added at the beginning and end of each sentence. These special tokens function as sentence boundary markers, which are essential in transformer-based sequence-to-sequence models. The final output of this process is stored in pickle format to facilitate integration with the model training stage. The three main outputs generated are:

1) *x*, a list of audio spectrogram tensors;
2) *y*, a list of token ID sequences;
3) fnames, a list of audio file names.

The model training process consists of two stages: the encoder and the decoder. The first stage, as shown in Fig. 8, *b*, begins by loading the dataset in pickle format into memory and processing it using a data loader with padding on the Mel-spectrogram so that the time steps are uniform for each batch. Padding is also applied to the token sequences so that their lengths meet the model requirements.

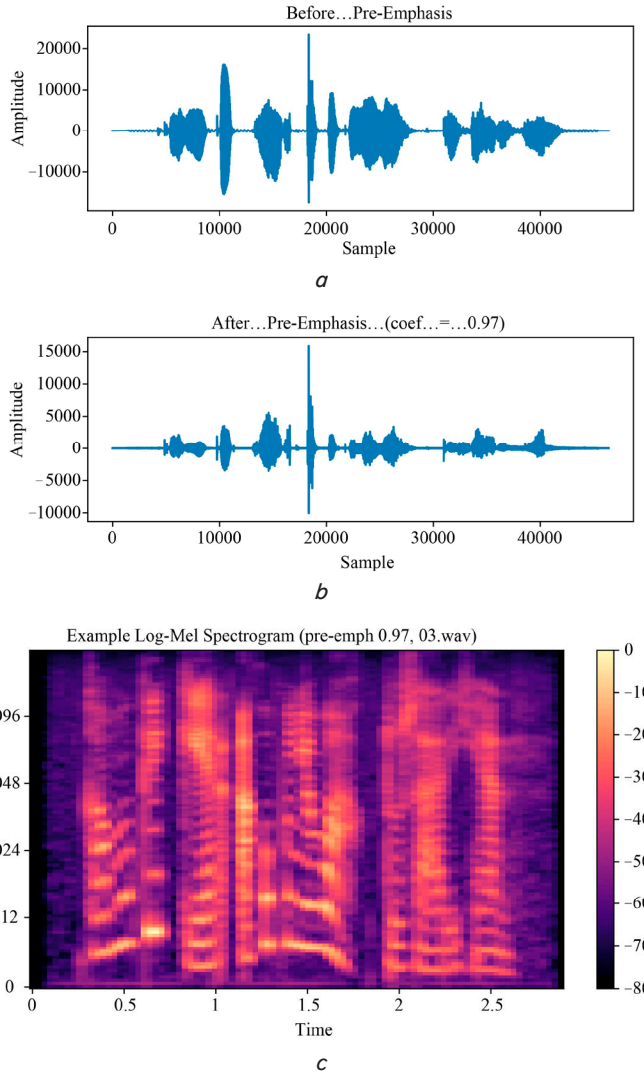
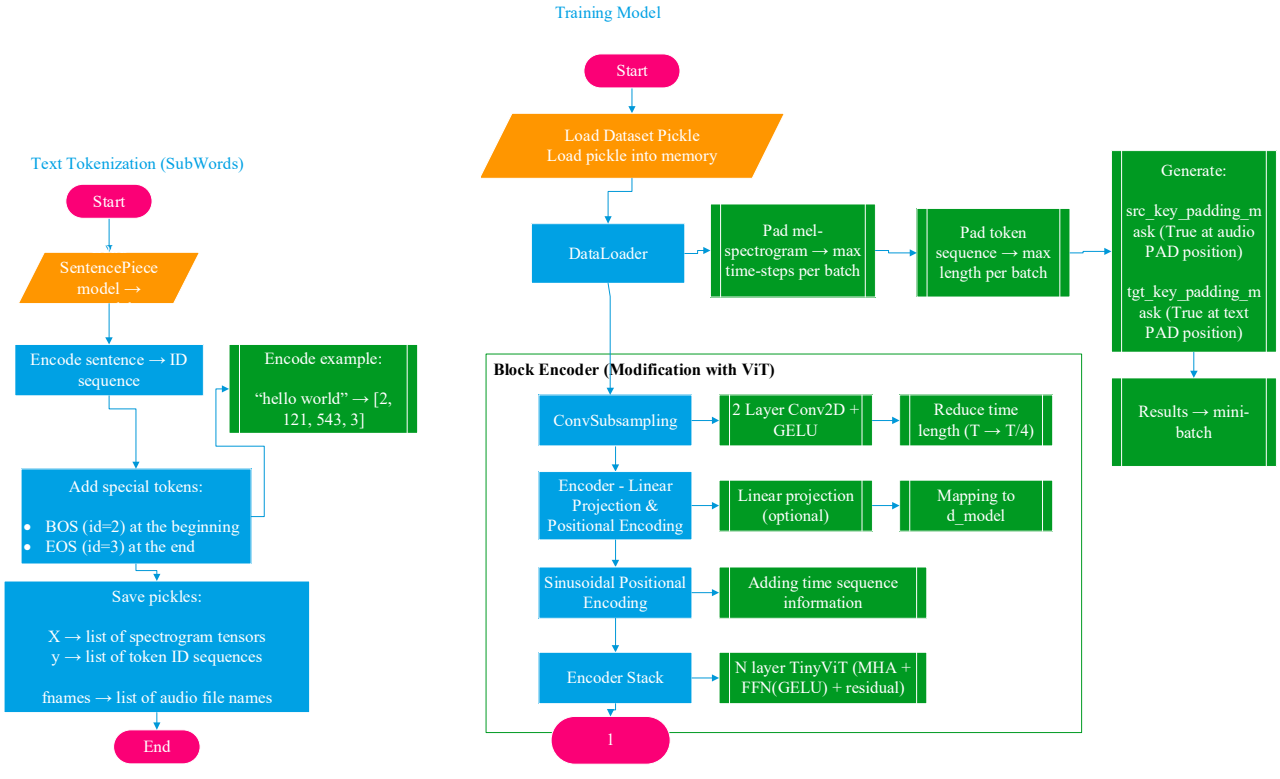


