Synthetic Image Generation With a Fine-Tuned Latent Diffusion Model for Organ on Chip Cell Image Classification

Maksims Ivanovs*[1], Laura Leja[1], Kārlis Zviedris[1], Roberts Rimsa[2], Karina Narbute[3], Valerija Movcana[3], Felikss Rumnieks[3], Arnis Strods[3], Kevin Gillois[3], Gatis Mozolevskis[2], Arturs Abols[2,3] and Roberts Kadiķis[1]

[1]Institute of Electronics and Computer Science, Riga, Latvia
[2]Cellbox Labs, Riga, Latvia
[3]Latvian Biomedical Research and Study Centre, Riga, Latvia
*Correspondence: maksims.ivanovs@edi.lv

***Abstract*—Augmentation of the datasets of authentic microscopy images with synthetic images is a promising solution to the problem of the limited availability of biomedical data for training deep neural network (DNN) based classifiers. In the present study, we use a text-to-image latent stable diffusion model fine-tuned by means of low-rank adaptation (LoRA) to augment a small dataset of the images of organ on chip cells. While the resulting synthetic images appear quite similar to the authentic images on which the low-rank adaptation was performed, we find that neither training the EfficientNetB7 DNN model solely on the synthetic data nor augmentation of the real-world dataset with different proportions (10, 25, 50, and 75 percent) of these data leads to the improvement of the accuracy of the model. The findings of our study suggest that a further exploration of the low-rank adaptation options is needed to fully use the capacity of latent diffusion models for the synthesis of biomedical images.**

I. Introduction

Deep neural networks (DNNs) have recently been successfully applied to a number of tasks for the digital processing and analysis of biomedical images such as segmentation of endoscopy and nuclei microscopy images [1] and classification of X-ray images [2]. Their main advantages are that they do not require manual feature engineering, as they learn from the training data on their own, and high accuracy: in particular, they have even outperformed human experts on some biomedical image classification tasks [3]. The main shortcomings of DNNs are that they typically require a lot of computing power, especially during training, their inner workings are not transparent, as they operate in a black box-like manner [4], and they need a lot of data for training [5]. The last problem is particularly topical for DNN applications to the biomedical image processing for two reasons: first, datasets of such images tend to be comparatively small; second, in multiclass classification tasks, some classes in biomedical datasets are typically underrepresented, as some conditions (e.g., a rare form of a tumour) are observed less frequently.

There are several approaches to tackling the problem of the availability of biomedical images. First, it is possible to use different data augmentation techniques such as image rotations, flipping and making images darker or lighter to increase the amount of the training data. Second, instead of training DNN from scratch, one can use transfer learning [5], that is, retrain (partly or fully) a DNN previously trained on a large dataset such as ImageNet [6] or COCO [7] on biomedical images, thus making use of the general visual representations that the model has already learned. Finally, a promising and rapidly developing possible solution is to generate synthetic biomedical images and either train DNN solely on them or use these images to augment real-world biomedical image datasets. A common approach to the generation of synthetic imagery is the use of Generative Adversarial Networks (GANs, [8]), yet they are known to be fragile and difficult to train [9]. Therefore, it is worth considering an alternative approach, namely, using one of the currently very popular large text-to-image models such as Midjourney [10], DALL-E [11] or Stable Diffusion [12], which have demonstrated an impressive capacity to generate synthetic data for various domains from cartoons to radiology [13] and have publicly available implementations that are quite easy to use. These large image synthesis models were not originally trained on the data that would make them capable of generating such domain-specific data as biomedical images, and retraining them from scratch would require a lot of computing resources and would be prohibitively expensive and time-consuming. However, it is possible to fine-tune them on a small amount of domain-specific data using consumer-grade hardware by means of using such recently developed methods for that as low-rank adaptation (LoRA, [14]).

