UDC 004.8
DOI https://doi.org/10.32782/2663-5941/2025.6.2/25

Kostyuchenko A.V.
Zhytomyr Polytechnic State University

Kushnir N.O.
Zhytomyr Polytechnic State University

Loktikova T.M.
Zhytomyr Polytechnic State University

DEVELOPMENT AND RESEARCH OF EFFECTIVE DEEP LEARNING MODELS FOR IMPROVING THE RESOLUTION OF IMAGES AND VIDEO SEQUENCES

Super-resolution (SR) of images and video sequences is a fundamental task in computer vision that aims to recover high-quality details from low-resolution input data. The end result is a high-resolution version of the original image or video, which is key to improving visual quality, increasing detail, and enhancing the accuracy of computer vision algorithms. This technology has great practical significance, covering areas such as medical imaging to improve diagnostic accuracy, satellite imaging to monitor environmental changes, security and surveillance to improve video quality, digital media, and consumer electronics. In recent years, deep learning, particularly convolutional neural networks (CNN) and generative adversarial networks (GAN), has led to significant breakthroughs in this field, enabling high-quality restoration. The article provides an overview of existing SR architectures, with a detailed analysis of well-known models such as Real-ESRAN and RT4KSR, identifying their advantages when applied in real-world scenarios. Taking into account the differences between synthetic and real degradations, as well as the need to balance quality and computational efficiency, a modified simplified architecture for real-time image scaling up to 4K is proposed, focused on practical deployment. The architecture includes five major improvements: tile processing with overlapping to reduce artifacts at tile boundaries; dynamic image scaling that adapts operations to available resources; optimization through quantization to reduce memory and computational complexity; convenient replacement of blocks of the corresponding model with simpler analogues without significant loss of quality; use of a set of heuristics for processing video sequences, ensuring temporal consistency and avoiding flickering. Possible ways to implement these improvements, their impact on performance and image quality, and methods for evaluating the results are considered. The proposed approach aims to overcome current limitations and facilitate the practical deployment of image enlargement systems.

Key words: *resolution, image, video, processing, scaling, deep learning, neural networks.*

Formulation of the problem. Super-resolution (SR) of images and videos is a fundamental task in computer vision that aims to reconstruct high-quality images or video sequences from their low-resolution (LR) counterparts. This procedure, also known as image scaling, interpolation, enlargement, or expansion, aims to restore lost detail and clarity, which is important for many practical applications. Image super-resolution (ISR) is a machine learning task where the goal is to increase the resolution of an image, typically by a factor of 4 or more, while preserving its content and details [1]. In turn, video super-resolution (VSR) extends this concept to dynamic sequences, requiring not only spatial enhancement but also maintaining temporal consistency between frames [2]. Standard methods usually work on perfectly distorted data, but in reality, images undergo complex distortions (zoom, noise, JPEG artifacts). The use of generative adversarial neural networks (GANs) has significantly improved visual quality, but they can create artifacts or require a lot of computing power. In addition, it is important for video to use temporal information (frame reuse) to avoid reprocessing each frame and reduce latency. Thus, the task lies in creating efficient and high-quality SR models that will ensure high accuracy and speed.

Analysis of recent research and publications. A significant breakthrough in improving image res-

olution has been achieved with the advent of deep learning. Deep learning-based methods have largely surpassed traditional interpolation-based approaches (e.g., bicubic), which are often characterized by excessive smoothing and the appearance of artifacts. Since around 2011, the introduction of convolutional neural networks (CNNs) has had a significant impact, establishing their dominance in the field of SR for almost a decade. Later, between 2017 and 2023, the integration of generative adversarial networks (GANs) and Transformer architectures led to exponential growth in SR research [3].

There has been a noticeable shift away from models optimized primarily for traditional pixel metrics, namely Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), to models implemented based on GAN and with perceptual losses, which provide more visually pleasing and realistic results. Despite significant progress, some aspects of the resolution enhancement task remain unresolved, creating opportunities for further research and innovation. Current research trends include: developing effective dynamic network models that can adapt to content complexity; analyzing multi-scale architectures for improved generalization and flexible scaling for different images magnification factors; integration of attention mechanisms to improve feature representation and computational efficiency by focusing on relevant parts of the image; development of complex degradation models to address the challenges of image degradation in reality.