Fig. 7. Sample of speech signal: *a* — before pre-emphasis; *b* — after pre-emphasis with coef = 0.97; *c* — spectrogram input sample



a *b*

Fig. 8. Flowchart: *a* — the process of text tokenization using SentencePiece (SubWords); *b* — training process for the encoder block

Next, masking is created in the form of src_key_padding_mask to mark PAD positions in the audio data and tgt_key_padding_mask to mark PAD positions in the text. This process produces mini-batch data that is ready to be processed by the model. In the encoder stage, the audio data is reduced through ConvSubsampling, which uses two Conv2D layers with GELU activation, reducing the time dimension to one-fourth of its original length ($T \rightarrow T/4$). After that, a linear projection and sinusoidal positional encoding are applied to provide temporal sequence information. Positional encoding is calculated using the following equation:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{1000^{2i/d_{model}}}\right),$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{1000^{2i/d_{model}}}\right), \quad (1)$$

where pos – the token position, i – the dimension index, and d_{model} – the model dimension. The data is then fed into an encoder stack consisting of several TinyViT layers, which include multi-head attention (MHA), a feed-forward network (FFN) with GELU activation, and residual connection mechanisms.

At the decoder stage, as shown in Fig. 9, *a*, the target data is prepared in the form of a shifted target (tgt_in) and supplemented with a causal mask to prevent information leakage from future tokens.