The goal of the present study is to design a classifier for Organ on Chip (OOC) cell microscopy images. To do that, we train EfficientNetB7 [15] DNN model pretrained on ImageNet both on the real-world image dataset and on the mix of the real-world data and synthetic images that we generate with Stable Diffusion fine-tuned with LoRA. We propose two hypotheses, namely:

- *Hypothesis 1*: By using a DNN classifier, it is possible to

achieve better classification accuracy on the real-world microscopy OOC image dataset than that of a naive classifier;
- *Hypothesis 2*: It is possible to improve the classification accuracy on the given task by means of augmenting the real-world microscopy OOC image dataset with synthetic data generated with Stable Diffusion model fine-tuned by LoRA.

The rest of the paper is organised as follows. In Section II, we provide background information pertaining to our study; in Section III, we explain the methodology of our study; Section IV describes the results of the study and offers a discussion; finally, Section V outlines conclusions and suggests directions for future work.

II. Background

A. DNN for organ on a chip technology

Human organ on a chip (OOC) technology is meant to replicate the environment of particular human organs in vitro and therefore makes it potentially possible to create biomedical models suitable for testing drugs and investigating the behaviour of different pathogens in the host. To use OOC with cells originating from human organisms such as primary cells or differentiated iPSC (induced pluripotent stem cells) derived from a donor, it is necessary to optimise the cultivation conditions of the cells, which is a slow, costly, and failure-prone process. This process can be improved and automated by implementing an OOC cultivation system supervised by machine learning (ML) algorithms rather than a human, as that would potentially allow to reduce experiment time, failure rate, and the cost of the functioning of the system. One of the key requirements for the design of such a system is its capacity to classify the state of the cell culture on a chip (accessible via miscroscopy images from the camera) as 'good', 'acceptable', or 'bad', as depending on that the flow of the solution to the chip should be maintained or increased, or the experiment should be stopped altogether. As state-of-the-art results on image classification tasks are currently achieved with DNNs, it would be reasonable to use a DNN-based classifier a a part of the ML-based system in question. However, the issue of the availability of the data for training a DNN model inevitably arises, as due to the current throughput of the OOC systems, the number of the images for training and evaluating such a model is in the order of hundreds rather than thousands, and there are not any publicly available datasets suitable for that task either.

B. Synthetic data for training DNN

The use of synthetic data for training DNN has recently become increasingly popular. In the field of biomedical image processing, synthetic data holds the promise of allowing to create more data in case of the scarcity of the real data; that applies both to the size of datasets in general and to the amount of the data for underrepresented classes. The main challenge for the use of synthetic data is that it differs from the real-world data by being less photorealistic, which is known as the domain gap [16].

Some well-established methods of generating synthetic data are Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN). VAE [17] is a generative model that combines the autoencoder with the principles of Bayesian theory, allowing the development of complex generative data models by adapting them to large datasets. VAE have been successfully used to generate such biomedical data as brain magnetic resonance images (MRI) [18] and endoscopic images [19]; however, VAE also tend to suffer from producing blurry output because of learning non-informative latent codes [20] and unrealistic distributions of the prior vs posterior data [21]. The basic blocks of GAN architecture are two networks, a generator and a discriminator, which play a zero-sum game of respectively generating new synthetic samples and distinguishing them from the real data [22]. GANs have been successfully applied to a number of tasks in biomedical image processing such as blood cell image generation [23] and generation of the images of retinal blood vessels [24]. However, training GANs can be challenging due to the frequently occurring issues such as mode collapse, instability of the model, and non-convergence [9]. Therefore, while GANs (and to a lesser extent, VAE) remain a lively area of research with a substantial potential for applications in biomedicine, it would be beneficial to explore other robust and simple means of biomedical image synthesis.

C. Text-to-image models for data synthesis

Text-to-image models such as Midjourney [10], DALL-E [11] and Stable Diffusion [12] are currently one of the most trending topics in AI research. While their large size and the vast amount of the date needed to train them from zero make their development much less accessible than that of VAE or GAN, which are not particularly resource-demanding, they have rapidly become very popular due to the ease of use of the trained models. The out-of-the-box output of text-to-image models is not likely to be suitable for biomedical purposes, as biomedical data are too specific for these general-purpose models; however, it is possible to overcome these obstacles by fine-tuning text-to-image models for that purpose. In particular, the method of low-rank adaptation (LoRA, [14]) allows to fine-tune Stable Diffusion on a small (in the order of dozens) number of images and use the prompts passed to the model during the fine-tuning stage to generate new data.

