Analysis of the limits of model improvement in deep learning and performance saturation assessment*

Samat Mukhanov[1,*,†], Orken Mamyrbayev[2,†], Dair Katayev[1,†], Alikhan Kalmurzayev[1,†], Zhasulan Oteuli[1,†], Shaim Yakupov[1,†] and Daryn Amrin[1,†]

[1] International Information Technology University, Manas St 34/1 050040 Almaty, Kazakhstan
[2] Institute of Information and Computational Technologies, Shevchenko str. 28 050010 Almaty, Kazakhstan

Abstract

Machine learning models, particularly those used in automatic speech recognition (ASR), generally exhibit diminishing returns when dataset size and resource utilization escalate. This study analyzes the performance saturation observed during ASR model training, using the Whisper-Tiny model to illustrate this trend. The research identifies key factors contributing to performance limitations, including dataset size, model architecture, and resource utilization. As dataset size exceeds 15,000 samples, improvements in Word Error Rate (WER) and Character Error Rate (CER) decline significantly, confirming diminishing returns.

The study also examines resource utilization, revealing that training time increases non-linearly with dataset size. While GPU memory usage remains relatively constant, CPU and RAM usage fluctuate, indicating potential inefficiencies. To address computational constraints, techniques such as streaming data processing and fixed-length audio segments are implemented to enhance training efficiency. Additionally, evaluation bottlenecks are mitigated by using fixed test dataset sizes, ensuring quicker and more consistent assessments.

Efficient processing strategies, including gradient accumulation and mixed-precision training, are explored to reduce resource consumption without compromising performance. Visualization techniques, such as correlation heatmaps and performance plots, highlight the trade-offs between dataset size, computational cost, and model accuracy.

The findings emphasize the importance of balancing resource allocation and data volume to optimize ASR training workflows. By acknowledging and addressing performance saturation, researchers can develop more scalable and efficient ASR models, making advanced speech recognition technology more accessible in resource-constrained environments.

Keywords

paper template, machine learning, automatic speech recognition (ASR), performance saturation, diminishing returns, Word Error Rate (WER), Character Error Rate (CER), training efficiency, resource optimization, computational cost, deep learning, model scalability, dataset size, GPU memory, Whisper-Tiny, streaming data processing.

1. Introduction

Machine learning models typically improve as they are trained with more data and computational power. However, at some point, this improvement diminishes or stops altogether. This phenomenon is known as diminishing returns—where further increases in training time, data [1], or model complexity result in minimal or no gains in performance [2], [3].

Understanding when and why this happens is crucial for optimizing machine learning workflows. Continuing training beyond a model's performance limits leads to wasted time and resources without tangible benefits. This article explores the primary reasons why a model ceases to improve, how to recognize these signs, and potential solutions.

*AIT 2025: 1st International Workshop on Application of Immersive Technology, March 5, 2025, Almaty Kazakhstan
[1,*] Corresponding author.
[†] These authors contributed equally.
kvant.sam@gmail.com (S. Mukhanov); morkenj@mail.ru (O. Mamyrbayev); 31151@iitu.edu.kz (D. Katayev); 31161@iitu.edu.kz (A. Kalmurzayev); 31160@iitu.edu.kz (Zh. Oteuli); 31163@iitu.edu.kz (Sh. Yakupov); 41376@iitu.edu.kz (D. Amrin)
0000-0001-8761-4272 (S. Mukhanov); 0000-0001-8318-3974 (O. Mamyrbayev); 0009-0003-0684-6947 (D. Amrin)

Most standard machine learning models improve with additional data and computational power. However, at some point, this improvement slows down or stops altogether. This phenomenon is known as diminishing returns—where extra training time, data, or model complexity leads to only marginal gains. Understanding when and why this happens is crucial for optimizing machine learning workflows. Training beyond a model's performance limits wastes time and resources without yielding any meaningful benefits.

This issue is particularly relevant for speech recognition models, as they typically require vast amounts of data and computational resources [2],[4]. Advances in Automatic Speech Recognition (ASR) have been closely tied to progress in deep learning, particularly through more advanced recurrent and convolutional neural networks [3]. These improvements have enabled models to better understand human speech, accounting for variations in accents, background noise, and speaking styles, and low-resource languages [5],[6],[7].

