Earth Science Informatics (2025) 18:441
https://doi.org/10.1007/s12145-025-01934-6

RESEARCH

# Generating microscopic images of rocks using generative artificial intelligence (GenAI)

**Mariusz Młynarczuk**[1] · **Magdalena Habrat**[1]

Received: 14 February 2025 / Accepted: 24 May 2025


## Abstract

The generation of synthetic images can be an important element in supporting the augmentation and analysis of multimedia data. It has applications in many scientific fields. Also, in geological and mining sciences. This study presents generative artificial intelligence approaches, particularly on Generative Adversarial Networks (GANs) and diffusion models (Stable Diffusion), as widely used techniques for generating new data based on existing training datasets. The performance of these algorithms and the results obtained both with and without transfer learning, using local resources as well as commercial solutions offering high resolution of the generated images, are presented. Results are presented from text to image, image(s) to image, and text/image(s) to image scenarios. Local model training, transfer learning based on a predefined model, and transfer learning using commercial tools were used. The results indicate that the choice of architecture and model significantly influences the quality of generated images, ranging from visuals that differ from real-world data to high-resolution representations that are nearly indistinguishable from original samples. As a result of this work, the possibilities of generating synthetic data as a tool to support geological and mining research were presented, considering the technological and practical aspects of implementing these solutions.

**Keywords** Generative Artificial Intelligence · Geological images · Microscopic observations

## Introduction

Microscopic imaging of rocks is a fundamental component of geological research, enabling the identification and characterization of various geological formations and mineral constituents. These techniques are widely employed in both quantitative and qualitative rock analyses. Contemporary studies in this field are increasingly supported by automated algorithms, including image analysis methods and artificial intelligence (AI) techniques (Izadi et al. 2017, Marmo et al. 2005, Li et al. 2017, Singh et al. 2010). Data analysis systems, understood as integrations of algorithms, data processing modules (Gerard et al. 1992, Młynarczuk 2010, Ładniak and Młynarczuk 2015, Habrat and Młynarczuk 2018), and machine learning techniques, are designed to address complex research challenges. Neural networks, as mathematical models inspired by the functioning of biological neurons, play a crucial role in this domain (Long et al. 2022, National Academies of Sciences 2020, Izadi et al. 2013). These networks consist of nodes (neurons), connections (weights), and activation functions, collectively enabling pattern recognition, decision-making, and data generation. As a practical implementation of neural network architecture, an artificial intelligence model learns from training data, capturing relationships between input data and expected outputs. An essential phase in machine learning is the proper preparation of training and testing datasets, which, following transformation, constitute the foundation for effective practical machine inference. This process may involve training a model from scratch or utilizing transfer learning. It is a technique in which knowledge acquired from one dataset or task is leveraged to enhance model performance in a new, related context (Li et al. 2017). The applications of machine learning models are diverse and include image recognition (e.g., object identification in photographs), natural language processing (e.g., translation, sentiment analysis), forecasting,

Communicated by Hassan Babaie.

✉ Mariusz Młynarczuk
mlynar@agh.edu.pl

[1] Faculty of Geology, Geophysics and Environmental Protection, AGH University of Krakow, Al. Mickiewicza 30, 30-059 Kraków, Poland

Published online: 07 June 2025

and the generation of novel, previously unknown data (e.g., image, text, or audio creation). The latter application has gained significant popularity under the concept of generative artificial intelligence (GenAI, GAI), a technology that enables the creation of new, original content based on existing data. GenAI is applied to generate texts, images, audio, video, and computer code. The synthetic data produced by GenAI can simulate the properties of real-world data, supporting research and analytical processes (Kaswan et al. 2023, Bengesi et al. 2024, Sengar et al. 2024).