D. Mathematical basis of diffusion models

Diffusion-based generative models are described by a Markov-type process to change multistage noise based on probabilistic models of diffusion (see [25]). The model is divided into two types of processes, forward noise process and reverse diffusion process variance, using reparameterization with a smaller forward magnitude. The mathematical principle is necessary to understand the complex distribution $q(x_0)$ that describes the pixel x_0 values at each position from the input

image set X_0. We can define a diffusion process to which noise is added at time t with variance $\beta_t \in (0, 1)$(forward process). Once we have obtained the T states of the Markov process, it is necessary to approximate $q(x_t)$ with the distribution $p(x_t)$ by cleaning it from noise (reverse process). Once the model is trained, we can take a dataset with complete noise, run it through the $p(x_{0:T})$ process, and thus obtain new images with X_0 features.

The forward process is defined as a Gaussian distribution $q(x_t|x_{t-1}) \sim N(\mu_t, \Sigma)$, where $\mu_t = \sqrt{1-\beta_t x_{t-1}}$ is the mean value, Σ is the sum over all variances. Markov process probabilities $q(x_k|x_{k-1})$ and $q(x_{k+1}|x_k)$ are independent of each other, it can be written as:

$$q(x_{1:T}) = \prod_{i=1}^{t} q(x_t|x_{t-1}), \tag{1}$$

where T is the final step. The reparameterization trick requires separating the stochastic part to obtain images at an arbitrary time step t without a loop [26]. It is necessary to isolate the stochastic component, which we denote by ϵ. At time step t, our image will look like this:

$$x_t = \sqrt{1-\beta_t x_t - 1} + \sqrt{\beta_t, \epsilon_{t-1}}. \tag{2}$$

After several transformations:

$$x_t = x_0 \sqrt{\prod_{i=1}^{t}(1-\beta_t)} + \epsilon \sqrt{1 - \prod_{i=1}^{t}(1-\beta_t)}. \tag{3}$$

The choice of variance β_t depends on the image resolution. The cosine schedule is slower with noise addition and will perform better than linear noise at low resolution [27]. The reverse process must start with the Gaussian noise obtained in the previous process in the last step. Next, a function has to be chosen to approximate the mean and variance of the distribution. Choosing a statistical distance between conditional distributions requires simpler expressions to calculate gradients easily during neural network training. Therefore, a Kullback-Leibler (KL) divergence is selected, which determines the measure of entropy - how much is lost when one distribution ($p(x)$) describes another ($q(x)$), estimating the mathematical expectation, and knowing the value of the density function at discrete points. In this case, the diffusion models have a loss function as KL divergence, a closed expression.

When rewriting the posterior probability $q(x_t|x_{t-1})$ in the backward direction $q(x_{t-1}|x_t)$, in order to compare with the approximation to use KL divergence, the condition on x_0 should be added. According to Bayes' theorem, the μ terms are obtained as follows:

$$\mu(x_t, t) = \frac{1}{\sqrt{1-\beta_t}}(x_t - \frac{1-(1-\beta_t)}{\sqrt{1-\prod_{i=1}^{t}(1-\beta_t)}}\epsilon_t). \tag{4}$$

The variance can be equal to the parameter from the forward process. $\Sigma(x_t, t)$. The loss function is:

$$L_t = D_{KL}[q(x_{t-1}|x_t, x_0)|p(x_{t-1}|x_t)], \tag{5}$$

where D_{KL} is KL divergence.