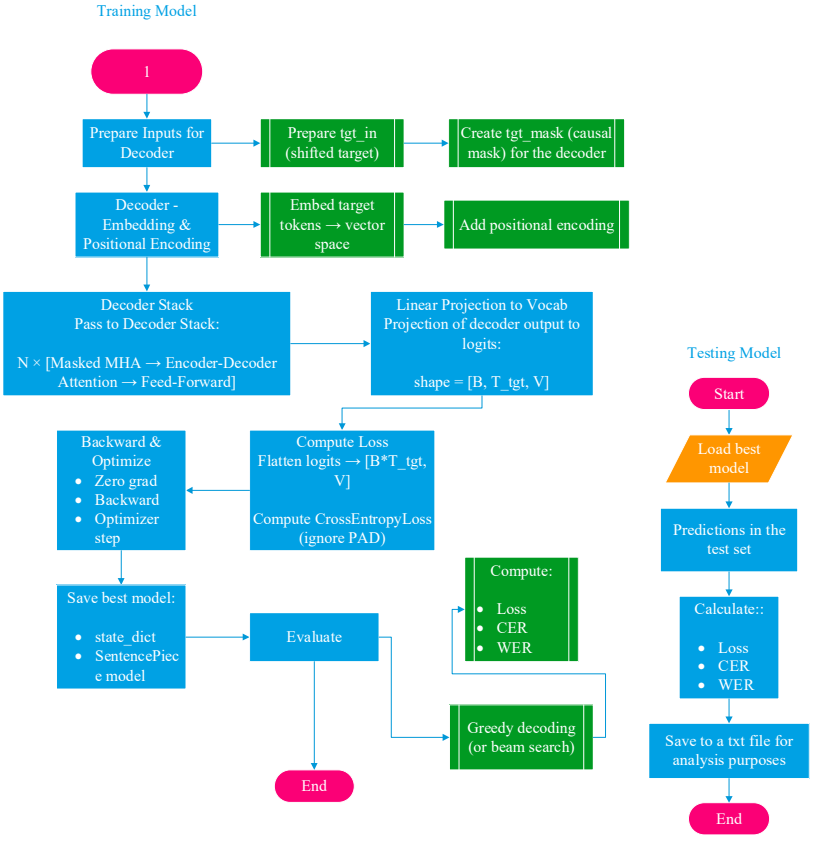


Fig. 9. Flowchart of the model modification: *a* — training process for decoders; *b* — the model testing process

The target tokens are embedded into a vector space, after which positional encoding is added. The data is then passed into the decoder stack, which consists of masked multi-head attention, encoder-decoder attention, and a feed-forward network. The main mechanism used is scaled dot-product attention, formulated as follows

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2)$$

where Q, K, and V – the query, key, and value of the linear transformation, respectively, and d_k is the dimension of the key.

The output from the decoder is linearly projected into the vocabulary space, producing logits with dimensions [B, $T_{t(t(t))}$, V]. The logit values are then normalized, and the loss function is calculated using Cross-Entropy Loss while ignoring the PAD token, as formulated in (3)

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i), \quad (3)$$

where y_i – the target label, $\hat{y}_i$ – the model's predicted probability, and N – the number of tokens considered.

The optimization process is carried out through the zero-gradient, backpropagation, and optimizer-step stages. The model with the best performance is saved as a state_dict, along with the SentencePiece model used. The final stage is the evaluation process, performed using greedy decoding or beam search. Model performance is measured using loss, character error rate (CER), and word error rate (WER), which respectively describe the prediction error at the character and word levels. Fig. 9, *b* shows the model testing conducted after the training process is completed to obtain the best model. This best model is reloaded into the system to evaluate the test set that has been separated since the beginning of the dataset formation process. At this stage, the model generates text transcription predictions from the test data. The prediction outputs are then analyzed using several key evaluation metrics, namely loss, CER, and WER. The loss value indicates the overall error rate of the model, while CER and WER measure transcription accuracy at the character and word levels. CER is calculated based on the number of substitution, insertion, and deletion errors compared to the reference label, as shown in (4), while WER calculates errors at the word level using (5):