However, even state-of-the-art models face performance saturation. Identifying the key factors contributing to this limitation—such as dataset size, model architecture, and training strategies—can help researchers build more efficient training pipelines [8],[9],[10]. This article explores these factors, drawing insights from recent studies and experiments, and offers practical recommendations for optimizing ASR workflows.

2. Literature Review or Problem Statement

Training a machine learning model involves teaching it to recognize patterns in data to make accurate predictions or decisions. The process typically includes the following key steps:

Process	**Process**	**Advantages**	**Disadvantages**
Collecting and Preparing Data	Datasets such as LibriSpeech provide spoken language Recordings [11].	Captures variations in accents, speaking styles, and noise conditions.	Large datasets require significant storage and processing.
Data Preprocessing	Cleaning noise, normalizing values, and converting data for Processing [12].	Improves model input quality and extraction of relevant features.	Noise removal may inadvertently lose useful information.
Model Selection and Architecture	Use of RNNs and CNNs for sequential data processing in speech recognition [13].	Effective in capturing sequential relationships in speech data.	High computational cost and risk of overfitting complex models.
Model Training	Optimization algorithms (e.g., SGD) iteratively minimize prediction errors [14].	Allows continuous performance improvement with additional iterations.	Overtraining may lead to diminishing returns and overfitting.
Evaluation and Testing	Metrics like WER and CER measure model performance on unseen Data [15].	Provides insight into model accuracy and generalization capabilities.	Evaluation may not capture all real-world speech conditions.
Visualizing the	Tools like	Enables visualization	Requires additional

Learning Process	TensorBoard track model performance over time [16], [17].	of metrics and training dynamics for better insights.	resources and time for visualization setup.

This Figure 2.1 – Correlation heatmap provides key insights into the relationships between different training metrics, helping to understand diminishing returns in model improvement. Strong positive correlations (e.g., between Train WER and Train CER, and between Test WER and Test CER) confirm that these error metrics behave similarly. The inverse correlation between Samples and Test WER (-0.92) suggests that increasing training data leads to better generalization, but diminishing gains are evident as correlation weakens for Valid WER. Notably, Training Time and RAM Usage exhibit high positive correlation (0.79), showing the growing computational cost with more samples. Meanwhile, Batch Processing Time does not always scale linearly with Samples, hinting at potential inefficiencies [6]. These findings support the argument that beyond a certain point, additional training data increases computational cost disproportionately to performance gains.

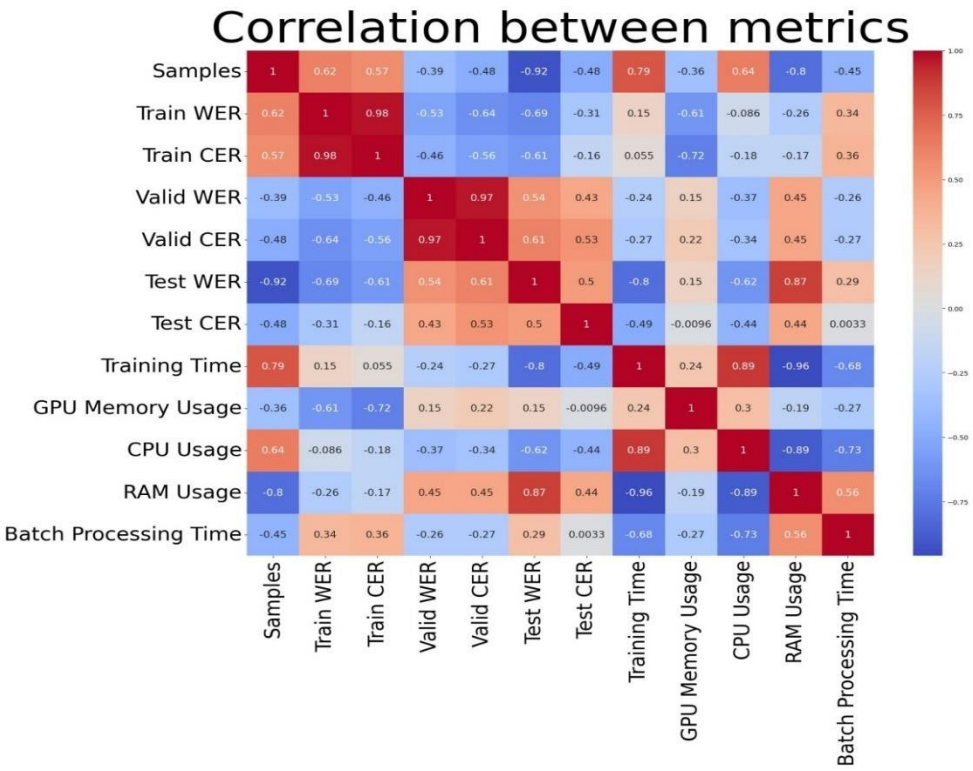

Figure 2.1: Correlation between metrics.