To fine-tune diffusion models, such methods as LoRA [14], which is used in our work, can be employed. LoRa works by converting the large matrix of parameters to a lower rank matrix and then modifying (i.e., fine-tuning) it; due to a much smaller size of the matrix than the original one, the operations are significantly less computationally expensive. More specifically, given the initial values of its parameters on pre-trained weights ϕ_0 and adjusted to $\phi_0 + \Delta\phi$: to maximize the objective:

$$\max \sum_{(x,y)} \sum_{t=1}^{|y|} log(p_\phi(y_t|x, y < t)) \tag{6}$$

where $p_\phi(y|x)$ is the pre-trained model. In case of diffusion models, LoRA specifically targets the attention layers in the U-Net network, as they connect the semantics of the prompts with the generative capacity of the model and therefore are particularly relevant for fine-tuning.

III. Methodology

A. Dataset of microscopy images

In our study, we used the dataset of microscopy images that we acquired in the course of in-vitro experiments with growing cells for OOC. The dataset was comparatively small in size, consisting of 822 images in JPG format, 32 of which were RGB images, while 790 images were grayscale images; the size of the images was 2048x1536 pixels (810 images) and 640x480 (12 images), and the bit depth was 24. The images were labelled by biomedical experts as belonging to one of the three classes: class 'good' with 500 images, class 'acceptable' with 212 images, and class 'bad' with 110 images. The uneven distribution of images among the different classes, in particular, the under-representation of the 'bad' class images, presented a challenge, as such an imbalance could potentially have an impact on the performance of the classification model.

The dataset had a unique specification because it comprised the images of cells of organs (lungs, intestines) taken from the time of seeding up to several months of cultivation. It included five distinct types of cells: Human umbilical vein endothelial cell line (HUVEC) type group represented by 16 images, all of which were 'good' class images; Human small airway epithelial cell line (HSAEC) type group consisted of 235 images distributed across 'good' (141 images), 'acceptable' (52 images), and 'bad' (42 images) classes; Human lung carcinoma cell line (A549) type group comprised 224 images with a distribution of 98 'good', 80 'acceptable', and 46 'bad'; human epithelial colorectal adenocarcinoma cell line (CACO) type group with 100 images with a distribution of 45 'good', 39 'acceptable', and 16 'bad'; lastly, Human Pulmonary Microvascular Endothelial Cells (HPMEC) type group, the largest in size, included 247 images, of which 200 were classified as 'good', 41 as 'acceptable', and 6 as 'bad'.

B. Synthetic data generation

The procedure for generating synthetic data involved splitting the dataset of the real-world images into 5 folds (see Section III-C) and using each fold for creating a LoRA model using a popular implementation[1] for that. LoRA models were generated with the following parameters: 2 repeats for each image, training for 10 epochs, training batch size equal to two, U-Net learning rate equal to 5E-4, text encoder learning rate equal to 1E-4. Created LoRA models were then used together with the Web UI implementation[2] of Stable Diffusion to generate synthetic data. Stable Diffusion Web UI was used with the following parameters: Euler A sampler, 20 sampling steps, CGF score of 7, random seed. We generated two datasets of synthetic images with these parameters, the difference between them being that for one dataset, we set the LoRA weight to 1.0, whereas for the other dataset we set that parameter to 0.8. The impact of the weight parameter is that a higher value results to a higher closeness of the generated images to the original ones, whereas a lower weight value implies a greater variability of the generated images in comparison to the original images that the LoRA model was generated on. Creating these two distinct datasets allowed us to explore the impact of different values of LoRA weights on the model performance. LoRA models were created in Google Colab environment[3], whereas synthetic images were generated with Web UI Stable Diffusion on a PC with 16 GB RAM, Intel i7-12700 CPU, NVIDIA RTX 3080 GPU with 10 GB VRAM, and Windows 11 OS.

C. Training and validating DNN

We conducted experiments training EfficientNet B7 [15] model available in Keras library pretrained on ImageNet [6] dataset. The architecture of the model after the modifications was as follows:

- the input layer with the resolution of 600x600 pixels;
- data augmentation layers (random rotations with the factor=0.25, random translations with the height factor=0.1 and width factor=0.1, random flips and the random contrast with the factor=0.1);
- basic EfficientNet B7 model with its weights frozen;
- GlobalAveragePooling2D layer;
- BatchNormalization layer with the dropout=0.2 applied to it;
- final Dense layer with 3 neurons with the softmax activation function.