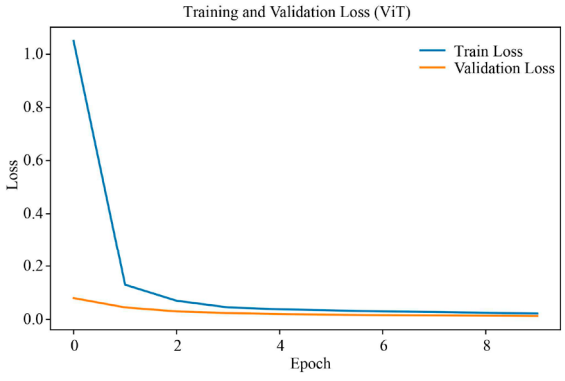
$$CER = \frac{S+D+I}{N}, \quad (4)$$

$$WER = \frac{S+D+I}{N}, \quad (5)$$

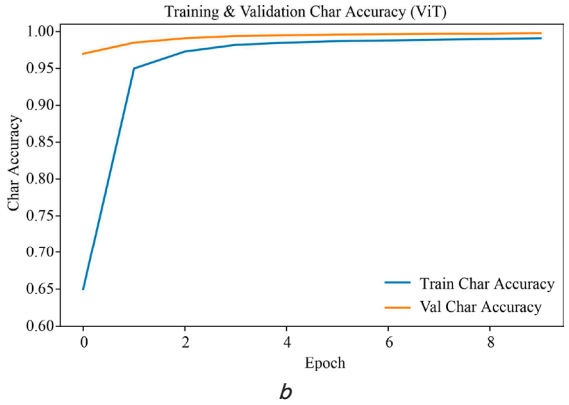
where S – the number of substitutions, D – the number of deletions, I – the number of insertions, and N – the number of reference words (character for *CER* and spoken word for *WER*). The results of this evaluation are then saved in a text file (.txt).

5. 2. Evaluation of vanilla and modified model training

The results of training the modified transformer architecture and the original vanilla model using the main dataset (NSS-ID) can be seen in Fig. 10, 11, respectively. The modified model outperforms the original vanilla transformer model. Based on these results, the modified model shows potential for further development.



a



b

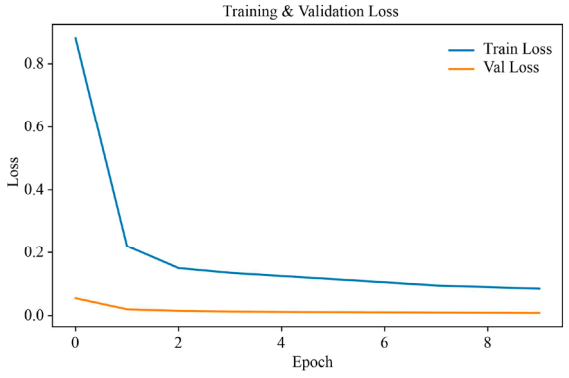
Fig. 10. Modified transformer model: *a* — training and validation loss; *b* — training and validation accuracy

Fig. 10, *a* shows the progression of training loss and validation loss during the model training process. Overall, the curves exhibit a stable convergence pattern, indicating consistent performance improvement as the number of epochs increases. At the beginning of training (epochs 0–1), the training loss decreases sharply from an initial value above 1.0 to around 0.25. This significant decline reflects rapid parameter adjustments during the early stages of optimization. In contrast, the validation loss begins at a relatively low value and decreases more gradually than the training loss, suggesting that the model is able to generalize reasonably well on the validation data from the outset. After the second epoch, both training and validation loss gradually decrease toward values close to 0.0. Fig. 10, *b* illustrates the increase in character-level accuracy for both the training and validation data of the modified model. Training accuracy rises sharply from around 0.65 in the early epochs to above 0.95 within just two epochs, while validation accuracy begins at a high value and quickly reaches a level close to 0.99.