The heatmap in Figure 2.1 presents the correlation between various performance and computational metrics used in evaluating the Whisper-Tiny model. The correlation coefficients range from **-1 to 1**, where **positive values** indicate a direct relationship, and **negative values** indicate an inverse relationship.

3. Methodology

3.1. Performance Saturation Analysis

As the dataset size increases from 5,000 to 50,000 samples, a pattern of diminishing returns becomes evident. Initially, with 5,000 samples, the test Character Error Rate (CER) is relatively

high, reflecting the model's struggle with limited training data. For instance, at this stage, the test Word Error Rate (WER) is 0.5300, indicating that the model misinterprets over 50% of words.

With an increase in sample size, the CER gradually decreases, demonstrating that the model benefits from additional training data. However, beyond a certain threshold, roughly between 30,000 and 40,000 samples, improvements in performance become marginal. This suggests that while adding more data helps in the earlier stages, the model reaches a saturation point where additional samples contribute little to further reducing the error rate.

The persistent gap between the test and training CER values suggests that the model may still be overfitting, learning patterns specific to the training set while struggling with generalization. Potential strategies to address this issue include refining the model architecture, optimizing hyperparameters, or incorporating regularization techniques. Furthermore, improvements in data quality, rather than sheer quantity, might yield better gains in accuracy at this stage:

$$\begin{aligned} Test\ WER_{5000} &= 0.5300 \\ Test\ WER_{5000} &= 0.4559 \end{aligned} \tag{1}$$

Beyond 15,000 samples, improvements slow significantly.

$$WER = \frac{\mathcal{S} + \mathcal{D} + \mathcal{I}}{N} = \frac{\mathcal{S} + \mathcal{D} + \mathcal{I}}{\mathcal{S} + \mathcal{D} + \mathcal{C}} \tag{2}$$

where S – is the number of substitutions, D – is the number of deletions, I – is the number of insertions, C – is the number of correct words, N – is the number of words in the reference (N=S+D+C)

$$CER = [\frac{i + s + d}{n}] * 100 \tag{3}$$

3.2. Training Time Scaling

Training time increases non-linearly with dataset size:

$$\begin{aligned} T_{5000} &= 818.23\ seconds \\ T_{50500} &= 2644.48\ seconds \end{aligned} \tag{4}$$

as:

$$samples\ per\ second = \frac{effective\ batch\ size \times steps\ per\ second}{gradient\ accumulation\ steps} \tag{5}$$

where:

effective_batch_size = per_device_train_batch_size * num_devices (In your case, 8 * 1 = 8, assuming 1 GPU)

steps_per_second = number of training steps completed per second (This depends on hardware, model size, and optimizations)

gradient_accumulation_steps = number of steps before updating weights (In your case, 1, so it doesn't change the formula)

3.3. Resource Usage

- GPU memory usage remains roughly constant.
- CPU and RAM usage fluctuates without a clear trend.

3.4. Model Performance Stability

Training loss decreases initially but levels off after 15,000 samples, confirming performance saturation. The plots illustrate the trade-offs in training a speech recognition model with increasing dataset size. While test WER and CER show improvement as more samples are added, training and validation metrics fluctuate, suggesting inconsistencies in generalization. Resource usage metrics (GPU, CPU, RAM) indicate increasing computational costs, with training time and batch processing time rising significantly beyond 30,000 samples. This demonstrates that scaling dataset size alone is not always optimal and reinforces the importance of balancing data volume with model efficiency. You should place this analysis in the section discussing the diminishing returns of increasing data in research, supporting the argument that more data is not always the best solution and computational constraints must be considered.