The training consisted of 30 epochs with the Adam optimizer (learning rate=0.001) and sparse categorical crossentropy loss. The data for training was divided into 5 folds with one fold (different each time) used as a holdout fold for cross-validation. When the model was trained on the augmented dataset, it was strictly observed that the synthetic data was created using LoRA that was created only on the training data rather than on the data in the respective holdout fold; that was done in order to prevent the otherwise possible information leak from the training data to the validation data. Training the DNN model was done on a PC with 8 GB RAM, Intel i5-2500K CPU, NVIDIA RTX 3090 GPU with 24 GB VRAM, and Ubuntu 18.04.6 LTS OS.

[1]https://civitai.com/models/22530

[2]https://github.com/AUTOMATIC1111/stable-diffusion-webui

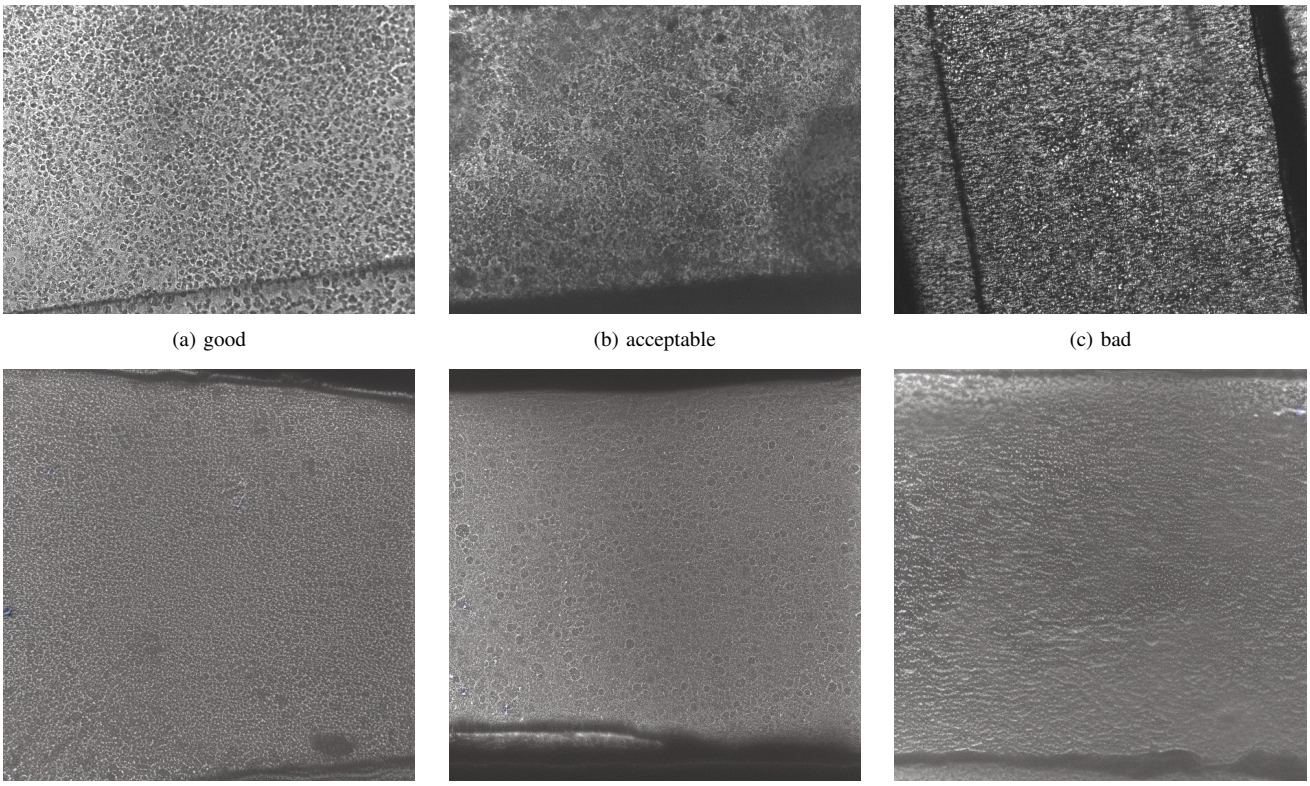
[3]https://colab.google/

IV. Results and discussion

The results of the study consist of the two main parts: first, we generated synthetic images of the cells; second, we trained a DNN on the datasets. The examples of the synthetic generated images are provided in the bottom row of Figure 1. As it can be seen, fine-tuning of Stable Diffusion with LoRA resulted in the images that look somewhat similar to their authentic counterparts: in particular, it can be see that the granularity of both authentic and synthetic 'good' images in the left column is different from that of the 'bad' images in the right column, which corresponds to the presence of the cells attached to the medium in the former case vs cells that did not attached to the medium in the latter case. However, due to the rather specific nature of the microscopy images, it is not entirely clear whether the degree of similarity is enough to ensure improved accuracy of DNN-based classifiers. Therefore, the evaluation of the quality of the generated images is based on the results of the experiments for the synthetic data with the LoRA weight of 1.0 reported in Table I and LoRA weight of 0.8 reported in Table II. As it can be seen, the results of the experiments confirm *Hypothesis 1*, as the accuracy of a naive classifier on the given dataset would be 60.8%, i.e., the percentage of the largest class, whereas our DNN model achieved the accuracy of 