DOI: 10.31319/2519-8106.1(52)2025.325954
UDC 004.657

Yalova Kateryna, Candidate of Technical Sciences, Associate Professor, Head of the Department of the Systems Software
Ялова К.М., кандидат технічних наук, доцент, кафедра програмного забезпечення систем
ORCID: 0000-0002-2687-5863
e-mail: yalovakateryna@gmail.com

Babenko Mykhailo, Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of the Systems software
Бабенко М.В., кандидат технічних наук, доцент, кафедра програмного забезпечення систем
ORCID: 0000-0003-1013-9383
e-mail: mvbab@ukr.net

Sheliuh Kostiantyn, PhD student, Department of the Systems software
Шелюг К.Ю., здобувач третього (доктора філософії) рівня вищої освіти, кафедра програмного забезпечення систем
email: kostia902@ukr.net

Dniprovsky State Technical University, Kamianske
Дніпровський державний технічний університет, м. Кам’янське

APPLICATION OF NEURAL NETWORKS IN THE TASK OF SPEECH RECOGNITION

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАДАЧІ РОЗПІЗНАВАННЯ МОВЛЕННЯ

The article is dedicated to describing a generalized neural network approach to solving the scientific and practical problem of speech recognition. The algorithm presented in the article for utilizing neural networks to transform an input audio signal into recognized text outlines the key steps in modeling and implementing a speech neural network model. The study provides a mathematical description and architectural overview of the three most effective types of neural networks that can be used in the development of automatic speech recognition systems: recurrent neural networks, convolutional neural networks, and transformer-based networks. For each of these neural network types, the article presents an implementation workflow for speech recognition tasks, along with graphical representations of their architectures. Additionally, the advantages and disadvantages of each neural network type are analyzed, and a comparative evaluation of expected recognition results is provided, including accuracy, computational complexity, memory requirements, and key performance metrics such as Word Error Rate, Character Error Rate, and BLEU score.

***Keywords**: speech recognition; neural network; recurrent neural network; convolution neural network; transformer-based neural network.*

Статтю присвячено опису узагальненого нейромережевого підходу до розв’язання завдання розпізнавання мовлення. Наведений у статті алгоритм використання нейронних мереж для перетворення вхідного аудіосигналу на розпізнаний текст описує основні кроки моделювання та програмної реалізації мовної нейромережевої моделі, такі як: збір даних, їх попередня обробка, виділення ознак, вибір та навчання моделі, декодування та впровадження у практичні системи.

У роботі представлено математичний опис та архітектури трьох найбільш ефективних типів нейронних мереж, які можуть бути використанні під час розробки системи автоматичного розпізнавання мовлення: рекурентні і згорткові нейронні мережі та мережі типу «трансформер», для яких представлено опис кроків їх впровадження в задачі розпізнавання мовлення із математичною формалізацією цього опису. Наведені графічні представлення архітектур нейронних мереж дають змогу наочно оцінити складність їх структури та ілюст-

рують схему перетворення вхідної послідовності до результуючої за допомогою специфічних програмних механізмів.

Для кожного типу нейронних мереж визначено переваги і недоліки їх використання та наведено порівняльна характеристика очікуваних результатів розпізнавання мовлення: точність, обчислювальна складність, вимога пам'яті, критерії WER, CER, BLEU. Визначено, що рекурентні нейронні мережі вимагають менше обчислювальних ресурсів, що робить їх оптимальними для застосування на невеликих наборах даних та у задачах з низькою складністю. Згорткові нейронні мережі є потужним інструментом для витягування акустичних ознак, забезпечуючи високу швидкість обчислень завдяки паралелізації, однак для врахування часової динаміки їх зазвичай комбінують з іншими нейронними мережами. У свою чергу, трансформерні архітектури демонструють найвищу точність розпізнавання мовлення завдяки здатності ефективно обробляти довгі послідовності, проте вони мають високу обчислювальну складність та великі вимоги до ресурсів і розміру вхідної послідовності.

Представлені результати дослідження можуть бути застосовані для обґрунтованого вибору типу нейронної мережі під час реалізації системи автоматичного розпізнавання мовлення.

***Ключові слова**: розпізнавання мовлення; нейронна мережа; рекурентні нейронні мережі; згорткові нейронні мережі; нейронні мережі типу «трансформер».*

Problem's formulation

Automatic speech recognition (ASR) is a relevant scientific and practical challenge within the field of natural language processing. The ultimate goal of solving this problem is to convert an input acoustic signal into text. Implementing ASR is a complex task, as it heavily depends on the quality of the acoustic signal, which is particularly critical for real-time speech recognition systems. The primary requirements for ASR systems include recognition accuracy, robustness, processing speed, and scalability. These requirements shape the system's key characteristics, such as noise resistance, minimal speaker dependency, low latency between speech input and output, minimal recognition errors, multilingual support, and adaptability to new words.