In geological sciences, the application of synthetic data can be particularly significant, not only in linguistic data processing (Hadid et al. 2024) but also in image-based analyses, especially when access to real-world data is limited. Consequently, contemporary scientific research often relies on analyzing complex spatial and image-based data, such as seismic data, three-dimensional models of geological structures, geochemical information, and remote sensing imagery. However, the processing and interpretation of such data involve several substantial challenges. Key challenges include the limited availability of high-quality field data, particularly in remote or deep-seated regions, and the rarity of atypical geological structures. Moreover, acquiring and processing empirical data is often costly and time-consuming. Generative models enable synthetic data creation that realistically replicates complex and rare geological configurations, such as faults, unconventional layer sequences, or concealed mineral deposits. This facilitates more effective training and testing of interpretative algorithms, thereby allowing for a deeper understanding of geological processes and enhancing the efficiency of analytical procedures (Abdellatif et al. 2022; Hadid, 2024). For instance, a practical application can be found in the work of Qaderi (Qaderi et al. 2025), where a data augmentation method based on a Deep Convolutional Generative Adversarial Network (DCGAN) was employed. This approach enabled the generation of a mineralization map that covered a greater proportion of known deposits while simultaneously reducing the analyzed area. Notably, full coverage of deposits was achieved within only 18% of the study area. This facilitated the identification of factors controlling ore formation and allowed for the optimization of exploration projects in terms of cost and efficiency. The study demonstrates the effectiveness of generative models and their positive impact on enhancing geological analyses and exploration outcomes (Qaderi et al. 2025). Synthetic data can also support the modeling of geological and geomatic processes, as well as the study of phenomena that are difficult to observe, such as rock formation or ore deposit generation (Zhang et al. 2022, Pierdicca and Paolanti 2022). Synthetic data is also widely used in training machine learning algorithms as a common step in data augmentation to enhance data diversity and representativeness. New methods for data generation are emerging (Ferreira et al. 2022), along with entire synthetic datasets (Nathanail 2023), which have been successfully applied in various studies, for instance, in the classification of Geo-Fossils (Saif et al. 2025). As a result, algorithms can be effectively trained even on relatively small datasets.

However, even though synthetic datasets address the limited availability of labeled geological images, enabling improved training of machine learning models for tasks such as image classification, object detection, and automated interpretation in geosciences, it is essential to note that their use also entails significant challenges. One of the key risks is the potential for misuse arising from the ease of data generation, which can lead to difficulties distinguishing real data from artificially generated content. Studies have highlighted, for example, the presence of biases in text-to-image models, revealing specific representational issues in the field of fluvial geomorphology. Images of rivers generated by the Stable Diffusion model exhibited strong preferences for certain morphological and environmental features, such as mountainous and forested landscapes, sunny weather, and summer settings. The authors rightly recommend additional verification of results in sensitive applications (Kupferschmidt et al. 2024). The study by Khanifar (2025) demonstrated that popular large language models (LLMs) provided correct answers to no more than 65% of the questions posed. Consequently, it is essential to ensure thorough documentation of data sources and acquisition methods for transparency and validation in scientific research. While using such data for scientific purposes yields auspicious results, enabling the rapid creation of datasets that may reflect reality, their actual impact on science requires further investigation. For this reason, in this research, the authors address this topic by presenting key aspects of synthetic geological image generation and showcasing selected concepts and results related to the application of generative artificial intelligence in creating microscopic rock images for geological purposes.

## Materials and methods

Artificial intelligence is a fast-growing field that is developing the automation of various business and scientific processes. A key aspect is machine learning, allowing computational models to learn from data instead of merely following predefined rules. AI architecture can be categorized by purpose, application, and techniques (Zhang et al. 2022). Several main strategies can be distinguished:

Rule-Based and Expert Systems – These systems rely on predefined rules established by experts and are commonly used in decision support systems.

Optimization and Heuristic Methods – This category includes optimization algorithms and metaheuristics, which are applied in tasks such as scheduling and resource

allocation (Hejducki and Podolski 2012, Piotrowski et al. 2012).

Machine Learning-Based Methods – divided into three main approaches: supervised learning, where the model is trained on prepared, labeled data (e.g., classifiers, decision trees, SVM, linear regression); unsupervised learning, allowing independent discovery of structures in unlabeled data, for example, through cluster analysis (k-means) or PCA; reinforcement learning, where an agent optimizes its actions in a dynamic environment to maximize rewards, as seen in algorithms such as Q-Learning or Deep Q-Networks (DQN). These approaches are often applied in data analysis, classification, prediction, and decision process optimization (Huang 2020). Deep learning architecture has gained popularity in recent years (Bashar 2019). Frequently used models include Convolutional Neural Networks (CNN) – applied in image processing, such as ResNet, GoogleNet (He et al. 2016, Szegedy et al. 2015, Krizhevsky et al. 2012), Recurrent Neural Networks (RNN) – adapted for sequential data analysis, such as time-series and linguistic data, e.g., LSTM, GRU (Hochreiter and Schmidhuber 1997, Cho et al. 2014), or generative methods (Radford 2015).