Model mismatch is one of the main challenges. Although deep learning has significantly improved SR, the mismatch between the degradation models used for training and the complex, unknown degradations that occur in low-resolution images remains a problem. Models trained on simplified synthetic data (e.g., downsampled using bicubic interpolation) typically perform poorly on real images due to complex degradations that arise from camera systems, image editing, and internet transmission. This is not just a technical detail, but a significant obstacle to the practical deployment of the model. This problem highlights that future research and practical solutions should prioritize robust degradation modeling or "blind" resolution enhancement techniques, as the model's robustness to real-world inputs is as important as its theoretical peak performance on baseline test sets.

Achieving both high perceptual quality and real-time performance, especially for 4K resolution, remains a significant challenge due to the high memory and computational costs of state-of-the-art models. Although GANs have improved perceptual quality, they can sometimes introduce unwanted artifacts, namely hard lines, unnatural noise, jagged edges, color distortion, or color shifts. Most existing models lack the internal flexibility to dynamically adapt to changing input conditions (e.g., different input resolutions, changing frame rate requirements for video streams) in real time without the need for separate models or intensive retraining. The effective use of temporal information in video sequences to improve quality and maintain consistency between frames, while achieving real-time performance, remains a challenging area of research [4]. This work will address these issues in a comprehensive manner.

Task statement. The goal of the research is to improve deep learning models by creating our own optimized model for scaling images and video sequences, taking into account real computing resources.

Outline of the main material of the study. Before describing the architecture of our own optimized model, let's look at the two main architectures on which it is based, namely Real-ESRGAN [5] and RT4KSR [6].

Real-ESRGAN is a network based on generative adversarial networks with ultra-high resolution (ESRGAN) with improved training on synthetic data. Real-ESRGAN features a specific degradation generation process (including blurring, noise, compression) and an upgraded discriminator (UNet with spectral normalization), which allows the model to successfully restore the textures of real images, thereby surpassing existing methods in terms of visual quality. The Real-ESRGAN network supports image scaling with arbitrary ratios, and a special RealESRGAN_x2plus model has been released for twofold (x2) magnification. In addition, Real-ESRGAN implements tile processing with the ability to specify the tile size in order to process large images in parts to reduce the amount of memory used.

That is, Real-ESRGAN, trained on complex synthetic data, is capable of improving details and effectively removing unpleasant artifacts for common real images, demonstrating high visual performance compared to previous methods on various test datasets.

RT4KSR is a core architecture specifically designed to solve the problem of achieving real-time performance for 4K resolution on commercial graphics processors. Starting with a simple scheme, the model gradually became more complex: important high-frequency details are highlighted in the early layers and then downsampled through deep feature maps, which reduces the amount of computation

while maintaining the fidelity of the reconstruction. RT4KSR uses pixel-unshuffle operations to transfer smaller-scale details to deep feature maps and structural reparameterization to speed up processing. According to the results obtained, the network outperforms bicubic scaling: for example, for a twofold image enlargement (1080→4K), PSNR increases from 33.916 to 34.193, and SSIM from 0.8829 to 0.8848. The network demonstrates successful preservation of detail at a much lower cost in terms of time compared to standard super-resolvers.

Based on the architectures described above, we created our own RealTimeSR architecture, which became the basis for developing a resolution enhancement model optimized for real-time applications. It includes several basic components that are common in modern SR architectures aimed at improving the efficiency and quality of image and video sequence processing.

The RealTimeSR network is designed to scale images up to 4K, improving efficiency through the use of deep but small 3x3 kernels, residual connections, pixel shuffle operations to increase size, and channel attention.

The network begins with a Conv convolutional layer with a 3x3 kernel, which converts three input channels (RGB) into nc feature channels (64 by default). This is followed by a ReLU activation function.

The model is based on a chain of eight residual blocks (RB). Each RB includes two Conv layers with a 3x3 kernel with padding to preserve spatial dimensions, Batch Normalization (BN) after each Conv layer, and a ReLU activation function after the first BN. An important component of the model is the Channel Attention mechanism [7], which is integrated into each residual block RB. This mechanism uses global average pooling (AdaptiveAvgPool2d) and max pooling (AdaptiveMaxPool2d) to aggregate spatial information, and then applies a nonlinear transformation (two linear layers with ReLU and Sigmoid activation functions) to compute weights for each channel. Thus, the weights scale the output features, allowing the model to dynamically focus on more informative channels. The output of the convolutional layers with the channel attention mechanism is added to the input residual tensor, which helps overcome the vanishing gradient problem and facilitates the training of deep networks.