The methods used for ASR have been modernized in response to the increasing demands of the systems that implement them [1]. Template matching methods were the first approaches applied to ASR. These methods analyzed the acoustic characteristics of the input signal and relied on a predefined set of templates for each word or phoneme. Early ASR systems based on template matching methods were inefficient and had a significant drawback — their vocabulary was limited to a predefined set of words available for recognition.

The next approach applied to ASR involved statistical methods and stochastic models, particularly hidden Markov models (HMM). The primary advantage of this method over its predecessor was the ability to operate with an unlimited vocabulary. This was achieved through the introduction of the phoneme-based speech recognition concept and the computation of word occurrence probabilities within a given context.

The development of neural networks (NN) and artificial intelligence (AI) has also significantly impacted ASR tasks. The primary motivation for applying NNs and deep learning was to enhance recognition accuracy, efficiently process audio signals, model acoustic features, and recognize continuous speech [2]. End-to-end models based on NNs learn directly from audio files without relying on separate models. Additionally, transformer-based NNs enable speech analysis in complex linguistic contexts without requiring manual annotation in datasets.

The NN approach introduces new opportunities for implementing ASR systems, ensuring efficient and accurate recognition, which is widely applied in voice assistants, transcription systems, and interactive human-machine interfaces [3]. Despite significant advancements in speech recognition quality, this task remains a subject of ongoing research aimed at developing even more effective algorithms, models, methods, and architectures.

Analysis of recent research and publications

The application of NNs in ASR represents a modern, efficient, and highly accurate approach that has significantly improved recognition accuracy and the ability to process large volumes of data. Despite substantial progress in this field, ASR remains a challenging task due to linguistic diversity, accents, background noise, and the stringent requirements for both speed and accuracy. Current research in this area focuses on optimizing NNs architectures, enhancing their performance — particularly for real-time speech recognition and low-resource languages — and reducing model size, which is critical for embedded and mobile ASR systems.

The analysis of recent scientific studies has shown that the most common types of NNs used for speech recognition are recurrent neural networks (RNN), convolutional neural networks (CNN), and transformer-based NNs. Numerous research papers are dedicated to the development of effective NN approaches for ASR.

Studies [3—5] describe methods and ideas for improving recognition accuracy in RNNs, while studies [6—7] focus on the use of BiLSTM for sequence annotation and the attention mechanism in interactive speech recognition.

The authors of studies [8—10] have dedicated their research to developing efficient approaches for applying NNs to the recognition of specific languages, particularly low-resource languages, with an emphasis on continuous speech recognition.

Studies [11—12] present novel approaches to using CNNs for speech recognition. In particular, the authors of study [11] propose using video sequences of speech, rather than audio input, by extracting information from lip movements.

The application of transformer-based NNs is described in studies [13—15], where the authors utilize the classical transformer architecture and modify it to enhance recognition accuracy or enable ASR under specific conditions. In study [13], the authors integrate a CNN into the transformer model to improve recognition quality by leveraging the CNN's strong feature extraction capabilities. Study [14] proposes adapting a transformer-based model for self-supervised speech recognition using unlabeled audio data. Meanwhile, study [15] explores the use of transformer-based models for language identification, speech-to-text conversion, and real-time subtitle generation.

Research in the field of NNs for speech recognition continues to evolve, revealing untapped potential for refining existing NN approaches and developing new models that can enhance the quality of ASR systems, bringing them closer to human-like speech processing.

Formulation of the study purpose

The objective of this paper is to present a generalized approach to the application of NNs in speech recognition tasks. To achieve this goal, the following research tasks were identified and implemented: analysis of the architecture and mathematical description of NNs, including RNN, CNN, and Transformers; development of algorithms for applying the selected NN models to speech recognition tasks; formation of a comparative analysis of the NNs and conclusions regarding the justified selection of an optimal NN architecture for speech recognition.

Presenting main materials

In speech recognition tasks, NNs of various architectures can be applied. However, a generalized algorithm for their use can be formulated, describing the main stages:

1. Data Collection. First, a large dataset of audio recordings must be prepared. For most NNs, it is necessary to provide a corresponding textual annotation for each audio recording.

2. Preprocessing. The entire input dataset must undergo signal preprocessing, including noise removal, volume normalization, and overall sound quality enhancement [16]. To make the input audio suitable for NN processing, it should be converted into a spectrogram or a set of Mel-Frequency Cepstral Coefficients (MFCC) [17].

