Lifelong VAEGAN for inpainting of damaged image regions

Victor Sineglazov[1,*,†], Yehor Khomik[1,†]

[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Beresteiskyi Ave., 37, Kyiv, 03056, Ukraine

Abstract

The work is devoted to the automatic restoration (inpainting) of hidden or damaged areas of images. To solve the problem, a hybrid generative-adversarial neural network Lifelong VAEGAN with a buffer of previous samples and a U-Net generator with skip connections and a self-attention mechanism is proposed. A review of modern methods of image inpainting and lifelong learning is conducted, and for the first time a modular Lifelong VAEGAN is proposed, which is capable of effectively restoring images thanks to a dual-encoder architecture and a self-attention block.

Keywords

Lifelong VAEGAN, self-attention, inpainting, latent space, replay, U-Net generator

1. Introduction

Artificial intelligence has become indispensable in scientific and engineering disciplines by improving the accuracy of UAV visual navigation and suppressing sensor noise [1, 2] ,by providing hybrid ensemble neural network architectures for advanced analytics [3], and by forming systematic taxonomies of multi-criteria optimization methods and neural network topologies that guide both fundamental research and practical development [4, 5].

In particular, image inpainting has a wide range of applications. For example, in satellite images, clouds often cover the surface of the image [6, 7]; there is also the problem of restoring old archival photographs that have been damaged over time [8] and restoring masked human faces [9]. Previously, it was necessary to retake pictures or redraw images manually, now all this can be automated thanks to artificial intelligence.

All these tasks share a common underlying principle. Any image restoration task requires embedding prior assumptions about the structure of the restored area. It is necessary to take into account the structure, spectral, semantic properties, and global consistency with the context.

Previously, images were restored in two main ways: by diffusing pixels from the edges of the intact area inward [10] and by copying similar fragments from entire areas [11, 12]. Such methods were good at taking local textures into account, but ignored the overall semantics of the images, which could lead to inconsistencies [13, 14]. Modern models have significantly improved the quality of automatic reconstruction. While generative adversarial networks (GANs) produce realistic structures, variational autoencoders (VAEs) provide stable learning and interpretable latent representations [15, 16]. Combining these approaches preserves good quality and semantic consistency of the reconstructed regions [17, 18].

CH&CMiGIN'25: Fourth International Conference on Cyber Hygiene & Conflict Management in Global Information Networks, June 20–22, 2025, Kyiv, Ukraine

*Corresponding author.

†These authors contributed equally.

svm@kai.edu.ua (V. Sineglazov); khomik.yehor@lll.kpi.ua (Y. Khomik)

0000-0002-3297-9060 (V. Sineglazov); 0009-0008-0467-9889 (Y. Khomik)

2. Lifelong VAEGAN architecture and features

In the literature, most image restoration models are trained on only one set of images, without further replenishment on others. With the popularization of artificial intelligence, it is very important that the user can restore any images, so the ability to perform lifelong learning is necessary. However, sequential training on new samples leads to catastrophic forgetting. When training on new data, the network forgets knowledge from previous domains [19, 20, 21].

Lifelong VAEGAN (L-VAEGAN) is a generative model that combines VAE and GAN architectures and provides the ability to train sequentially on multiple datasets [22]. L-VAEGAN, unlike previous generative replay methods that did not have a separate autoencoder, is able to build a common latent space for all domains.

The basic variables in the architecture of the L-VAEGAN model are:

- x is an input image from a specific domain. In continual learning, there is a sequence of image sets $\{X^1, X^2, \ldots, X^t\}$, each of which corresponds to a certain domain;
- z is a continuous latent variable that encodes the style or content of the image;
- c is a discrete latent variable that reflects the categorical content of the image. We do not use it in our implementation, but in many datasets, for example with animals, it is necessary for high-quality restoration;
- a is a discrete domain variable (dataset). In lifelong learning, the encoder estimates which domain x belongs to, and the generator reproduces the image, having received a certain a.