After the Residual Block chain, another Conv convolution layer with BN is applied, designed for further processing of features before directly increasing the resolution.

Next, the PixelShuffle pixel rearrangement layer is used [8]. First, the Conv convolution layer transforms nc feature channels into nc* channels, which is equal to nc multiplied by the square of the enlargement factor. Then PixelShuffle rearranges the channel elements into spatial coordinates, effectively increasing the height and width of the image by n times. The model ends with a final Conv convolution layer, which converts the feature channels back into three RGB channels.

Schematically, the RealTimeSR architecture can be represented as shown in Fig. 1: the input RGB image enters a set of nc feature maps, then passes through eight residual blocks with channel attention, then is reduced to the same number nc, and finally the image is enlarged. This architecture combines efficient processing (due to its depth and pixel rearrangement) with adaptive channel attention. At one time, Residual Channel Attention Networks (RCAN) showed that a deep network with attention channels is effective, so a similar one was implemented inside each residual block. After scaling, the output convolutional layer restores a high-resolution three-channel color image.

The basic architecture of RealTimeSR already includes elements that contribute to high quality (residual connections, channel attention) and efficiency (pixel-shuffle). However, further improvements are needed to achieve the highest performance and adaptability for real-time 4K applications, especially considering complex real-world degradations and resource constraints.

It is proposed to integrate five major improvements into the basic architecture of RealTimeSR and its training process.

Processing images and videos with 4K or higher resolution can quickly exhaust the VRAM memory of a graphics processing unit (GPU), especially on commercial devices with limited resources. Even if the model can handle large images, this can lead to a significant increase in rendering time, so an alternative solution is needed to remedy this situation.

The best solution was to split the image into smaller parts tiles or patches each of which is processed separately, instead of processing the entire image as a whole. To avoid artifacts at the tile boundaries (e.g., seams), an “overlap” strategy is used. Each tile is processed with an additional “halo” or “ghost area” around it, equal to half the receptive field of the network. After processing, only the central part of each tile (Zone of Responsibility) without overlap is used to reconstruct the final image, and the overlap is discarded [9].

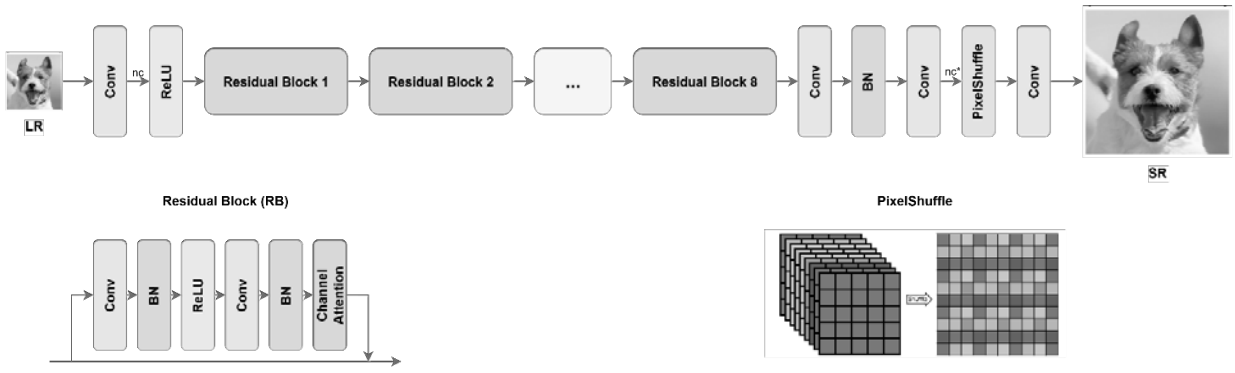

Fig. 1. Basic architecture of RealTimeSR

Thus, thanks to the division, VRAM usage is reduced, since only part of the image is stored in memory at a time. The proposed approach effectively eliminates artifacts at the boundaries, ensuring seamless reconstruction. Additionally, the tile size can be adapted to the available GPU memory, making the system more flexible and resistant to different hardware configurations.