**Generative Algorithms** – enabling the generation of new data based on existing datasets, such as text, image, and video generation (Lv 2023, Uppalapati et al. 2024, Liu et al. 2023). These methods have gained popularity recently, especially following the widespread adoption of tools based on generative language models, such as Claude, ChatGPT, and Copilot. In each case, the prompt plays a crucial role (Liu et al. 2023), which defines how the model operates. In interaction with language models (e.g., ChatGPT), a prompt is an input text that the user provides to obtain a response. It can be a question, a command, or any other textual instruction that guides the model's behavior. Generative artificial intelligence can be applied in geological sciences, enabling the creation of synthetic geological data for research and analytical purposes. Among the popular generative image creation methods, several fundamental approaches can be distinguished (Durgadevi 2021):

- Generative Adversarial Networks (GANs) – consisting of two networks that compete: the generator (creates data) and the discriminator (evaluates the quality of the data) until the data meet specified conditions (e.g., loss function value). The well-known examples may be Vanilla GAN (as one of the basics architecture), Deep Convolutional GAN (DCGAN, the use of convolutional layers in the GAN architecture, improving the stability and quality of generated images), Conditional GAN (cGAN, an extension of GAN, in which the model conditions image generation on additional labels), and others, such as WGAN, Cycle GAN (Durgadevi 2021).
- Diffusion models (Bengesi et al. 2024, Ho et al. 2020, Lee et al. 2024, Chen et al. 2024, Yang et al. 2023, Gallon et al. 2024), which assume that the model learns to generate images through a training process that includes forward diffusion, where the model takes an image as input and iteratively adds noise to it. Observing changes in the image as noise is added allows the model to capture statistical patterns and dependencies in the data. Subsequently, reverse diffusion occurs, where the model aims to reconstruct the original, clean image from its noisy version. Since the model knows the noise added at each stage of the forward diffusion process, it can iteratively reverse this process. It predicts the added noise and subtracts it from the image, gradually removing noise and restoring the original image. The model learns to denoise images through precise noise prediction, which has attracted global attention due to its ability to generate images. Among popular techniques, notable examples include models capable of generating high-resolution images based on textual prompts (often referred to as Stable Diffusion models), frequently utilizing U-Net architecture (Gallon et al. 2024), Denoising Diffusion Probabilistic Models (DDPM). Models based on the diffusion process gradually reduce noise to generate images. Another well-known technology is Imagen (Google, a model generating images from text using diffusion techniques), DALLE (OpenAI, where models generate images based on textual descriptions using transformers and diffusion techniques). Unlike GANs, these models allow for multimodality, meaning they can create images based on textual prompts (they are also fine-tuned with images).

In the context of environments for generative artificial intelligence (GenAI) used for image generation, several fundamental approaches to model training can be distinguished, such as:

- **Local training on own data** – models can be trained (from scratch) in a closed local environment, such as on a personal computer or a private server, without transferring data to external servers. This approach ensures full control over the training process and safeguards data confidentiality; however, it requires significant computational resources and the customization of data pipelines.
- **Use of pre-trained models with the possibility of further customization—**users can use pre-trained models that can be further customized by training on their data, even in a local environment (known as transfer learning). Predefined models are often available on platforms such as GitHub or model repositories (e.g., Hugging Face, Kaggle). However, being aware of the license terms associated with their use is crucial.

- **Application of commercial solutions (e.g., cloud-based/web-based)** – some platforms offer the ability to train existing models further using advanced computational resources. However, it is crucial to consider that in such cases, the data sent to the system is stored and processed in the cloud, which involves transferring it to external computing resources and may have implications for data privacy and security.

These approaches allow image generation locally via self-training or transfer learning, or by using remote resources like computational power or ready-made content tools generation.

## Material

The study evaluates the potential for generating microscopic images of rocks. Images of dolomite from Rędziny were used as a sample training data set. The materials included in this study were recorded using a polarizing microscope under transmitted light. The images were captured at a resolution of $1200 \times 800$ with a constant tenfold magnification. Sample images of the original training data are shown in Fig. 1. The experiments used datasets ranging from 1, 50 images (Stable Diffusion or OpenArt) to 300 microscopic images (local training of GAN and Stable Diffusion).

## Methodology

This study examined the results of applying selected generative models in various configurations. The analyzed use cases (UC) include:

- UC1: Text-to-Image, generation of images based on textual prompts using pre-trained models without applying transfer learning. The experiments were conducted in commercial web-based environments, including DALLE (ChatGPT) and Stable Diffusion (OpenArt platform, StableDiffusion.com).
- UC2: Image-to-Image, generation of images based on training models from scratch in a local environment. The analyzed architectures were modeled after Vanilla GAN and DC-GAN and implemented in Python.
- UC3: Text/Image-to-Image, generation of images using transfer learning on pre-trained models such as Stable Diffusion. The experiments were conducted in a local environment (Python, UC 3.1) and on a commercial platform such as OpenArt (UC 3.2).