These variables build a probabilistic model of images from continuous factors such as style and object variations and discrete factors such as class and domain.

The main components of Lifelong VAEGAN are an inference encoder [23], a generator, a discriminator, and a generative replay mechanism. The inference encoder encodes the input image into a set of latent variables. For each input image, a vector of means $\mu_\phi(x)$ and variances $\sigma^2_\phi(x)$ is generated, which define the distribution $q_\phi(z \mid x)$. Next, a reparameterisation trick is used to select the value z for the generator. In general, the L-VAEGAN generator is a decoder that builds images from latent variables. In the context of VAE, the generator reconstructs the original image from the code, in the context of GAN, it generates images that are so realistic that the discriminator cannot distinguish them from the real ones. However, in our implementation, the generator has an additional encoder that extracts the usual spatial features. The discriminator receives real or generated images as input and learns to distinguish them. It is a critic of the generator.Thus, the generator's task is to produce an image that the discriminator cannot distinguish from real ones, and the discriminator's task is to distinguish between real and generated images.

Another important mechanism of L-VAEGAN is generative replay. During replay, the generator reproduces samples from previous domains, thus the model does not forget them (Figure 1).

However, in our implementation, generative replay was abandoned, and a buffer of previous samples was used [24, 25].

Although the original Lifelong VAEGAN documentation stated the inefficiency of the buffer [22], in the inpainting task it showed better results. Instead of restoring damaged generated images, a small sample of real ones was used. When training on multiple datasets, no memory overrun problems were observed, but the GPU overhead was lower. One of the reasons for abandoning generative replay is its tendency to produce less diverse images due to the shift toward the new domain. With generative replay, the more training epochs there are, the poorer the samples the model can reproduce, which impairs the retention of prior knowledge. However, when training on a large number of different domains, generative replay may have better results. Figure 2 shows the architecture of the Lifelong VAEGAN model.

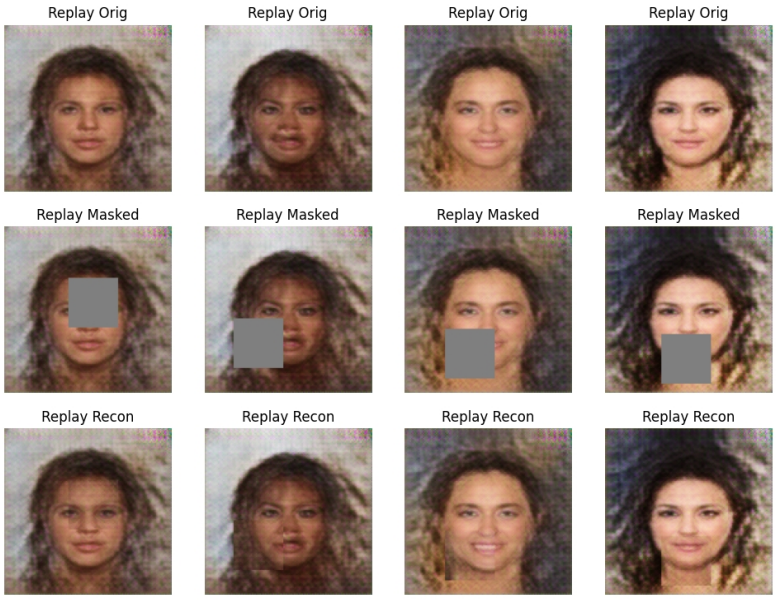

Figure 1: Images created during generative replay.