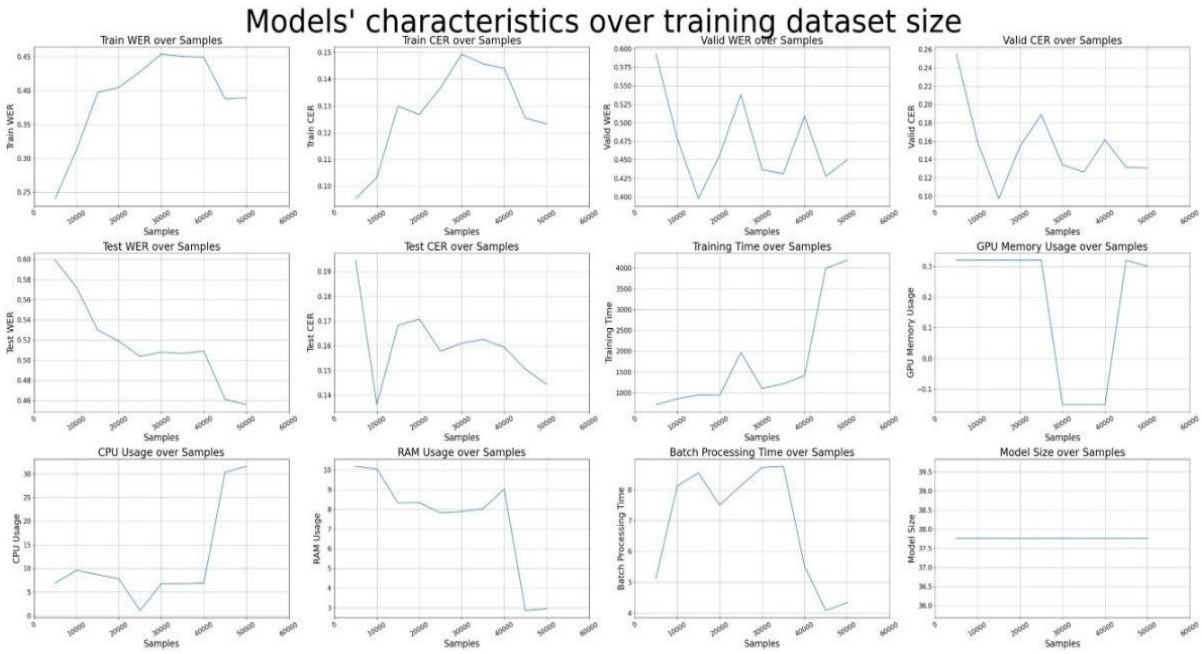

Figure 3.1: Models' characteristics over training dataset size.

following formula:

$$num\,epochs = min\left(6, \frac{max\,steps \times batch\,size \times gradient\,accumulation\,steps}{num\,samples}\right) \quad (6)$$

where:

- num_samples = total number of training samples,
- batch_size = per_device_train_batch_size (8 in your case),
- gradient_accumulation_steps (1 in your case),
- max_steps = 3000.

4. Experiments and Results

4.1. Efficient Processing Strategies

The Whisper-Tiny model was selected primarily to minimize computational costs while achieving faster convergence. As a lightweight version of the Whisper architecture, it enables efficient training and inference without requiring extensive hardware resources. By using a smaller model, the risk of overfitting is reduced, allowing for a more generalized learning process even with a moderate dataset size. Additionally, the smaller model size facilitates quicker saturation, meaning that improvements in performance diminish at an earlier stage compared to larger models. A Whisper-Tiny model was chosen to reduce computational costs and reach saturation faster. Streaming Dataset Approach: Uses IterableDataset to dynamically load and process small data batches. Librosa-Based Audio Processing: Keeps only essential audio fragments in memory. Padding & Truncation: Audio samples standardized to 10-second segments.

Robust Model Evaluation:

WER: Measures incorrect transcriptions at the word level.

CER: Provides a finer, character-level evaluation.

The bar plot shows how Figure 4.1 – Word Error Rate (WER) decreases as training data increases, confirming that more data initially improves model performance. However, beyond 15,000–20,000 samples, the improvement slows significantly, demonstrating performance saturation. The training set (blue bars) shows steady WER reduction, but validation (gray) and test (red) sets maintain a gap, highlighting limited generalization.