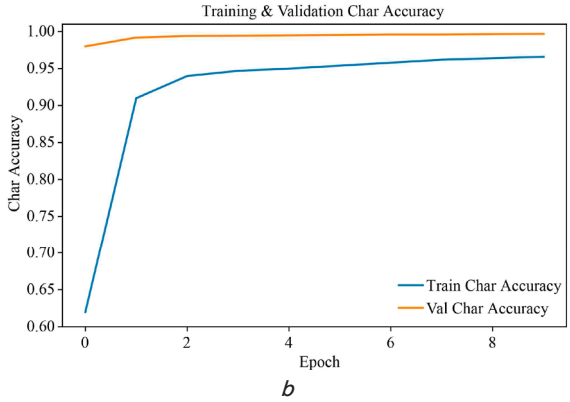
Fig. 11, *a* shows the training and validation loss throughout the vanilla model's training process. Fig. 11, *b* presents the training and validation character-level accuracy curves for the vanilla model, showing consistent performance improvement during training.

The results of the WER and CER comparison between the vanilla model and the Modified model can be seen in Table 4.

The WER and CER results for both models are shown in Fig. 12.



a



b

Fig. 11. vanilla transformer: *a* — training and validation loss; *b* — training and validation accuracy

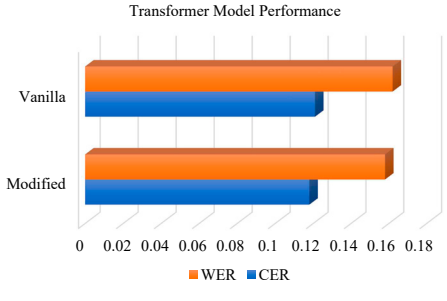


Fig. 12. Transformer model performance

Table 4

WER and CER results of the modified and vanilla transformer models

Model	CER	WER
Vanilla	0.121	0.162
Modified	0.118	0.158

In Table 4 dan Fig. 12, the WER value of the modified vanilla model is 0.158, while the WER value of the vanilla model is 0.162. Similarly, the CER value of the modified vanilla model is 0.118, compared to 0.121 for the vanilla model. Overall, these results show that the modified vanilla model achieves better WER and CER values than the vanilla model.

6. Discussion of results: comparison of vanilla and modified model performance using the NSS-ID dataset

As illustrated in Fig. 6, the initial audio signal undergoes a preprocessing stage prior to model training. Each speech signal is filtered using a pre-emphasis technique (Fig. 7) to enhance high-frequency components. SentencePiece BPE training is then performed according to the parameters specified in Table 1. Feature extraction is conducted using Mel-spectrogram representations based on the configuration presented in Table 2. The textual data are subsequently tokenized using SentencePiece. The encoder block is trained following the architecture shown in Fig. 8, with hyperparameters detailed in Table 3. Finally, as depicted in Fig. 9, the decoder stage is executed using the corresponding target sequences.

Fig. 10, *a* shown the consistency between the training and validation loss indicates that the model not only learns meaningful patterns from the training data but also maintains strong generalization capabilities. The convergence of both curves toward low values in the final epoch suggests that the training configuration including hyperparameters, optimization techniques, and regularization strategies functions effectively in achieving stable and optimal performance, demonstrating the model's strong potential for evaluation or further testing. In Fig. 10, *b*, the close alignment of the two curves, with only a very small gap up to the final epoch, indicates that the model learns effectively without overfitting. This convergence pattern shows that the architecture and training configuration successfully produce a model with excellent generalization capability for character-level data.