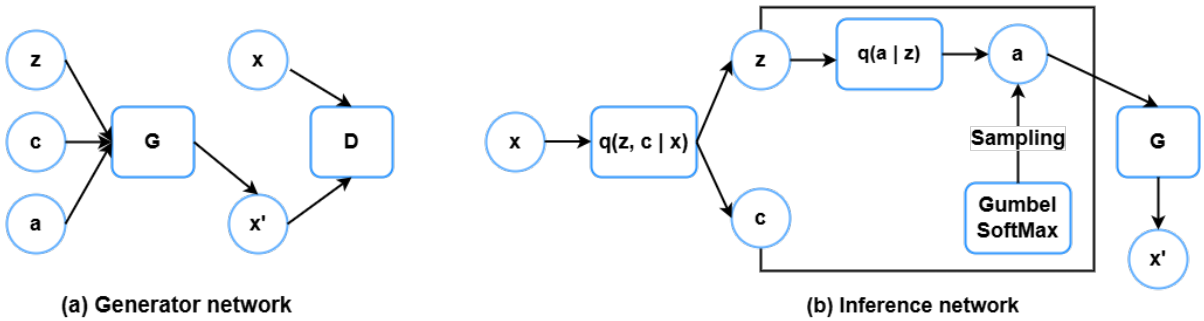

Figure 2: Architecture of Lifelong VAEGAN.

The left part (Figure 2a) shows the generator and discriminator, which are trained adversarially. The right part (Figure 2b) shows the inference network [22]. The encoder estimates the distribution $q_\phi(z \mid x)$ and the domain classifier $q_\phi(a \mid z)$.

However, in our implementation, the domain variable is estimated directly from the input image $q_\phi(a \mid x)$. The generator produces a restored image x', which is then compared by the discriminator with the original images.

The Lifelong VAEGAN model has two learning phases - Wake and Dreaming [26]. In the first, the generator and discriminator are trained adversarially, in the second, the variational autoencoder part is updated on real and generated data. In the Wake phase, the network is trained using the Wasserstein GAN loss with gradient penalty (WGAN-GP). The loss function is shown in equation (1).

$$\min_G \max_D L_{\text{GAN}}(\theta, \omega) = \mathbb{E}_{z\sim p(z),\, c\sim p(c),\, a\sim p(a)}\big[D\big(G(c,z,a)\big)\big] \;-\; \mathbb{E}_{x\sim p(x)}\big[D(x)\big] + \lambda\, \mathbb{E}_{\tilde{x}\sim p(\tilde{x})}\big(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1\big)^2 \tag{1}$$

where:

- $G(c, z, a)$ is the generated image;

- $D(\cdot)$ is the discriminator output;
- $p(z)$ is the prior distribution of the latent vector z;
- $p(c), p(a)$ are the prior distributions of the discrete variables c and a;
- Gradient penalty limits the discriminator's gradient and prevents mode collapse.

In the Dreaming phase, the model learns to maximize the log-likelihood of the training data by optimizing the ELBO, which is calculated by the formula (2).

$$L_{\text{VAE}}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z,c,a|x)}\big[\log p_\theta(x \mid z, c, a)\big] \; - \; D_{\text{KL}}\big[q_\phi(z, c, a \mid x) \,\|\, p(z, c, a)\big]. \tag{2}$$

where:

- $\mathbb{E}_{q_\phi(z,c,a|x)}[\log p_\theta(x \mid z, c, a)]$ estimates the reconstruction accuracy;
- the second term (KL divergence) acts as a regularization, preventing overfitting and ensuring a smooth latent space.

3. Proposed model architecture

The visual results show that the basic L-VAEGAN model can restore the damaged area. Figure 3 confirms that the model correctly captures the domain and main facial features.

Figure 3: Restoration results using the baseline L-VAEGAN.

However, factors such as hair, skin, and eye color do not quite correspond to the original area. It can be concluded that the basic model has limitations in capturing various small details. To eliminate the shortcomings, the following modifications were implemented.

A U-Net generator was implemented, which forms a dual-encoder architecture. The model has both a separate inference encoder and a generator with the U-Net architecture, which also has an encoder that "extracts" spatial features. Figure 4 presents a general diagram of the updated architecture.

The model has two processing paths. The inference encoder compresses the image to the latent vector z and determines the domain label a. The U-Net generator encoder compresses the image, extracting multi-level features. At the bottleneck level, the concatenation of $[z, a]$ occurs, their transformation

through a fully connected layer into a tensor of the desired size and the addition of spatial features. Also at the bottleneck level, a self-attention block is added. After that, the decoder performs upsampling to the original size taking into account skip connections.