Custom model training, such as in the use cases UC2 (Image-to-Image) and UC3 (Text/Image-to-Image), may require several essential stages, including data preparation, environment configuration, and the training process itself, which involves:

1. Data collection: acquiring an appropriate dataset of images tailored to the target application of the model.
2. Model selection: defining the requirements and characteristics in the context of the analyzed problem.
3. Preparation of training data: loading and organizing data into the appropriate structure, augmenting and preprocessing the data according to the model's requirements, including, for example, scaling images to the required size (e.g., $256 \times 256$ pixels), normalizing pixel values (e.g., transforming the range to 0 to1 or $-1$to1).
4. Environment configuration: installation of necessary tools and dependencies (if needed), including e.g., installation of Python and creation of a virtual environment, installation of required libraries (torch, diffusers, transformers, etc.), checking GPU availability, due to high computational requirements, the use of GPU

**Fig. 1** A visual representation of microscopic rock images used as training data – Dolomite from Rędziny

resources is recommended to reduce model training time.
5. Model initialization: downloading and adapting the model (e.g., Stable Diffusion from Hugging Face), configuring model parameters, including fine-tuning, setting the number of epochs, mini-batch size, etc.
6. Training process: definition of the objective function and optimization method, iterative training process, including fine-tuning of the model, validation of the model on the test dataset, e.g., analysis of the consistency of generated images with descriptions (for text-to-image models), adjustment of parameters based on validation results, possibility of transfer learning, i.e., further fine-tuning of a pre-trained model on a new, smaller dataset to adapt it to specific applications, saving and potential reuse of the model.
7. Image generation: after completing the model training process, it is possible to generate new images, e.g., based on random noise (GAN) or noise and textual prompts (Stable Diffusion). The selected and prepared model transforms input data into output images according to the defined parameters.

Such a pipeline enables local model training to achieve specific outcomes. Particular attention is given to the fact that these workflows are highly parameterized and should be tailored to individual analytical objectives. As an example, implementation details are provided below (Table 1) in the form of pseudocode, along with training parameters (Table 2) used in the study of models based on Stable Diffusion (SD) and Generative Adversarial Networks (GAN/DCGAN).

## Results and discussion

### UC1. Generation of images based on textual prompts using pre-trained models without applying transfer learning (Text-to-Image)

The study utilized tools such as OpenArt, stablediffusionweb.com, and ChatGPT OpenAI, available free or through paid subscriptions, providing access to generative models trained on general datasets. The same prompt was used for testing:"A microscopic image of Dolomite rock, magnification ×20", applied across different models available within these tools. The objective was to simulate using these tools for generating geological images (e.g., as synthetic research data). The results generated by the selected models are presented in Fig. 2 (where the model's name is provided in the caption, followed by the name of the tool providing the model in parentheses). These results differ significantly from the sample real-world data (Fig. 1). While the generated images exhibit high technical quality, the absence of additional input data beyond the textual prompt, such as authentic images, precluded the application of transfer learning, resulting in outputs that are not visually similar to the expected microscopic images of rocks (Fig. 1).

Tools providing easy access to generative models often include configuration options that allow customization of the data generation process. One such feature is the automatic generation of a textual prompt based on an uploaded image, which can be used to create a new image. The algorithm automatically interprets the content of the input image (Fig. 3a) and presents the results in textual form (Fig. 3b). Based on this, it generates an image using the automatically generated prompt (Fig. 3c). The obtained results were not satisfactory; the generated prompt did not indicate the geological nature of the image, although the model recognized the microscopic scale of the depicted image. The generated image depicted a damaged phone, consistent with the prompt's content, highlighting the limitations of automatic recognition of microscopic image content.

As shown in Fig. 3c (using the example of the Stable Diffusion 3.5 Large Turbo model), the algorithm did not correctly recognize the image's content, failing to generate a result within the broadly understood domain of microscopic data. Nevertheless, it is possible to simultaneously analyze multiple data sources, which serve as a preliminary form of multimodality. In the image-to-image category, the algorithm processes both a textual prompt and a reference image, which theoretically should allow for more precise control over the generated output. However, in practice (e.g., using the tool stablediffusion.com, Fig. 4), the results obtained when providing both a textual prompt and an image still deviate from the source image, with a noticeable dominance of the textual prompt's influence. When inputting only a single image and a textual prompt suggesting a specific domain, the results remain unsatisfactory, as the generated images significantly differ from the reference image.