Most existing models built on RT4KSR and ESRGAN are developed and trained for a fixed scaling factor, which also limits their flexibility, requiring separate models or retraining for different scaling needs.

This project implements a multi-scale architecture and logic for automatically selecting the scaling factor (2, 3, or 4) depending on the size of the input frame or the desired frame rate. The idea is as follows: if the source image is small or the system has processing speed requirements, the model selects a smaller scale (x2 or x3) to avoid increasing computational costs. Compared to a rigidly fixed x4 solution, the dynamic approach maintains high quality when processing speed needs to be changed [10].

To adapt the model to mobile and edge devices, the accuracy of some of its parameters is reduced. In particular, the model is converted to FP16 or INT8 format using the NVIDIA TensorRT tool to reduce the size of the model and speed up processing. According to research, special methods after training quantization can reduce processing time by 54% for some SR models with an imperceptible drop in quality. For example, by removing clipped activations in some approaches, processing delays are significantly reduced and model stability is improved with INT8 quantization [11]. This work uses standard PyTorch/TensorRT quantization schemes and a representative dataset to adjust the scales to avoid significant artifacts. The result is a model that runs two to three times faster on modern GPUs with FP16/INT8 compared to FP32, while PSNR and SSIM metrics change by no more than a few tenths of a point.

The replacement of conventional Residual Blocks with simpler blocks has been investigated. One option is to use the C3 block of the YOLOv5 model, which divides feature maps into two parts: one part is processed by a sequence of convolutional layers, and the other is passed through unchanged, after which the parts are combined. This is a technique for "thinning" features, which allows you to reduce the amount of computation without losing expressiveness. Another option is to use Ghost modules (GhostNet) or shift operations (GhostSR). GhostSR proposes generating "ghost" features using simple shifts to save over 40% of parameters and FLOPs while maintaining the same accuracy. Similarly, GhostNet shows that Ghost modules can reproduce information through simple linear transformations, which guarantees efficiency close to that of standard convolution [12]. It is planned to replace standard Residual Block convolutional blocks with Ghost blocks or MobileNetConv mobile convolutions. The latter use separate convolutional layers, which have long proven themselves to be effective for mobile networks [13]. All these replacements lead to a reduction in the amount of computation with a minimal decrease in PSNR and SSIM metrics.

For video, strategies will be implemented that allow not every frame to be processed completely again. Alternatively, previous results (or features) can be used to initialize the next frame. If some deep features have already been calculated in the previous frame, they can be transferred to the next frame using optical flow and spatial transformation [14]. For example, the BasicVSR architecture has shown that reusing features from neighboring frames and processing them sequentially can improve the quality and speed of video SR. A simple version of this approach can be implemented: when processing video frames, intermediate features will be stored and used as part

of the next input frame (before that, the frames will be aligned using optical flow). This approach will avoid repeated processing and bring the speed closer to real-time performance.

Let's move on to considering the implementation and training process of the model.

The RealTimeSR architecture is implemented in the PyTorch framework of the Python programming language. Balanced datasets of low (LR) and high (HR) resolution DFO image pairs (DIV2K + Flickr2K + OST) are used for training. L1 Loss (absolute pixel difference) was used as the loss function, as it correlates significantly with the PSNR metric. Adam was used as the optimizer.

The training procedure includes: for each mini-batch LR→HR transfer through the model, loss calculation, and backpropagation. After the training epoch, the average values of the PSNR and SSIM metrics are evaluated on the validation set. To avoid overly large images in the test, the SR output and HR target are cropped to the minimum common size. If PSNR improves, the current model is retained.

The training procedure uses channel attention, which obtains information for each block based on the average and maximum values of features across channels, and then generates a scaling factor through a two-layer FC block with a ReLU activation function. Thus, each Residual Block includes: two Conv convolutional layers with batch normalization BN, a ReLU activation function inside, channel attention, and an initial feature. Finally, after all the Residual Blocks, there is a series of upsampling layers: Conv → PixelShuffle → Conv, which forms the final SR image.