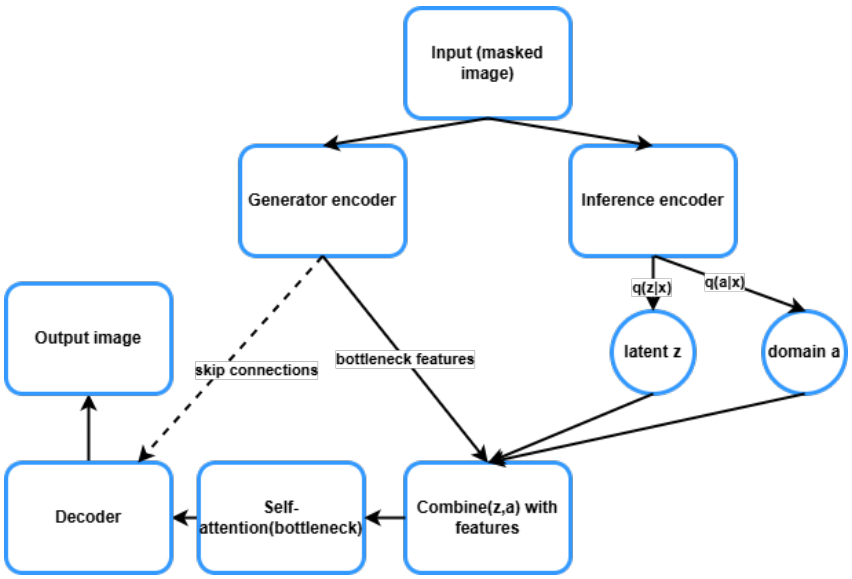

Figure 4: Model architecture with two encoders.

3.1. U-Net generator with two inputs

The generator based on the U-Net architecture has a symmetric encoder-decoder path with skip connections [27]. The encoder part of the generator consists of a cascade of Conv→Norm→ReLU layers, which reduce the image dimension to $4 \times 4 \times 512$. In our implementation, the generator has five encoder and decoder blocks. Since the input images had a size of 128×128, at the fifth level the feature size becomes 4×4, further reducing it is inefficient, since such actions can lead to the loss of information about the spatial location of objects. If, on the contrary, the network depth is made smaller, the model generates more generalized and inconsistent objects.

The inference encoder specifies the vector z and a. Both encoders converge at the bottleneck level. The following approach was used to combine information: a continuous vector z is concatenated with a vector of the domain variable a (one-hot representation of the domain [28]. Then the resulting vector passes through a fully connected layer, which expands it into a $4 \times 4 \times 512$ tensor with subsequent ReLU activation. The resulting tensor is treated as additional noise, then it is passed element by element to the output bottleneck representation of the U-Net encoder. Such actions are a form of information fusion, where the latent vector brings semantic features, and spatial features provide local context. As a result, the enriched tensor is fed to the decoding part of the generator.

3.2. Decoder and skip connections

The decoder performs sequential upsampling. Transposed convolutions (ConvTranspose2d) restore the spatial size of the layer first to 8×8, then to 16×16 and so on up to the original size of 128×128. At each level, the decoder receives a skip connection from the corresponding encoder layer of the generator. Skip connections are very important for the image reconstruction task. Without their use, the generator would reconstruct the image based only on the low-dimensional representation in the bottleneck, which would lead to the loss of many structures and blurring of the result. Thanks to skip connections, the decoder directly receives high-resolution features from the encoder that would be lost during downsampling. In our implementation, skip connections are performed as a concatenation of