When an uninformative prompt is introduced, the generator produces images entirely dissimilar to the reference (Fig. 4c). The results generated using the Image-to-Image function largely depend on the balance between the input image and the textual prompt. In many cases, the textual prompt has a greater influence on the output than the actual image, leading to the generation of aesthetically pleasing images that often deviate from geological details and the original features of the input image. Adjusting generation parameters, such as the weight of the input image's influence and creativity level, can improve the consistency of the results with real data. Tools like OpenArt allow customization of these parameters, affecting the final output. An example is the Creativity Level parameter, adjustable from 0 to 1. A value of 0 indicates the lowest creativity level, where

**Table 1** Example of pseudocode for local image generation

**SD based**

```
1. IMPORT required libraries:
- os, PIL.Image, torch, transformers, diffusers,
accelerate, torchvision
2. DEFINE parameters:
- MODEL_ID = "runwayml/stable-diffusion-v1-5"
- NUM_EPOCHS = 10
- BATCH_SIZE = 4
- LEARNING_RATE = 5e-6
- IMAGE_SIZE = (512, 512)
- DEVICE = "GPU" if available else "CPU"
- NOISE_SCHEDULER = DPMScheduler
3. DEFINE CustomDataset(data_dir, image_size):
- CHECK if data_dir exists
- LOAD all images with extensions
- APPLY transformations:
  - Resize to IMAGE_SIZE
  - Convert to a tensor
  - Normalize to range [-1, 1]
- RETURN transformed image tensor
4. INITIALIZE:
- dataset ← CustomDataset(data_directory)
- dataloader ← DataLoader(dataset, BATCH_SIZE,
shuffle=True)
5. LOAD Stable Diffusion pipeline from MODEL_ID:
- pipeline←
  StableDiffusionPipeline.from_pretrained(MODEL_I
  D)
- Extract components: unet, vae, tokenizer,
  text_encoder
6. SET device for training:
- models to DEVICE
- optimizer ← AdamW(unet.parameters(),
  lr=LEARNING_RATE)
- (optional) Accelerator setup (cpu/gpu
  synchronization)
7. FOR each epoch in NUM_EPOCHS:
- SET model to training mode
- FOR each batch of images:
  - MOVE images to DEVICE
  - GENERATE textual prompts (e.g., geological
  descriptions)
  - TOKENIZE prompts with tokenizer
  - ENCODE text using text_encoder →
  text_embeddings
  - ENCODE images to latent space using vae →
  latents
  - SCALE latent values (typically ×0.18215)
  - SAMPLE random Gaussian noise, shape as
  latents
  - SAMPLE random timesteps in range [0, 1000]
  - ADD noise to latents using noiseScheduler →
  noisyLatents
  - PREDICT noise using unet
  - COMPUTE loss ← MSE(noise_pred, noise)
  - BACKPROPAGATE:
    - loss.backward()
    - optimizer.step()
    - optimizer.zero_grad()
    - PRINT progress every N batches / SAVE
  pipeline checkpoint to output_directory after
  each epoch
8. AFTER training:
  - LOAD trained pipeline from output_directory
  - DEFINE generation prompt (e.g., "dolomite
polarized")
  - GENERATE synthetic images
    - FOR i in number_of_images:
      - image ← pipeline(prompt).images
      - SAVE image
```