3. Feature Extraction. The input signal is analyzed and transformed into a time-frequency representation, for example, using Fourier transforms or wavelet transformations.

4. Model Selection. The choice of NN architecture largely depends on the requirements of the speech recognition system and the expected output format, which ultimately affects both performance and recognition accuracy.

5. Model Training. The designed NN must be trained to recognize speech using a dataset containing paired examples of audio signals and their corresponding textual transcriptions. Performance analysis and evaluation of the obtained results help select the most optimal network parameters while preventing overfitting.

6. Decoding. The output of the NN consists of probability distributions that need to be

decoded and converted into structured text sequences. To enhance the final speech recognition output, statistical models or additional NN language models are often applied.

7. Model Evaluation. The trained NN undergoes testing to assess its effectiveness and accuracy.

8. Deployment. The final step in applying NNs for speech recognition involves integrating the trained model into the ARS system and deploying it on designated resources, such as cloud platforms or resource-constrained edge devices. The performance of the deployed model can be optimized using hardware acceleration, parallelization, model pruning, or quantization techniques.

All NNs architectures ultimately transform an input audio signal into a textual sequence represented as a set of characters, phonemes, or tokens. However, each type of NN is characterized by a specific architecture and has distinct requirements for the input dataset [3]. RNNs ensure sequential processing of speech fragments [5, 18], CNNs are used when the input signal is provided in the form of spectrograms [11, 12], and for complex speech recognition tasks, transformer-based NNs are the most suitable [13—15].

RNNs are a type of NNs designed to process sequential speech fragments [2, 16]. A schematic representation of the generalized RNN architecture is shown in Fig. 1.

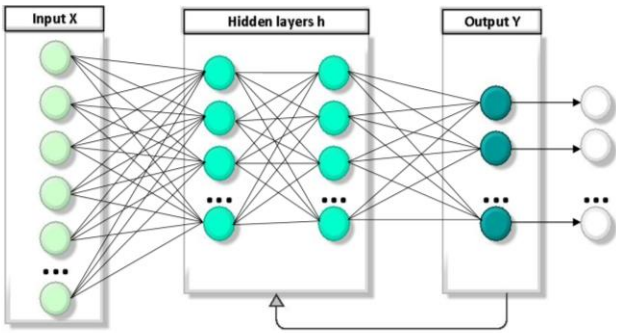

Fig. 1. Generalized architecture of RNNs

Then, the algorithm for applying RNNs in the context of speech recognition task consists of the following steps:

1. Creating the input sequence X=(x_1, x_2,...,x_n) of feature vectors, such as those represented using MFCC or Mel-spectrograms, where n is the total length of the input sequence.

2. Processing the speech signal at each time step while preserving the context of previous states. The calculation of the weight coefficients in the RNN takes into account the hidden states at each time step t, which is determined as:

$$h_t = f\left(W_h h_{t-1} + W_x x_t + b_h\right), \quad (1)$$

where h_t — is the hidden state vector, W_h — is the weight matrix for the previous state, W_x — is the weight matrix for the current input vector, b_h — is the bias vector, f — is the activation function, typically tanh or ReLU.

3. Transforming hidden states into probabilities of recognized symbols or phonemes in the Softmax layer, which is mathematically expressed as:

$$y_t = softmax\left(W_y h_t + b_y\right), \quad (2)$$

where W_y — is the weight matrix for the output, b_y — is the bias vector.

4. Decoding the output sequence of symbols or words Y = (y_1, y_2,...,y_m) into the final text using various algorithms, where $y_m \in V$ — represents a symbol or word from the vocabulary V. For this purpose, techniques such as Greedy Decoding (selecting symbols with the highest probability), Beam Search Decoding (computing the best possible text variants), or Connectionist Temporal Classification (enabling speech recognition without precise alignment of audio and text) are commonly used.

5. Post-processing and text normalization to remove repeated symbols, correct errors, and introduce punctuation marks for structuring the final sentences.

The most widely used and modern RNN models suitable for ASR are Long Short-Term

Memory (LSTM) and Bidirectional LSTM (BiLSTM). LSTM and BiLSTM are types of RNNs with long-term memory, which effectively process sequential data and long-range dependencies, providing moderate expected accuracy [5—7]. However, they have several significant limitations:

- high computational complexity due to sequential processing;
- limited parallelization capabilities;
- information loss in long sequences, which is partially mitigated in BiLSTM;
- complex training process.