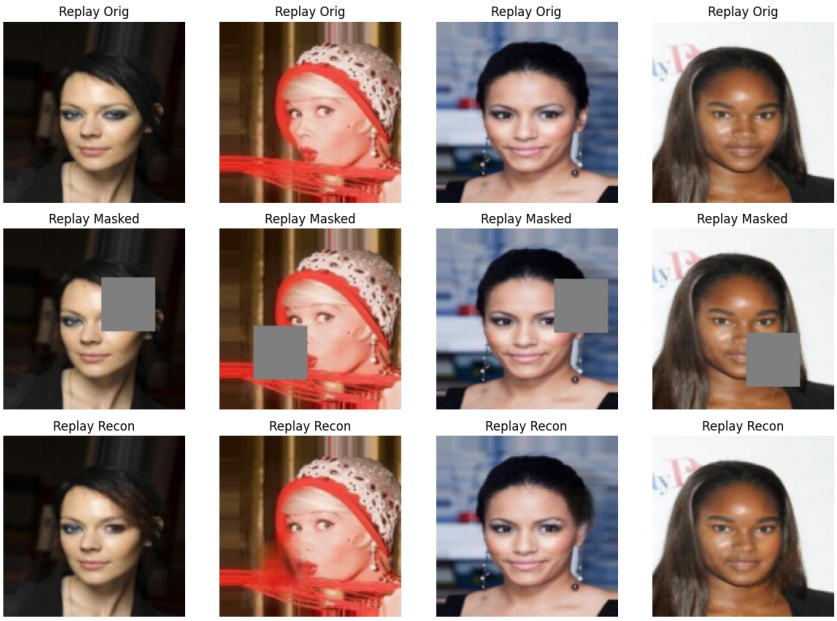

Figure 5: Images generated during replay with a buffer of previous samples.

tensors. The output activations of the encoder blocks are combined by channels with the corresponding decoder tensor after upsampling to the appropriate size.

As a result, even considering that the latent vector z carries only the global information of the image, local details are restored thanks to skip connections, which makes the reconstruction more consistent.

At the output of the decoder, a filled image x' is obtained, which realistically fills the masked areas. As a result, thanks to the variational encoder, the images have high semantic consistency, and thanks to U-Net+GAN they are reproduced with high reliability.

3.3. Self-attention module

To improve the restoration of damaged areas, the self-attention mechanism was added to the generator in the bottleneck zone. At this stage, the spatial size is very small, but the channel depth is high, so each of the feature map elements contains information about a large area of the original sample. This makes it possible to take into account long-range relationships between different parts of the image and to coordinate them with each other. This is quite important for the continuation of the background and symmetry of different parts of the objects. After adding self-attention, brightness mismatches across a face were greatly reduced.

3.4. Wake and dreaming phases

During the Wake phase, the model receives real data from the current domain. The inference encoder calculates z and a, while the generator restores damaged areas, acquiring new knowledge. The parameters of the inference encoder and generator are updated based on the gradients from the reconstruction and adversarial losses (the discriminator is also updated). During each iteration, examples of old domains from the image buffer are mixed in [29], so we can say that the model "dreams" of past experiences within the main learning phase. In our implementation, both encoders and the generator are involved in the Wake phase, but are trained on a combination of new and buffer data, performing the role of consolidation (dreaming) [30]. This approach is similar to the idea of off-line sleep in neural networks, where during learning the model switches between repeating external cues and replaying internal memories. Our result demonstrates that even with the buffering approach the model prevents the forgetting of old skills. Figure 5 illustrates this.

3.5. Adding perceptual loss

During training, an additional perceptual loss was used, which is calculated not at the pixel level, but at the level of features extracted by the VGG16 model pre-trained on ImageNet. Minimizing only MSE tends to blur fine textures by averaging them. For human perception, images can be similar if they have the same structure, even if they are different pixel by pixel. Perceptual loss compares the outputs of certain internal layers, and not pixel values. Layers were selected that cover both low-level and high-level features. The original and generated images are passed through VGG16 and for each layer, the feature tensors $\phi_l(I_{\text{orig}})$ and $\phi_l(I_{\text{gen}})$ are calculated. Perceptual loss is calculated as the average L_1-norm of the difference of these features, summed over all selected layers, formula (3):

$$L_{\text{perc}}(I_{\text{orig}}, I_{\text{gen}}) = \sum_{l \in \mathcal{L}} \lambda_l \left\| \phi_l(I_{\text{gen}}) \; - \; \phi_l(I_{\text{orig}}) \right\|_1, \tag{3}$$

where:

- $\mathcal{L}$ is the set of indices of the selected layers;
- λ_l are the weighting factors for each layer.