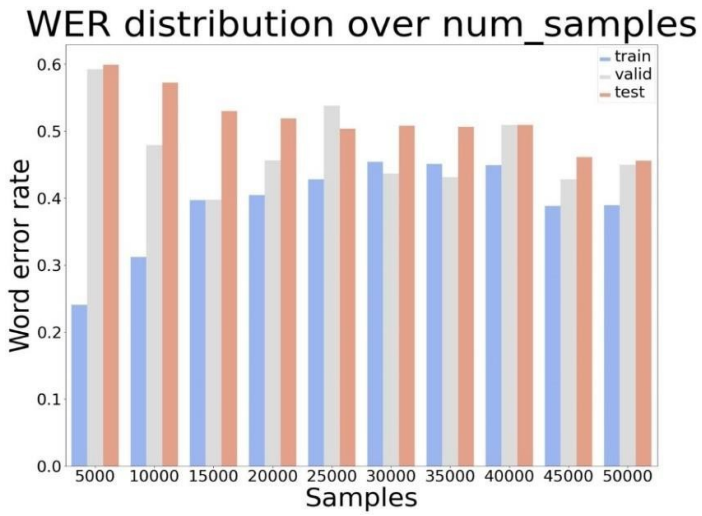

Figure 4.1: Word Error Rate (WER).

This supports the claim that simply adding data is not always effective and suggests the need for alternative optimizations like better architecture or regularization. Visualization reinforces that increasing dataset size beyond a threshold has diminishing returns, making computational efficiency a key consideration.

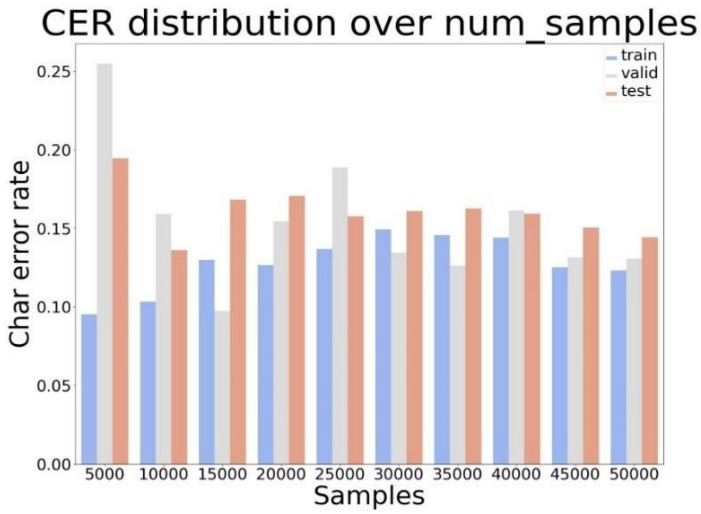

Figure 4.2: Char error rate.

The chart in Figure 4.2 represents the Character Error Rate (CER) distribution over different sample sizes for training, validation, and test datasets.

5. Discussion

One of the largest challenges was Out-of-Memory (OOM) errors. Initially, attempting to load the entire dataset into RAM caused system crashes. To resolve this, an IterableDataset was implemented, enabling data to be streamed in small chunks instead of being loaded all at once. Another major issue was inefficiency in audio processing. The dataset contained large, variable-length audio files, which led to batching problems during training. This was addressed by segmenting the audio into uniform 10-second chunks, ensuring consistency across batches.

Training instability also emerged as a challenge, primarily due to prolonged training on limited hardware, which led to overfitting. To mitigate this, several strategies were applied: the maximum number of training steps was capped at 3,000, and a small batch size of 8 was used in combination with gradient accumulation to optimize resource utilization.

Finally, evaluation bottlenecks slowed down development. Processing the entire dataset was time-consuming, making evaluations inefficient. To streamline this process, the test dataset size was fixed at 5,000 samples, allowing for faster evaluations without significantly compromising accuracy.

6. Conclusion

This study optimizes Whisper-Tiny for low-resource training by leveraging streaming data processing, mixed-precision training, and efficient memory batching. The pipeline is designed to accommodate real-world constraints, making ASR model training both scalable and efficient in resource-limited environments.

The research underscores the concept of performance saturation in deep learning models. Initially, as the number of training samples increases, the Word Error Rate (WER) decreases, reflecting improved model performance. However, beyond a certain threshold (around 15,000 samples), the rate of improvement slows significantly, demonstrating the phenomenon of diminishing returns. This aligns with the study's findings, highlighting that while expanding dataset size and computational resources can enhance model performance, there is a limit beyond which further investments yield minimal gains.

Recognizing this saturation point is essential for optimizing machine learning workflows, especially in resource-constrained settings. This reinforces the importance of strategies like the Whisper-Tiny model and streaming dataset approaches, which promote faster convergence and

efficient resource utilization. The findings in this section support the broader discussion on effective processing techniques and model performance stability in machine learning.