72.9%. However, the results of the experiments do not confirm *Hypothesis 2*, as augmentation of the real-world image dataset with synthetic images resulted in the deterioration rather than improvement of classification accuracy with the trend towards worse accuracy corresponding to the larger percentage of the data used for augmentation. A noteworthy trend is that the distribution of the synthetic data appears to be more similar between samples within each class than the respective distribution of the real-world data, as the DNN model converges much better when trained on the synthetic data than when trained on the real-world data or the augmented dataset. However, the distribution of the data between the corresponding classes of the real-world and synthetic data appears to be markedly different, which is particularly obvious in case of training the DNN model only on synthetic data: in that case, the accuracy of the trained model remains approximately on the level of a naive classifier and the model fails to learn representations from the data.

V. Conclusions and future work

The goal of this study was to develop a classification method for the Organ on Chip system. We proposed two hypotheses; *Hypothesis 1* was confirmed, as DNN trained on the dataset of the real-world microscopy images achieved the accuracy of classification of 72.9%, which is better than the accuracy of a naive classifier. However, *Hypothesis 2* was not confirmed,



(a) good (b) acceptable (c) bad

(d) good (e) acceptable (f) bad

Fig. 1: Examples of three data classes: good, acceptable, bad; a, b, c - original data images; d, e, f -synthetic images generated with Stable Diffusion fine-tuned on our dataset with LoRA.

TABLE I: Classification Results on Synthetic Data with LoRA weight=1.0. The best result for each metric is in bold.

Dataset	Accuracy	Precision	Recall
Baseline: Real-world	**0.729**	**0.731**	**0.715**
Real-world & 10% synthetic	0.721	0.729	0.701
Real-world & 25% synthetic	0.707	0.719	0.699
Real-world & 50% synthetic	0.710	0.720	0.697
Real-world & 75% synthetic	0.696	0.702	0.680
Real-world & 100% synthetic	0.699	0.707	0.687
Synthetic only (100%)	0.614	0.617	0.608

TABLE II: Classification Results on Synthetic Data with LoRA weight=0.8. The best result for each metric is in bold.