As a result, thanks to the perceptual loss, results were obtained that are of higher quality for human perception, without artifacts and with correct textures.

4. Image-quality assessment metrics

The quality of the restored images was described through the following indicators: MSE, PSNR, SSIM and FID. None of the metrics covers all aspects of perception. Therefore, for better objectivity, one metric is not enough.

The mean square error (MSE) is defined as the arithmetic mean of the squares of the difference between the pixels of the original and reconstructed image, formula (4).

$$\text{MSE}(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^{N} \big(x_i - \hat{x}_i\big)^2, \tag{4}$$

where:

- x_i is the value of the i-th pixel of the original image;
- $\hat{x}_i$ is the value of the corresponding i-th pixel of the reconstructed image;
- N is the total number of pixels in the image.

However, MSE does not take into account the peculiarities of human perception. For this problem, the peak signal-to-noise ratio (PSNR) was used. This metric shows how much the maximum signal exceeds the existing error level, formula (5).

$$\text{PSNR}(x, \hat{x}) = 10 \, \log_{10}\!\left(\frac{{\text{MAX}_\text{I}}^2}{\text{MSE}(x, \hat{x})} \right) \text{dB}, \tag{5}$$

where:

- MAX_I is the maximum possible signal value;
- $\log_{10}$ is a decimal logarithm that converts the linear scale to a scale close to human perception.

However, MSE and PSNR only take into account pixel differences. To assess structural similarity, the SSIM metric was used. This metric effectively takes into account changes that humans see (contrast, texture). It is based on comparing the means, variances and covariances of local blocks of the original (x) and reconstructed (y) images, formula (6)

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{6}$$

where:

- μ_x, μ_y are the mean values of pixels in the local window for images x and y;
- σ_x^2, σ_y^2 are the variances of pixel values in the window for x and y;
- σ_{xy} is the covariance between corresponding pixels of the two images;
- C_1, C_2 are small positive constants that avoid division by 0.

Frechet Inception Distance (FID) was also used, which compares statistical properties. Inception v3 neural network was used, which receives feature vectors that are compared as two Gaussian distributions, formula (7)

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 \; + \; \mathrm{Tr}\Big(\Sigma_r + \Sigma_g - 2\,(\Sigma_r\,\Sigma_g)^{1/2}\Big), \tag{7}$$

where:

- μ_r, μ_g - average values of feature vectors for real and generated images;
- Σ_r, Σ_g - covariance matrices of these images;
- $\|\cdot\|_2$ - Euclidean norm;
- Tr - trace of the matrix (sum of elements on the main diagonal).

5. Results and discussion

The model was trained on two datasets: first on CelebA, then on Facade.

The following results were obtained on CelebA: MSE = 0.205, PSNR = 25.65, SSIM = 0.9139, FID = 6.81. According to the SSIM indicator, it is clear that the model perfectly preserves the key structural features of the faces. FID indicates good generation quality: the generated images almost do not differ from the real ones in terms of their features. The visual results of CelebA restoration are shown in Figure 6.

Figure 6: Visual restoration results on the CelebA dataset.

On the Facade dataset, which contains a relatively small number of images, the results obtained are: MSE = 0.303, PSNR = 23.94 dB, SSIM = 0.8643, FID = 44.24.

Given the small amount of training data, these results are quite high.

The visual results of Facade restoration are shown in Figure 7.

Figure 7: Visual restoration results on the Facade dataset.