In the early epochs in Fig. 11, *a*, the training loss decreases sharply from a value close to 0.9 to around 0.2. Compared to the modified model, the vanilla model starts with a slightly lower initial training loss (≈ 0.9), indicating that ViT begins the optimization process from a more challenging condition (> 1.0). After the second epoch, the decline in training loss slows but remains consistent, reaching a value close to 0.1 at the end of training. When the two models are compared, both exhibit a similar loss-reduction pattern an initial sharp decline followed by stable convergence in later epochs. However, the modified model demonstrates more aggressive training behavior, evident from its dramatic drop to a very low loss value within the first two epochs. Meanwhile, the validation loss in the vanilla model begins at a lower value and shows a steady decline before converging near 0.05. In this respect, both models achieve low and stable final values, but the ViT model maintains a smaller gap between the training and validation loss, indicating slightly better generalization compared to the vanilla model. Training accuracy increases in Fig. 11, *b* sharply in the early epochs, rising from around 0.65 to above 0.90 by the second epoch, then continues to increase gradually to around 0.97. Meanwhile, validation accuracy begins at a higher value approaching 0.98 and remains stable throughout the training process.

In Fig. 12, the WER and CER of the modified transformer improve by up to 2% compared to the original vanilla model. These results indicate that the modified vanilla transformer is able to recognize entire words more accurately than the baseline model. The modified architecture incorporates a multi-layer perceptron (MLP) with greater flexibility in its layer structure compared to the standard feed-forward layers in the original transformer, allowing backpropagation to minimize loss more effectively. In addition, the reduction in WER achieved by the modified model suggests that the architectural enhancements more effectively support word-level comprehension in the ASR process. The convolutional layers help stabilize local acoustic patterns and emphasize phoneme boundaries, enabling the model to form clearer representa-

tions of word units. In parallel, the ViT components strengthen the model's ability to capture long-range dependencies across the Mel spectrogram through patch-based global attention, allowing the system to recognize entire word structures more consistently, even under variations in speaking rate or pronunciation. When these enriched features are processed by the transformer's attention mechanisms, the model gains a stronger contextual understanding of how characters combine into words, reducing common errors such as substitutions, deletions, and fragmented outputs. As a result, the modified architecture demonstrates improved lexical accuracy and overall robustness compared to the baseline vanilla transformer.

This study successfully introduced a new transformer architecture model by modifying the encoder with the integration of convolutional and vision transformer (ViT) blocks, then comparing the results with the baseline model, namely the original transformer. This study also used an Indonesian language sound dataset as the primary dataset, which was recorded by the researchers themselves to be trained on this modified architecture model. This differs from studies [13, 14], which successfully implemented transformer-based architecture only through pre-training models. It also differs from studies [8, 12, 15, 18, 19], which optimized the accuracy of transformer models for text translation using secondary datasets in the form of text data.

The advantage of this study is that the dataset used is voice data recorded by the researchers themselves in a special room, consisting of 80,000 wav format voice data files. This is different from study [9], which used a dataset in the form of news articles, and study [13], which used a Wikipedia corpus dataset.

This study is limited to the use of the NSS-ID dataset as the main data source and character-level metric-based evaluation in a controlled experimental environment. Variations in dialect, noise conditions, and more complex acoustic scenarios are not yet fully represented, so the model's ability to generalize to real-world conditions still requires further validation. Furthermore, the evaluation does not yet include a comprehensive analysis of word error rate (WER), semantic aspects at the word level, or performance testing in real-time two-way interaction scenarios that take into account latency and computational efficiency. Model comparisons are also still limited to vanilla transformer as a baseline without involving the latest ASR architecture.

The proposed model shows potential for development in the implementation of real-time two-way interaction systems, multi-dialect adaptation, multimodal integration, and optimization for devices with limited resources. In addition, this architecture opens up opportunities for expansion into large-scale pre-training approaches and the development of domain-specific applications in the Indonesian language ASR ecosystem.

7. Conclusion

1. This study successfully developed an automatic speech recognition (ASR) system based on the transformer architecture with a modified encoder that incorporates convolutional and vision transformer (ViT) blocks:

a) the primary Indonesian language dataset was processed through audio preprocessing and text tokenization using SentencePiece with byte pair encoding (BPE), and then divided into training, validation, and testing sets with a ratio of 80:10:10. The evaluation results show that the modified model achieves lower WER and CER values than the original vanilla model, with a word error rate (WER) of 0.158 and a character error rate (CER) of 0.118. Meanwhile, the original vanilla model used as a baseline produces a WER of 0.162 and a CER of 0.121;

b) These results confirm that the proposed architecture has strong potential for further development, as its performance surpasses that of the original vanilla model based on the WER and CER comparisons.