CNNs are traditionally used for image analysis but are also effectively applied to speech recognition, particularly when processing time-frequency representations of audio signals (e.g., spectrograms or MFCCs). CNNs excel at extracting local features, improving robustness to noise and speech variability. The primary concept of using CNNs in speech recognition is leveraging convolutional layers to extract high-level features from the input audio signal [8]. A schematic representation of the generalized CNN architecture is shown in Fig. 2.

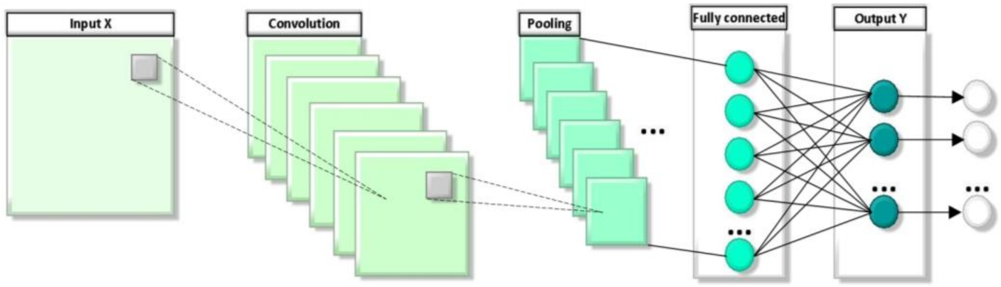

Fig. 2. Generalized architecture of CNNs

The generalized algorithm for applying CNNs to speech recognition consists of the following steps:

1. Formation of the input dataset *X*, typically represented as an acoustic feature matrix *T*×*F*, where *T* — is the number of time frames, and *F*— is the number of acoustic features.

2. Performing the core CNN operation — convolution, which is applied using a set of filters. Mathematically, the convolution operation in CNN can be described as:

$$Z_{i,j,k} = \sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{m=1}^{N} W_{k,c,m,n} \cdot X_{c,i+m,j+n} + b_k \,, \tag{3}$$

where $Z_{i,j,k}$ — is the value of the output tensor after convolution, $W_{k,c,m,n}$ — is the convolution kernel for the *k*-th filter of size *M*×*N*, $X_{c,i+m,j+n}$ — is the local patch of the input signal, and b_k — is the bias of the *k*-th filtr.

3. Applying the non-linear activation function ReLU:

$$A_{i,j,k} = \max(0, Z_{i,j,k}). \tag{4}$$

4. Applying the pooling operation at the pooling layers, which reduces the dimensionality of the features and enhances resistance to variations in the input data. Mathematically, the pooling operation is described as:

$$P_{i,j,k} = \max_{(m,n)\in R} A_{i+m,j+n,k} \,, \tag{5}$$

where *R* is the pooling window.

5. Processing the feature vector in the fully connected layer for classification of speech features.

6. Applying the Softmax function to determine the probabilities of characters, words, or phonemes.

7. Decoding the resulting text, for example, using Beam Search — a heuristic search algorithm for finding the most probable sequence or Viterbi decoding — a dynamic programming algorithm for finding the most probable sequence in HMM.

8. Error correction and formatting the resulting sequence of tokens $Y = (y_1, y_2, ..., y_m)$.

Despite their ability to reduce data dimensionality, high robustness to noise, and fast training, CNNs have certain drawbacks when applied to speech recognition, namely:

– the inability to preserve the context of previous steps, which is critically important for speech;

– the need for a large amount of training data;

– the reduction in detail in the input spectrograms of the audio signal due to the use of pooling layers, leading to the loss of valuable information;

– poor adaptability to processing audio signals of variable length.

Transformers are the modern standard in ASR, demonstrating high accuracy and efficiency [2]. Showing significantly higher accuracy, they have replaced RNNs and CNNs in many modern ASR systems, bringing recognition quality closer to human-level accuracy. The main advantages of transformers are their ability to account for long-term context, high robustness to noise and speaker-specific pronunciation, and support for multilingual models. The use of transformer-based models (Whisper, Wav2Vec2, Conformer) has opened new possibilities for voice assistants, automatic subtitles, and other applications. A schematic of the generalized architecture of transformer-based NNs is shown in Fig. 3.