After training the model for 300 epochs, its effectiveness can be evaluated based on the results obtained. Experiments were conducted on standard test sets (Set5/Set14/BSD100) and on our own video sequences. Quality metrics – PSNR and SSIM – were calculated in the restored range [0,1] of pixel values. Table 1 shows a comparison of model testing using bicubic interpolation, Real-ESRGAN, RT4KSR, and RealTimeSR (Set14).

As can be seen from the table, the basic RealTimeSR architecture demonstrates better results (PSNR, SSIM) compared to other scaling methods. At the same time, RealTimeSR provides a value of approximately 33.21 PSNR metrics on this set, primarily due to the use of additional channel attention blocks.

The same dataset was also used to evaluate the processing speed of models in real time. This test did not include Real-ESRGAN, as it is not designed to work in real time.

Table 1

Comparison of model testing results on the Set14 dataset

Method	PSNR (RGB)	SSIM (RGB)
Bicubic	31.24	0.8625
Real-ESRGAN	29.15	0.7843
RT4KSR	33.51	0.9212
RealTimeSR	33.21	0.9116

Table 2 shows the average processing time per image and the corresponding frame rate obtained during testing. As can be seen from the table, the model built using the proposed RealTimeSR architecture has a shorter processing time and a higher frame rate per second compared to the RT4KSR architecture. This confirms that RealTimeSR is suitable for use in real-time mode.

Table 2

Comparison of processing time and frame rate during testing

Method	Processing speed (s)	Frames per second
RT4KSR	0.0425	24.17
RealTimeSR	0.0299	33.40

Fig. 2 shows an example of restoring a 720p 4K image using RealTimeSR: on the left is the low-resolution output, in the center is the result of the model's work, and on the right is the original 4K image. As you can see, the model successfully restores fine details (leaves, bridge, castle) without noticeable artifacts.

Fig. 2. Image restoration to 4K

To demonstrate the difference between the approaches, the results of other methods were reproduced. Fig. 3 shows a comparison of the outputs of several algorithms on the same input frames.

Real-ESRGAN restores very sharp details (e.g., leaves) significantly better than conventional interpolation, but also generates additional details compared to other methods. RT4KSR and RealTimeSR show almost identical results, although the former produces a visually better option.

Performance was also compared graphically: GPU load decreases almost proportionally to tile area, while accuracy remains stable at ~10–20% overlap. Quantization to FP16 format reduced processing time by approximately 1.8 times without a noticeable decrease in PSNR metrics. The transition to the INT8 format provides an approximately 1.5-fold increase in speed but requires careful calibration (the model's own PSNR dropped by only 0.2–0.3 dB after optimal post-training quantization (OPTQ)) [15].

The proposed modifications (temporal processing, dynamic scaling, quantization, simple blocks, temporal heuristics) to the basic architecture have made it possible to create a model that is close to real-time performance when playing 4K. At the same time, image quality remains among the highest in the category of "effective SR systems".

Conclusions. The article discusses issues of effective resolution enhancement of images and video sequences using deep learning. Existing architectures are described: Real-ESRGAN (for real images) and RT4KSR (for real-time operation up to 4K), their shortcomings and limitations in terms of resource usage and adaptability are identified. The proposed RealTimeSR architecture uses Channel Attention and pixel-shuffle operations, as well as five significant improvements: tile processing with overlap to reduce memory usage; dynamic selection of image magnification scale (x2, x3, x4) to ensure system flexibility; quantization to FP16 or INT8 formats for adaptation to mobile devices; use of simple blocks in the model (Ghost, MobileNet) to reduce the amount of computation; use of information from previous video frames to speed up their processing.

Experimental testing of the implemented model using RealTimeSR's own architecture showed that the proposed approach ensures its proximity to the base models in terms of PSNR and SSIM metrics at a significantly higher image processing speed. In particular, this approach achieves results that are better than bicubic scaling, close to the results of competing models Real-ESRGAN and RT4KSR, and this is done with resources suitable for practical implementation. In the future, we plan to fully integrate the BasicVSR architecture for video and deploy it on devices with limited resources.