6. Conclusions

An intelligent system for restoring damaged parts of images based on the Lifelong VAEGAN model has been developed. The basic architecture has been extended to a dual-encoder architecture. A separate inference encoder forms the latent vector, and a generator encoder with a U-Net structure preserves spatial features. Then both encoders are combined into a bottleneck with a self-attention mechanism. Thanks to skip connections, high-resolution details are transmitted to the decoder, while self-attention better matches distant image areas. For lifelong training, a real-sample buffer was used, which mixes data from previous domains into the current ones. Due to this, catastrophic forgetting was avoided. Perceptual loss on VGG16 features was also added, which eliminates blurring and improves textural plausibility. Results on the CelebA and Facade sets confirmed the effectiveness of the model. he results (MSE = 0.205, PSNR = 25.65, SSIM = 0.9139, FID = 6.81) after training on the new domain are quite convincing and show that catastrophic forgetting was avoided.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] V. M. Sineglazov, V. S. Ishchenko, Intelligent visual navigation system of high accuracy, in: 2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2019, pp. 123–127. doi:10.1109/APUAVD47061.2019.8943916.

[2] M. G. Lutsky, V. M. Sineglazov, V. S. Ishchenko, Suppression of noise in visual navigation systems, in: IEEE 6th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), 2021, pp. 7–10. doi:10.1109/APUAVD53804.2021.9615405.

[3] V. Sineglazov, A. Kot, Design of hybrid neural networks of the ensemble structure, Eastern-European Journal of Enterprise Technologies (2021) 31–45. doi:10.15587/1729-4061.2021.225301.

[4] M. Zgurovsky, V. Sineglazov, E. Chumachenko, Classification and analysis of multicriteria optimization methods, in: Studies in Computational Intelligence, volume 904, Springer, 2021, pp. 59–174. doi:10.1007/978-3-030-48453-8_2.

[5] M. Zgurovsky, V. Sineglazov, E. Chumachenko, Classification and analysis topologies known

artificial neurons and networks, in: Studies in Computational Intelligence, volume 904, Springer, 2021, pp. 1–58. doi:10.1007/978-3-030-48453-8_1.

[6] M. Xu, F. Deng, S. Jia, X. Jia, A. J. Plaza, Attention mechanism-based generative adversarial networks for cloud removal in landsat images, Remote Sensing of Environment 271 (2022) 112902. doi:10.1016/j.rse.2022.112902.

[7] M. Czerkawski, P. Upadhyay, C. Davison, A. Werkmeister, J. Cardona, R. Atkinson, C. Michie, I. Andonovic, M. Macdonald, C. Tachtatzis, Deep internal learning for inpainting of cloud-affected regions in satellite imagery, Remote Sensing 14 (2022) 1342. doi:10.3390/rs14061342.

[8] C. Mendoza-Dávila, D. Porta-Montes, W. Ugarte, Photorestorer: Restoration of old or damaged portraits with deep learning, in: Proceedings of the 19th International Conference on Web Information Systems and Technologies (WEBIST 2023), 2023, pp. 104–112. doi:10.5220/0012190000003584.

[9] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, J. Jia, MAT: Mask-aware transformer for large-hole image inpainting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10758–10768. doi:10.1109/CVPR52688.2022.01049.

[10] M. Bertalmío, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00), 2000, pp. 417–424. doi:10.1145/344779.344972.

[11] A. Criminisi, P. Pérez, K. Toyama, Object removal by exemplar-based inpainting, in: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03), 2003, pp. 721–728. doi:10.1109/CVPR.2003.1211538.

[12] I. Ostroumov, et al., Modelling and simulation of DME navigation global service volume, Advances in Space Research 68 (2021) 3495–3507. doi:10.1016/j.asr.2021.06.027.

[13] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, A. A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2536–2544. doi:10.1109/CVPR.2016.278.

[14] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Transactions on Graphics 36 (2017) 107:1–107:14. doi:10.1145/3072959.3073659.