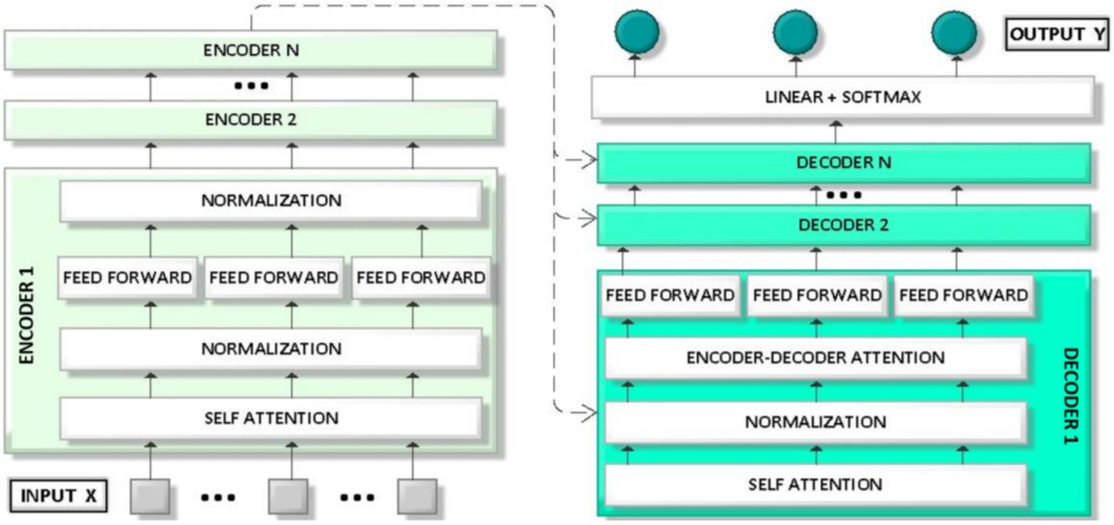

Fig. 3. Generalized architecture of transformer-based NN

Assuming that $X = (x_1, x_2, \dots, x_n)$ is the input feature sequence of the audio signal, which is transformed into the output token sequence $Y = (y_1, y_2, \dots, y_m)$, the generalized algorithm for applying transformers to solve the speech recognition task consists of the following main steps:

1. Feature extraction of the input audio signal and positional encoding (PE) of each element in the input sequence:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right), \qquad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right), \tag{6}$$

where *pos* — is the position of each element in the sequence, and *i* — is the index of the component.

2. Calculation of Self-attention. For each element in the input sequence, the relationship with other elements is computed. The attention matrix is computed as:

$$A = soft\max\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{7}$$

where $Q=XW_Q$, $K= XW_K$, $V= XW_V$ — are the query, key, and value matrices, respectively, d — is the feature dimensionality, and W_Q , W_K, W_V — are the weight matrices.

3. Computation through the feed-forward network:

$$FFN(x) = \mathrm{Re}\,LU(xW_1 + b_1)W_2 + b_2\,. \tag{8}$$

4. Decoding the output sequence using CTC-loss, Beam Search, or Greedy Decoding.

5. Formation of the resulting output text.

The most commonly used transformer-based NNs applied to ASR are Wav2Vec2 (a NN that implements the basic functionality of the transformer attention mechanism), Whisper (a NN used not only for ASR but also for audio translation between languages), and Conformer (a hybrid NN that combines CNN and transformer mechanisms to improve recognition accuracy). The ability of transformer-based NNs to capture complex linguistic dependencies makes this type of NN effective in speech processing and recognition tasks. The main drawbacks of transformers are:

– high computational complexity;
– high memory requirements;
– the need for a large annotated input data corpus;
– delays in processing due to the complexity of computations.

To evaluate the effectiveness of speech recognition using NNs, metrics such as Word Error Rate (WER), Character Error Rate (CER), and Bilingual Evaluation Understudy (BLEU) Score are commonly used [2, 4]. WER is the primary metric that determines the percentage of incorrect words in the recognized text and is calculated as:

$$WER = \frac{S + D + I}{N}, \tag{9}$$

where S — is the number of substituted words, D — is the number of deleted words, I — is the number of inserted words, N — is the total number of words in the reference text.

CER is a metric for evaluating the quality of speech recognition systems, determining the proportion of errors at the character level. It is calculated using the same formula as WER but at the character level instead of words. CER measures the extent to which a speech recognition system distorts the text at the character level. It is often used in conjunction with WER.

If WER and CER determine the level of distortion in the obtained results relative to the reference, the BLEU metric is used to evaluate the quality of automatic machine translation by comparing the translated text with one or more reference (human) translations. BLEU measures the accuracy of n-gram matches between the generated text and the reference translation. It is defined as:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{11}$$

where p_n — presents the n-gram precision (the number of matched n-grams divided by the total number of n-grams in the generated text), w_n — are the weighting coefficients, BP is the brevity penalty applied for excessively short translations.

BLEU evaluates the quality of automatic translation by measuring the match of phrases of varying lengths with the reference translation.

Tabl. 1 presents the results of the speech recognition performance analysis for RNNs, CNNs, Whisper, Wav2Vec2, and Conformer transformer-based NN models.