2. This study proves that the significantly modified transformer architecture outperforms the original vanilla transformer model in Indonesian automatic speech recognition tasks using the NSS-ID dataset. The proposed model shows faster training convergence, consistent reduction in training loss and validation loss, and an increase in character-level accuracy that exceeds 0.95 on the training data and approaches 0.99 on the validation data with minimal generalization gap. These results indicate stable optimization dynamics and strong generalization capabilities from the early stages of training. The main characteristics, namely accelerated convergence, learning curve stability, and high character precision, are significant differentiators compared to the baseline model. These advantages contribute directly to addressing the challenge of developing a more adaptive transformer architecture for Indonesian, particularly in reducing the gap between experimental evaluation and practical implementation. With the support of a representative Indonesian speech dataset, the proposed model has the potential to be applied to real-time two-way interaction systems. Overall, this study not only improves the performance of the transformer baseline but also provides relevant empirical contributions to the development of a more reliable and implementable Indonesian ASR system.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing

The research/publication of this article was funded by the Directorate of Research and Community Service, Directorate General of Research and Development, Ministry of Higher Education, Science, and Technology, under Research Contract Number: 109/C3/DT.05.00/PL/2025.

Data availability

Data will be made available on reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies in creating the submitted work.

Acknowledgments

The authors would like to thank the Directorate of Research and Community Service, Directorate General of Research and Development, Ministry of Higher Education, Science, and Technology for the funding, as well as Universitas Sriwijaya and the State Polytechnic of Sriwijaya for their support during this study.

Authors' contributions

Ratna Atika: Writing – original draft, Methodology, Data curation, Formal analysis, Investigation, Visualization; **Suci Dwijayanti**: Writing – review and editing, Methodology, Formal analysis, Conceptualization; **Bhakti Yudho Suprapto**: Writing – review and editing, Formal analysis.