Dataset	Accuracy	Precision	Recall
Baseline: Real-world	**0.729**	**0.731**	**0.715**
Real-world & 10% synthetic	0.718	0.729	0.702
Real-world & 25% synthetic	0.704	0.709	0.69
Real-world & 50% synthetic	0.693	0.698	0.679
Real-world & 75% synthetic	0.707	0.718	0.696
Real-world & 100% synthetic	0.701	0.709	0.685
Synthetic only (100%)	0.621	0.63	0.595

as the augmentation of the real-world dataset with synthetic images resulted in the deterioration rather than improvement of the accuracy of the model. Therefore, we conclude that a further refinement of the fine-tuning of the Stable Diffusion model with LoRA on the cell microscopy images is needed. In particular, we intend to explore various LoRA parameters such as the number of epochs of training, the SimScore, and the number of steps of generating the images. We intend to do that by means of a large-scale search for the optimal parameters of fine-tuning. Furthermore, we intend to fine-tune models on more specific subsets of the available data: not just 'good', 'bad', and 'acceptable' classes, but also classes corresponding to specific cell types in our dataset.

ACKNOWLEDGMENT

This work was supported by the project 'AI-improved organ on chip cultivation for personalised medicine (AimOOC)' (contract with Central Finance and Contracting Agency of Republic of Latvia no. 1.1.1.1/21/A/079; the project is co-financed by REACT-EU funding for mitigating the consequences of the pandemic crisis).

References

[1] A. Iqbal, M. Sharif, M. A. Khan, W. Nisar, and M. Alhaisoni, "Ff-unet: A u-shaped deep convolutional neural network for multimodal biomedical image segmentation," *Cognitive Computation*, vol. 14, no. 4, pp. 1287–1302, 2022.

[2] S. Sharma, S. Gupta, D. Gupta, *et al.*, "Performance evaluation of the deep learning based convolutional neural network approach for the recognition of chest x-ray images," *Frontiers in oncology*, vol. 12, p. 3111, 2022.

[3] A. Rodriguez-Ruiz, K. Lång, A. Gubern-Merida, *et al.*, "Stand-alone artificial intelligence for breast cancer detection in mammography: Comparison with 101 radiologists," *JNCI: Journal of the National Cancer Institute*, vol. 111, no. 9, pp. 916–922, 2019.

[4] M. Ivanovs, R. Kadikis, and K. Ozols, "Perturbation-based methods for explaining deep neural networks: A survey," *Pattern Recognition Letters*, vol. 150, pp. 228–234, 2021.

[5] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: A literature review," *BMC medical imaging*, vol. 22, no. 1, p. 69, 2022.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[7] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.

[9] H. Chen, "Challenges and corresponding solutions of generative adversarial networks (gans): A survey study," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1827, 2021, p. 012 066.

[10] *Midjourney*, 2022. [Online]. Available: https://www.midjourney.com.

[11] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8821–8831.

[12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[13] H. Ali, S. Murad, and Z. Shah, "Spot the fake lungs: Generating synthetic medical images using neural diffusion models," *arXiv preprint arXiv:2211.00902*, 2022.

[14] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[16] S. I. Nikolenko, *Synthetic data for deep learning*. Springer, 2021, vol. 174.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] A. Volokitin, E. Erdil, N. Karani, *et al.*, "Modelling the distribution of 3d brain mri using a 2d slice vae," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*, Springer, 2020, pp. 657–666.

[19] D. E. Diamantis, P. Gatoula, and D. K. Iakovidis, "Endovae: Generating endoscopic images with a variational autoencoder," in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, IEEE, 2022, pp. 1–5.

[20] Y. Kim, S. Wiseman, A. Miller, D. Sontag, and A. Rush, "Semi-amortized variational autoencoders," in *International Conference on Machine Learning*, PMLR, 2018, pp. 2678–2687.

[21] J. Tomczak and M. Welling, "Vae with a vampprior," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 1214–1223.

[22] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[23] L. Ma, R. Shuai, X. Ran, W. Liu, and C. Ye, "Combining dc-gan with resnet for blood cell image classification," *Medical & biological engineering & computing*, vol. 58, pp. 1251–1264, 2020.

[24] T. Iqbal and H. Ali, "Generative adversarial network for medical images (mi-gan)," *Journal of medical systems*, vol. 42, pp. 1–11, 2018.

[25] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, PMLR, 2015, pp. 2256–2265.

[26] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[27] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8162–8171.