Table 1. Expected speech recognition quality

Model	Accuracy%	WER (%)	CER (%)	BLEU	Computational complexity	Memory requirements
LSTM	Up to 85—90	12—25	5—15	0,50—065	$O(n \cdot d^2)$	$O(n \cdot d)$
BiLSTM	Up to 88—92	10—22	4—12	0,55—0,70	$O(n \cdot d^2)$	$O(n \cdot d)$
CNN	Up to 90—94	6—10,5	4—7.97	60—80	$O(n \cdot k \cdot d^2)$	$O(n \cdot k \cdot d)$
Wav2Vec2	Up to 95	Up to 5.0	2.5	-	$O(n \cdot d^2)$	$O(n \cdot d)$
Whisper	Up to 94	6.0	3.0	-	$O(n^2 \cdot d)$	$O(n^2 \cdot d)$
Conformer	Up to 95	4.5	2	-	$O(n \cdot d^2 + n^2 \cdot d)$	$O(n \cdot d + n^2)$

The data presented in Tabl. 1 illustrate the range of average values for recognition accuracy and the WER, CER, and BLEU criteria determined for RNN, CNN, and transformer-based NNs of various architectures, trained on datasets of different sizes and for different languages. The studies, values, and descriptions of these models are provided in [2, 6, 8, 13–15, 19]. For NNs such as LSTM, BiLSTM, CNN, Wav2Vec2, and Conformer, computational complexity depends on the sequence length and the dimensionality of the hidden state or feature space d. Each processing step has quadratic computational complexity, and the data in Table 1 illustrate how the performance of NNs varies depending on input sequence size and architectural specifics. Since Wav2Vec2 employs a CNN, its dependence on input size is linear, whereas for Whisper, it is quadratic due to self-attention matrix computations. The Conformer model combines CNN with self-attention mechanisms, which affects its computational complexity.

Memory requirements depend on the sequence length n and the dimensionality of the hidden state or feature space d. In LSTM, each sequence element retains a hidden state of size d, while in BiLSTM, memory requirements double due to bidirectional processing. However, overall costs remain comparable to LSTM due to parallel computation. For CNNs, memory usage is determined by sequence length n, convolutional kernel size k, and feature dimension d, as each convolution processes k elements. Wav2Vec2 uses convolutional layers for feature extraction, ensuring efficient memory management, whereas Whisper requires more memory due to self-attention matrix storage. The highest memory requirements are observed in Conformer, which must store both the self-attention matrix and convolutional components, adding to computational complexity.

Overall, the effectiveness of ASR using NNs depends on the quality and volume of training data, model parameters, language complexity, and the applied post-processing and optimization mechanisms. Evaluating the obtained results using the described metrics enables the development of improvement strategies to refine the model and enhance recognition accuracy.

Conclusions

ASR is a crucial area of AI development, with applications across a wide range of industries. The integration of NN models has significantly improved ASR system accuracy, enabling their effective use in real-world scenarios. The objective of this study was to describe a generalized approach and algorithms for applying NNs, including RNN-based models (specifically LSTM, BiLSTM), CNNs, and transformer-based NNs, to the speech recognition task. The research findings indicate that NNs represent a modern and efficient approach to ASR. Each architectural type has specific implementation characteristics that impact recognition accuracy, computational complexity, and memory requirements. LSTM networks allow for context retention and the processing of variable-length input signals, whereas BiLSTM networks account for context in both directions. RNNs require fewer computational resources, making them suitable for small datasets and less complex tasks. CNNs serve as a powerful tool for extracting acoustic features in ASR and offer high computational speed due to parallelization capabilities. However, since they do not inherently capture temporal dynamics, they are often combined with RNN or transformers. Transformer-based architectures achieve the highest speech recognition accuracy by effectively processing long sequences, capturing global dependencies, and adapting to various accents, noise conditions, and speech styles. However, their primary drawback is high computational complexity and substantial memory requirements, which can be critical for resource-constrained implementations.

The architectural descriptions of modern NNs presented in this study provide a well-founded basis for selecting an appropriate ASR model for specific application contexts. The study outlines input requirements, areas of application, advantages, and limitations of each NN type. The comparative analysis of NN performance presented in this work can be utilized to predict the effectiveness of implementing a particular NN architecture.

References

[1] Penaloza, M. (2024). *Analysis of progress in speech recognition models*. URL: https://forum.effectivealtruism.org/posts/2i4SyjScgsQ4qbfDH/analysis-of-progress-in-speech-recognition-models-2?utm_source=chatgpt.com

[2] Ahlawat, H., Aggarwal, N., & Gupta, D. (2025). Automatic speech recognition: a survey of deep learning techniques and approaches. *International Journal of Cognitive Computing in Engineering*, 6, 201—237. doi: 10.1016/j.ijcce.2024.12.007.