References

1. Loubser, A., De Villiers, P., De Freitas, A. (2024). End-to-end automated speech recognition using a character based small scale transformer architecture. Expert Systems with Applications, 252, 124119. https://doi.org/10.1016/j.eswa.2024.124119
2. Ro, J. H., Stahlberg, F., Wu, K., Kumar, S. (2022). Transformer-based Models of Text Normalization for Speech Applications. arXiv. https://doi.org/10.48550/arXiv.2202.00153
3. Alastruey, B., Gállego, G. I., Costa-jussà, M. R. (2021). Efficient Transformer for Direct Speech Translation. arXiv. https://doi.org/10.48550/arXiv.2107.03069
4. KHu, K., Pang, R., Sainath, T. N., Strohman, T. (2021). Transformer Based Deliberation for Two-Pass Speech Recognition. 2021 IEEE Spoken Language Technology Workshop (SLT), 68–74. https://doi.org/10.1109/slt48900.2021.9383497
5. Le, P.-H., Gong, H., Wang, C., Pino, J., Lecouteux, B., Schwab, D. (2023). Pre-training for Speech Translation: CTC Meets Optimal Transport. arXiv. https://doi.org/10.48550/arXiv.2301.11716
6. Ahmadian, H., Abidin, T. F., Riza, H., Muchtar, K. (2023). Transformer-Based Indonesian Language Model for Emotion Classification and Sentiment Analysis. 2023 International Conference on Information Technology and Computing (ICITCOM), 209–214. https://doi.org/10.1109/icitcom60176.2023.10442970
7. Heryadi, Y., Wijanarko, B. D., Fitria Murad, D., Tho, C., Hashimoto, K. (2022). The Effect of Encoder and Decoder Stack Depth of Transformer Model to Performance of Machine Translator for Low-resource Languages. Proceedings of the International Conference on Industrial Engineering and Operations Management, 2766–2776. https://doi.org/10.46254/ap03.20220479
8. Heryadi, Y., Wijanarko, B. D., Fitria Murad, D., Tho, C., Hashimoto, K. (2023). Revalidating the Encoder-Decoder Depths and Activation Function to Find Optimum Vanilla Transformer Model. 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), 162–167. https://doi.org/10.1109/iccosite57641.2023.10127790
9. Sonata, I., Heryadi, Y., Tho, C. (2023). Topic Segmentation using Transformer Model for Indonesian Text. Procedia Computer Science, 227, 159–167. https://doi.org/10.1016/j.procs.2023.10.513
10. Suyanto, S., Arifianto, A., Sirwan, A., Rizaendra, A. P. (2020). End-to-End Speech Recognition Models for a Low-Resourced Indonesian Language. 2020 8th International Conference on Information and Communication Technology (ICoICT), 1–6. https://doi.org/10.1109/icoict49345.2020.9166346
11. Sonata, I. (2023). Automatic Speech Recognition in Indonesian Using the Transformer Model. 2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS), 263–266. https://doi.org/10.1109/icimcis60089.2023.10349042
12. Wijanarko, B. D., Fitria Murad, D., Heryadi, Y., Tho, C., Hashimoto, K. (2023). Exploring the Effect of Activation Function on Transformer Model Performance for Official Announcement Translator from Indonesian to Sundanese Languages. 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), 827–831. https://doi.org/10.1109/iccosite57641.2023.10127770
13. Wongso, W., Setiawan, D. S., Suhartono, D. (2021). Causal and Masked Language Modeling of Javanese Language using Transformer-based Architectures. 2021 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 1–7. https://doi.org/10.1109/icacsis53237.2021.9631331
14. Fuadi, M., Wibawa, A. D., Sumpeno, S. (2023). idT5: Indonesian Version of Multilingual T5 Transformer. arXiv. https://doi.org/10.48550/arXiv.2302.00856
15. Musyafa, A., Gao, Y., Solyman, A., Wu, C., Khan, S. (2022). Automatic Correction of Indonesian Grammatical Errors Based on Transformer. Applied Sciences, 12 (20), 10380. https://doi.org/10.3390/app122010380
16. Fudholi, D. H., Nayoan, R. A. N. (2022). The Role of Transformer-based Image Captioning for Indoor Environment Visual Understanding. International Journal of Computing and Digital Systems, 12 (3), 479–488. https://doi.org/10.12785/ijcds/120138
17. Aditya Rachman, A., Suyanto, S., Rachmawati, E. (2021). Leveraging CNN and Bi-LSTM in Indonesian G2P Using Transformer. 2021 13th International Conference on Machine Learning and Computing, 161–165. https://doi.org/10.1145/3457682.3457706
18. Sirwan, A., Thama, K. A., Suyanto, S. (2022). Indonesian Automatic Speech Recognition Based on End-to-end Deep Learning Model. 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 410–415. https://doi.org/10.1109/cyberneticscom55287.2022.9865253
19. Warto, Muljono, Purwanto, Noersasongko, E. (2023). Improving Named Entity Recognition in Bahasa Indonesia with Transformer-Word2Vec-CNN-Attention Model. International Journal of Intelligent Engineering and Systems, 16 (4), 655–668. https://doi.org/10.22266/ijies2023.0831.53
20. Hutama, L. B., Suhartono, D. (2022). Indonesian Hoax News Classification with Multilingual Transformer Model and BERTopic. Informatica, 46 (8). https://doi.org/10.31449/inf.v46i8.4336
21. Lin, T., Wang, Y., Liu, X., Qiu, X. (2022). A survey of transformers. AI Open, 3, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001
22. Xu, P., Zhu, X., Clifton, D. A. (2023). Multimodal Learning With Transformers: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45 (10), 12113–12132. https://doi.org/10.1109/tpami.2023.3275156