Special thanks to Diana Baranovskaya for generously providing a computing machine for model training.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Thompson, N. C. *Deep Learning's Computational Cost*. In IEEE Spectrum, 2024. Available at: https://spectrum.ieee.org/amp/deep-learning-computational-cost-2655082754.

[2] Yu, D., & Deng, L. *Automatic Speech Recognition: A Deep Learning Approach*. In Springer, 2014. Available at: Automatic speech recognition book.

[3] Jurafsky, D., & Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd Edition)*. Stanford University. In Github, 2021. Available at: Speech and Language Processing.

[4] Huang, X., Acero, A., & Hon, H.-W. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. In Github, 2001. Available at: Spoken Language Processing.

[5] Asma Trabelsi , Sébastien Warichet, Yassine Aajaoun, Séverine Soussilane. "Evaluation of the efficiency of state-of-the-art Speech Recognition engines In *ScienceDirect,* 2022. Available at: https://www.sciencedirect.com/science/article/pii/S1877050922014338.

[6] Korbinian Kuhn, Verena Kersken, Benedikt Reuter, Niklas Egger, Gottfried Zimmermann. "Measuring the Accuracy of Automatic Speech Recognition Solutions." In *ACM Digital Library vol 16, no 4. 2022*. Available at: https://dl.acm.org/doi/full/10.1145/3636513.

[7] Stolypin Vestnik. *Advances in Speech Recognition Technologies for Low-Resource Languages*. 2024. Available at: https://stolypin-vestnik.ru/wp- content/uploads/2024/05/15.pdf.

[8] Rabiner, L., & Juang, B.-H. *Fundamentals of Speech Recognition*. In Github, 1993. Available at: Speech Recognition Fundamentals.

[9] Manning, C., & Schütze, H. *Foundations of Statistical Natural Language Processing*. In Github, 1999. Available at: Foundations of NLP.

[10] Belcic, I. *Hyperparameter Tuning: Approaches and Best Practices*. In IBM, 2024. Available at: https://www.ibm.com/think/topics/hyperparameter-tuning.

[11] C. Kenshimov, S. Mukhanov, T. Merembayev, and D. Yedilkhan, "A comparison of convolutional neural networks for Kazakh sign language recognition," EEJET, vol. 5, no. 2 (113), pp. 44–54, Oct. 2021, doi: 10.15587/1729-4061.2021.241535.

[12] S. B. Mukhanov and R. Uskenbayeva, "Pattern Recognition with Using Effective Algorithms and Methods of Computer Vision Library," in Optimization of Complex Systems: Theory, Models, Algorithms and Applications, vol. 991, H. A. Le Thi, H. M. Le, and T. Pham Dinh, Eds., in Advances in Intelligent Systems and Computing, vol. 991. Cham: Springer International Publishing, 2020, pp. 810–819. doi: 10.1007/978-3-030-21803-4_81.

[13] Mukhanov, Samat & Uskenbayeva, Raissa & Rakhim, Abd & Akim, Akbota & Mamanova, Symbat. (2024). Gesture recognition of the Kazakh alphabet based on machine and deep learning models. Procedia Computer Science. 241. 458-463. 10.1016/j.procs.2024.08.064.

[14] Bazarbekov I.M., Ipalakova M.T., Daineko E.A., Mukhanov S.B. DEVELOPMENT AND DATA ANALYSIS OF A ROBO-PEN FOR ALZHEIMER'S DISEASE DIAGNOSIS: PRELIMINARY RESULTS. Herald of the Kazakh-British technical university. 2024;21(3):78-89. (In Kazakh) https://doi.org/10.55452/1998-6688-2024-21-3-78-89.

[15] Alpar, S., Faizulin, R., Tokmukhamedova, F., & Daineko, Y. (2024). Applications of Symmetry-Enhanced Physics-Informed Neural Networks in High-Pressure Gas Flow Simulations in Pipelines. Symmetry, 16(5), 538. https://doi.org/10.3390/sym16050538.
[16] Nuralin M.; Daineko Y.; Aljawarneh S.; Tsoy D.; Ipalakova M. The real-time hand and object recognition for virtual interaction. 2024. PeerJ Computer Science.
[17] Borodkin K.; Nurtas M.; Altaibek A.; Daineko Y.; Otepov T. Data Pre-processing and Visualization for Machine Learning Models and its Applications in Education. 2024. CEUR Workshop Proceedings.