[3] Chan, W., Jaitly, N., Le, Q. V., & Vinyals, O. (2016). Listen, attend and spell: a neural network

for large vocabulary conversational speech recognition. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP`16), Shanghai, China.

[4] Chiu, C.-C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., & Chen, Z. (2018). State of the art speech recognition with sequence-to-sequence models. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP`18), Calgary, Alberta, Canada.

[5] Kons, Z., Aronowitz, H., Morais, E., Damasceno, M., Kuo, H., Thomas, S., & Saon, G. (2022). Extending RNN-T-based speech recognition systems with emotion and language classification. *IBM Research AI*, 7, 1—4. doi: org/10.48550/arXiv.2207.13965.

[6] Zhang, Z., & Wu, Y. (2020). BiLSTM-CRF model for sequence labeling: a comparative analysis. *Journal of Artificial Intelligence Research*, 67, 731—755.

[7] Feng, Y. (2023). Intelligent speech recognition algorithm in multimedia visual interaction via BiLSTM and attention mechanism. *Neural Computing and Applications*, 36, 2371—2383. doi: 10.1007/s00521-023-08959-2.

[8] Hamzah, A., Abdelaziz, A., Hegazy, I., & Fayed, Z. (2021) Arabic speech recognition using end-to-end deep learning. *IET Signal*, 15(8), 521—534. doi: 10.1049/sil2.12057.

[9] Samin, A., Kobir, H., Kibria, S., & Rahman, S. (2021). Deep learning based large vocabulary continuous speech recognition of an under-resourced language Bangladeshi Bangla. *Acoustical Science and Technology*, 42(5), 252—260. doi: 10.1250/ast.42.252.

[10] Bekarystankyzy, A., Mamyrbayev, O., & Anarbekova, T. ACM transactions on Asian and low-resource language. *Information Processin*g, 23(6), 1—17. doi:10.1145/366356.

[11] Sarhan, A., Elshennawy, N., & Ibrahim, D. (2021). HLR-Net: a hybrid lip-reading model based on deep convolutional neural networks. *Computers, Materials & Continua*, 68(2), 1531—1549. doi: 10.32604/cmc.2021.016509.

[12] Thejha, B., Yogeswari, S., & Jeyalakshmi, J. (2023). Speech recognition using quantum convolutional neural network. VIII International Conference on Science Technology Engineering and Mathematics (ICONSTEM`23), TamilNadu, India.

[13] Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). *Conformer: convolution-augmented transformer for speech recognition*. URL: https://arxiv.org/abs/2005.08100.

[14] Baevski, A., Zhou, P., & Auli, M. (2020). *Wav2Vec 2.0: a framework for self-supervised learning of speech representations*. URL: https://arxiv.org/abs/2006.11477.

[15] Graham, C., & Roll, N. (2024). Evaluating OpenAI's Whisper ASR: Performance analysis across diverse accents and speaker traits. *The Journal of the Acoustical Society of America*, 4(2), ID 025206. doi: 10.1121/10.0024876.

[16] Yalova, K., Babenko, M., Sheliuh, K. (2024). Audio signal Pre-processing within speech recognition task. *Mathematical modelling*, 2(51), 9—18. doi: 10.31319/2519-8106.2(51)2024.317425.

[17] Yalova, K., Yashyna, K., Babenko, M. (2023). Automatic speech recognition system with dynamic time warping and Mel-Frequency Cepstral coefficients. *CEUR*, 3396, 141—151.

[18] Sruthi, V. T., Sidharth, K., Srivibhushanaa, S., Sanoj, C.S. (2020). Automatic speech recognition using Recurrent Neural Network. *International Journal of Engineering Research & Technology*, 9, 777—781. doi: 10.17577/IJERTV9IS080343.

[19] Magalhaes, R. P., Vasconcelos, D., Fernandes, G., Cruz, L., Sampaio, M., Fernandes de Macedo, J. & Coelho da Silva, T. (2022). Evaluation of automatic speech recognition approaches. *Journal of Information and Data Management*, 13(3), 366—377.

Список використаної літератури

1. Penaloza M. Analysis of progress in speech recognition models. URL: https://forum.effectivealtruism.org/posts/2i4SyjScgsQ4qbfDH/analysis-of-progress-in-speech-recognition-models-2?utm_source=chatgpt.com (дата звернення 01.02.2025).
2. Ahlawat H., Aggarwal N., Gupta D. Automatic speech recognition: a survey of deep learning techniques and approaches. *International Journal of Cognitive Computing in Engineering*. 2025. Vol. 6. P. 201—237. DOI: 10.1016/j.ijcce.2024.12.007.