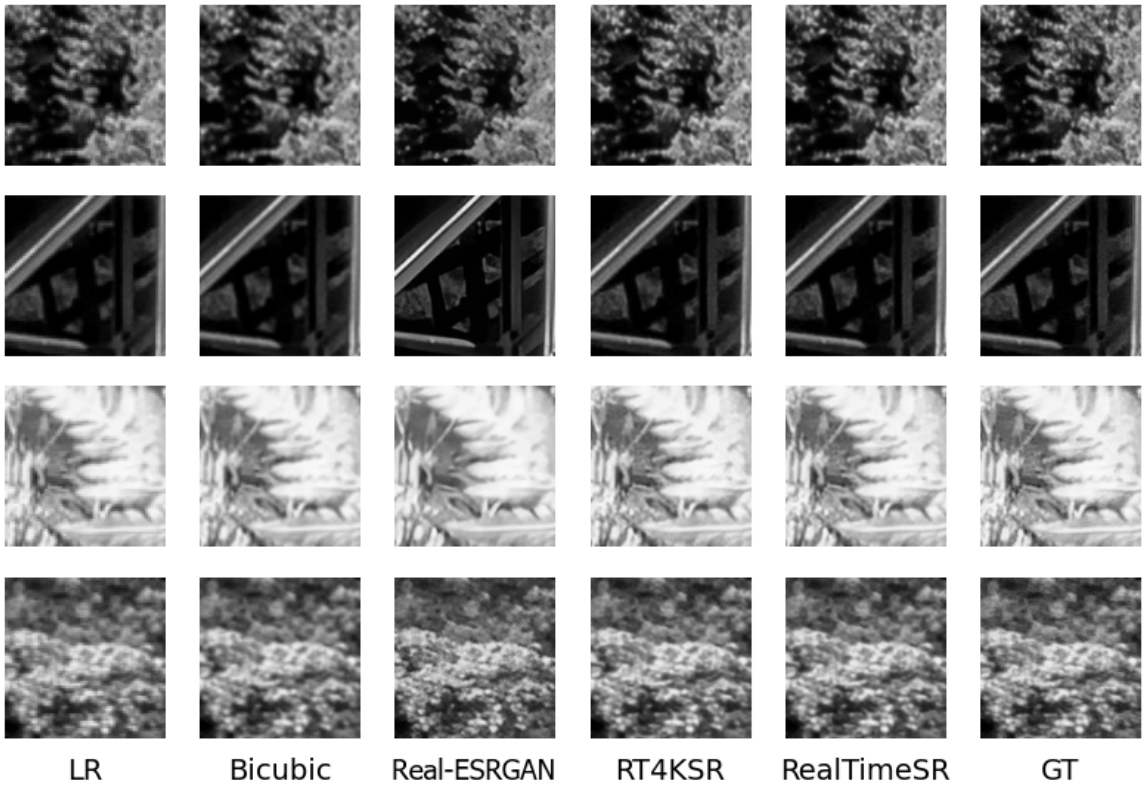

Fig. 3. Comparison of model performance results using different methods

Bibliography:

1. Image super-resolution: A comprehensive review, recent trends, challenges and applications / D. C. Lepcha et al. Information Fusion. 2023. Vol. 91. P. 230–260.
2. Arbitrary-Scale Video Super-Resolution with Structural and Textural Priors / W. Shang et al. arXiv. 2024. URL: https://arxiv.org/abs/2407.09919.
3. Unlocking Super-Resolution in Computer Vision. Number Analytics – AI Statistical Software. URL: https://www.numberanalytics.com/blog/ultimate-guide-super-resolution-computer-vision.
4. A single video super-resolution GAN for multiple downsampling operators based on pseudo-inverse image formation models / S. López-Tapia et al. Digital Signal Processing. 2020. Vol. 104. P. 102801. URL: https://doi.org/10.1016/j.dsp.2020.102801.
5. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data / X. Wang et al. arXiv. 2021. URL: https://arxiv.org/abs/2107.10833.
6. Towards Real-Time 4K Image Super-Resolution / E. Zamfir et al. CVPRW. 2023. URL: https://openaccess.thecvf.com/content/CVPR2023W/NTIRE/html/Zamfir_Towards_Real-Time_4K_Image_Super-Resolution_CVPRW_2023_paper.html.
7. Image Super-Resolution Using Very Deep Residual Channel Attention Networks / Y. Zhang et al. ECCV. 2018. URL: https://openaccess.thecvf.com/content_ECCV_2018/html/Yulun_Zhang_Image_Super-Resolution_Using_ECCV_2018_paper.html.
8. Combining Attention Module and Pixel Shuffle for License Plate Super-Resolution / V. Nascimento et al. arXiv. 2024. URL: https://arxiv.org/abs/2210.16836.
9. Majurski M., Bajcsy P. Exact Tile-Based Segmentation Inference for Images Larger than GPU Memory. *Journal of Research of the National Institute of Standards and Technology*. 2021. Vol. 126. URL: https://doi.org/10.6028/jres.126.009.
10. Muhammad W., Aramvith S. Multi-Scale Inception Based Super-Resolution Using Deep Learning Approach. *Electronics*. 2019. Vol. 8, no. 8. P. 892. URL: https://doi.org/10.3390/electronics8080892.
11. Towards Clip-Free Quantized Super-Resolution Networks: How to Tame Representative Images / A. Kalay et al. arXiv. 2023. URL: https://arxiv.org/abs/2308.11365.
12. GhostSR: Learning Ghost Features for Efficient Image Super-Resolution / Y. Nie et al. arXiv. 2021. URL: https://arxiv.org/abs/2101.08525.
13. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / A. Howard al. arXiv. 2017. URL: https://arxiv.org/abs/1704.04861.
14. BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond / K. Chan et al. arXiv. 2021. URL: https://arxiv.org/abs/2012.0218.
15. Crudu V. Boost TensorFlow Serving Performance with TensorRT Optimization. Custom software development company. URL: https://moldstud.com/articles/p-boost-tensorflow-serving-performance-with-tensorrt-optimization.

Костюченко А.В., Кушнір Н.О., Локтікова Т.М. РОЗРОБКА ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНИХ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ПІДВИЩЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ЗОБРАЖЕНЬ І ВІДЕОПОСЛІДОВНОСТЕЙ

Підвищення роздільної здатності (Super-Resolution, SR) зображень та відеопослідовностей є фундаментальною задачею в галузі комп'ютерного зору, яка має на меті відновити високоякісні деталі з низькороздільних вхідних даних. Кінцевим результатом є версія вихідного зображення або відео з високою роздільною здатністю, що є ключовим для покращення візуальної якості, підвищення деталізації та збільшення точності алгоритмів комп'ютерного зору. Ця технологія має велике практичне значення, охоплюючи такі напрями, як медична візуалізація для підвищення точності діагнозів, супутникова зйомка для моніторингу екологічних змін, безпека та спостереження для покращення якості відеозаписів, цифрові медіа та побутова електроніка. За останні роки глибоке навчання, зокрема згорткові нейронні мережі (CNN) та генеративно змагальні мережі (GAN), призвело до значних проривів у цій галузі, дозволяючи досягти високої якості відновлення. У статті надано огляд існуючих архітектур SR, з детальним аналізом таких відомих моделей, як Real-ESRAN та RT4KSR, визначивши їхні переваги при застосуванні в реальних сценаріях. Із урахуванням відмінностей між синтетичними та реальними деградаціями, а також необхідністю балансу між якістю та обчислювальною ефективністю, пропонується модифікована спрощена архітектура для масштабування зображень в реальному часі до 4K, орієнтована на практичне розгортання. Архітектура включає п'ять основних удосконалень: тайлову обробку з перекриттям для зменшення артефактів на межах тайлів; динамічне масштабування зображень, яке адаптує операції під доступні ресурси; оптимізацію через квантування для зменшення обсягу пам'яті та обчислювальної складності; зручну заміну блоків відповідної моделі простішими аналогами без значних втрат якості; використання набору евристик для обробки відеопослідовностей, що забезпечує часову узгодженість та уникнення мерехтіння. Розглядаються можливі способи впровадження цих удосконалень, їхній вплив на продуктивність та якість зображень, а також методи оцінки результатів роботи. Запропонований підхід спрямований на подолання нинішніх обмежень та сприяння більшій зручності практичного розгортання систем збільшення зображень.

Ключові слова: *роздільна здатність, зображення, відео, обробка, масштабування, глибоке навчання, нейронні мережі.*

Дата надходження статті: 25.10.2025
Дата прийняття статті: 13.11.2025
Опубліковано: 30.12.2025