[15] D. P. Kingma, M. Welling, Auto-encoding variational bayes, 2013. doi:10.48550/arXiv.1312.6114. arXiv:1312.6114.

[16] F. Yanovsky, Inferring microstructure and turbulence properties in rain through observations and simulations of signal spectra measured with doppler-polarimetric radars, in: NATO Science for Peace and Security Series C: Environmental Security, volume 117, 2011, pp. 501–542. doi:10.1007/978-94-007-1636-0_19.

[17] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: Proceedings of the 33rd International Conference on Machine Learning (ICML), 2016, pp. 1558–1566. doi:10.48550/arXiv.1512.09300, arXiv.1512.09300.

[18] N. Ruzhentsev, et al., Radio-heat contrasts of UAVs and their weather variability at 12 ghz, 20 ghz, 34 ghz, and 94 ghz frequencies, ECTI Transactions on Electrical Engineering, Electronics, and Communications 20 (2022) 163–173. doi:10.37936/ecti-eec.2022202.246878.

[19] T. Lesort, H. Caselles-Dupré, M. Garcia-Ortiz, A. Stoian, D. Filliat, Generative models from the perspective of continual learning, 2018. doi:10.48550/arXiv.1812.09111. arXiv:1812.09111.

[20] H. Shin, J. K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: Proceedings of the 30th Conference on Neural Information Processing Systems (NeurIPS 2017), 2017, pp. 2990–2999. doi:10.48550/arXiv.1705.08690.

[21] K. Dergachov, et al., GPS usage analysis for angular orientation practical tasks solving, in: Proceedings of the IEEE 9th International Conference on Problems of Infocommunications Science and Technology (PICST), 2022, pp. 187–192. doi:10.1109/PICST57299.2022.10238629.

[22] F. Ye, A. G. Bors, Learning latent representations across multiple data domains using lifelong VAEGAN, in: Proceedings of the European Conference on Computer Vision (ECCV 2020), volume LNCS 12365, Springer, 2020, pp. 777–795. doi:10.1007/978-3-030-58565-5_46.

[23] J. Zhu, D. Zhao, B. Zhang, B. Zhou, Disentangled inference for gans with latently invertible autoencoder, International Journal of Computer Vision 130 (2022) 1259–1276. doi:10.1007/

s11263-022-01598-5.

[24] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, J. Choi, Rainbow memory: Continual learning with a memory of diverse samples, in: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 8218–8227. doi:10.1109/CVPR46437.2021.00812.

[25] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. S. Torr, M. Ranzato, On tiny episodic memories in continual learning, 2019. doi:10.48550/arXiv.1902.10486. arXiv:1902.10486.

[26] G. E. Hinton, P. Dayan, B. J. Frey, R. M. Neal, The wake–sleep algorithm for unsupervised neural networks, Science 268 (1995) 1158–1161. doi:10.1126/science.7761831.

[27] L. Yin, W. Tao, D. Zhao, T. Ito, K. Osa, M. Kato, T.-W. Chen, Unet–: Memory-efficient and feature-enhanced network architecture based on u-net with reduced skip-connections, in: Proceedings of the 17th Asian Conference on Computer Vision (ACCV 2024), volume 15478, Springer, 2024, pp. 185–201. doi:10.1007/978-981-96-0963-5_11.

[28] S. Huang, C. He, R. Cheng, Sologan: Multi-domain multimodal unpaired image-to-image translation via a single generative adversarial network, IEEE Transactions on Artificial Intelligence 3 (2022) 722–737. doi:10.1109/TAI.2022.3187384.

[29] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, in: Advances in Neural Information Processing Systems 30 (NeurIPS 2017), ????, pp. 6467–6476. doi:10.48550/arXiv.1706.08840.

[30] A. Sorrenti, G. Bellitto, F. P. Salanitri, M. Pennisi, S. Palazzo, C. Spampinato, Wake-sleep consolidated learning, IEEE Transactions on Neural Networks and Learning Systems (2024) 1–12. doi:10.1109/TNNLS.2024.3458440.