3. Chan W., Jaitly N., Le Q. V., Vinyals O. Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, China, 2016, pp. 4960—4964.
4. Chiu C.-C., Sainath T. N., Wu Y., Prabhavalkar R., Nguyen P., Chen Z. State of the art speech recognition with sequence-to-sequence models. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Alberta, Canada, 2018, pp. 4774—4778.
5. Kons Z., Aronowitz H., Morais E., Damasceno M., Kuo H., Thomas S., Saon G. Extending RNN-T-based speech recognition systems with emotion and language classification. *IBM Research AI*. 2022. Vol. 7. P. 1—4. DOI: org/10.48550/arXiv.2207.13965.
6. Zhang Z., Wu Y. BiLSTM-CRF model for sequence labeling: a comparative analysis. *Journal of Artificial Intelligence Research*. 2020. Vol. 67. P. 731—755.
7. Feng Y. Intelligent speech recognition algorithm in multimedia visual interaction via BiLSTM and attention mechanism. *Neural Computing and Applications*. 2023. Vol. 36. P. 2371—2383. DOI: 10.1007/s00521-023-08959-2.
8. Hamzah A., Abdelaziz A., Hegazy I., Fayed Z. Arabic speech recognition using end-to-end deep learning. *IET Signal*. 2021. Vol. 15. I. 8. P. 521—534. DOI: 10.1049/sil2.12057.
9. Samin A., Kobir H., Kibria S., Rahman S. Deep learning based large vocabulary continuous speech recognition of an under-resourced language Bangladeshi Bangla. *Acoustical Science and Technology*. 2021. Vol. 42. I. 5. P. 252—260. DOI: 10.1250/ast.42.252.
10. Bekarystankyzy A., Mamyrbayev O., Anarbekova T. ACM transactions on Asian and low-resource language. *Information Processin*g. Vol. 23. I. 6. P. 1—17. DOI:10.1145/366356.
11. Sarhan A., Elshennawy N., Ibrahim D. HLR-Net: a hybrid lip-reading model based on deep convolutional neural networks. *Computers, Materials & Continua*. 2021. Vol. 68. I. 2. P. 1531—1549. DOI: 10.32604/cmc.2021.016509.
12. Thejha B., Yogeswari S., Jeyalakshmi J. Speech recognition using quantum convolutional neural network. *Proceeding of the VIII International Conference on Science Technology Engineering and Mathematics*, TamilNadu, India, 2023, pp. 1—7.
13. Gulati A., Qin J., Chiu C., Parmar N., Zhang Y., Yu J., Han W., Wang S., Zhang Z., Wu Y., Pang R. Conformer: convolution-augmented transformer for speech recognition. URL: https://arxiv.org/abs/2005.08100 (дата звернення 20.02.2025).
14. Baevski A., Zhou P., Auli M. Wav2Vec 2.0: a framework for self-supervised learning of speech representations. URL: https://arxiv.org/abs/2006.11477 (дата звернення 01.03.2025).
15. Graham C., Roll N. Evaluating OpenAI's Whisper ASR: Performance analysis across diverse accents and speaker traits. *The Journal of the Acoustical Society of America*. 2024. Vol. 4. I. 2. ID 025206. DOI: 10.1121/10.0024876.
16. Yalova K., Babenko M., Sheliuh K. Audio signal Pre-processing within speech recognition task. *Mathematical modelling*. 2024. Vol. 2(51). P. 9—18. DOI: 10.31319/2519-8106.2(51)2024.317425.
17. Yalova K., Yashyna K., Babenko M. Automatic speech recognition system with dynamic time warping and Mel-Frequiency Cepstral coefficients. *CEUR*. 2023. Vol. 3396. P. 141—151.
18. Sruthi V. T., Sidharth K., Srivibhushanaa S., Sanoj C.S. Automatic speech recognition using Recurrent Neural Network. *International Journal of Engineering Research & Technology*. 2020. Vol. 9. P. 777—781. DOI: 10.17577/IJERTV9IS080343.
19. Magalhaes R. P., Vasconcelos D., Fernandes G., Cruz L., Sampaio M., Fernandes de Macedo J. Coelho da Silva T. Evaluation of automatic speech recognition approaches. *Journal of Information and Data Management*. 2022. Vol. 13. I. 3. P. 366—377.

Надійшла до редколегії 31.03.2025