**(DC)GAN based**

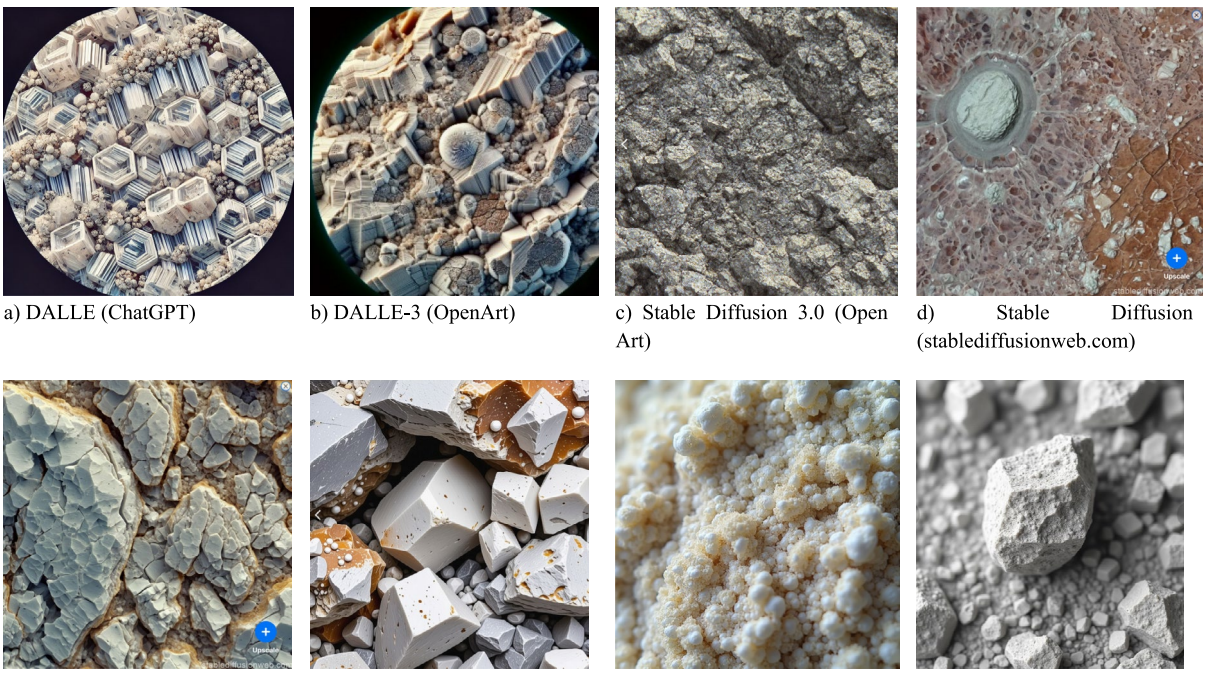
```
1. IMPORT required libraries:
- torch, torchvision, matplotlib, PIL, os, glob
2. SETUP & DEFINE parameters:
- DEVICE = "GPU" if available else "CPU"
- Define hyperparameters:
  - latent_dim = 100
  - batch_size = 32
  - epochs = 500
  - image_size = 64
3. INITIALIZE containers to track training loss:
  - loss_d_values, loss_g_values, epochs_list
4. DEFINE data transformation pipeline:
  - Resize images
  - Convert to a tensor
  - Normalize to range [-1, 1]
5. DEFINE CustomDataset:
  - LOAD all image files from the given folder
  - Apply transformations to each image
  - Return image tensor
6. INITIALIZE DataLoader:
  - Use CustomDataset with a defined transform
  - Shuffle data, use defined batch_size
7. DEFINE Generator model:
- GAN (7 layers, MLP): Sequence of Linear,
   BatchNorm1d, LeakyReLU, Tanh, OR
   - DC-GAN (12 or 15 layers): FCL, Sequence of
   Conv2d blocks with BatchNorm2d, ReLU, Tanh
   (output normalized RGB image)
8. DEFINE Discriminator model
   - GAN (6 layers, MLP): Linear, LeakyReLU,
   Sigmoid
- OR DC-GAN (ex.PatchGAN, 13 layers):
  - Sequence of Conv2d blocks with LeakyReLU,
   BatchNorm2d
  - Sigmoid
9. INITIALIZE models and optimizers:
  - Generator and Discriminator
  - Adam optimizer for both G and D
  - BCELoss for adversarial loss
10. DEFINE helper functions:
  - save_generated_images
  - plot_losses(): plot loss values over epochs
   (optional)
11. TRAINING LOOP (for each epoch):
    FOR each batch of real images:
    - Send real images to the device
    - CREATE label tensors:
      - valid = 0.9 (real), fake = 0.1 (fake)
      - shape matches Discriminator output
    # Generator:
    - z ← Sample random noise (batch_size,
   latent_dim)
    - Generate synthetic images from z
    - Compute generator loss:
    - g_loss = BCE(Discriminator(gen_images),
   valid)
    - Backpropagate and update Generator weights
    # Discriminator:
    - Compute real_loss:
        BCE(Discriminator(real_images), valid)
    - Compute fake_loss:
        BCE(Discriminator(detached_fake_images),
   fake)
    - Combine: d_loss = (real_loss + fake_loss)/2
    - Backpropagate and update Discriminator
   weights
    AFTER each epoch:
    - Save generated images every X epochs
    - Save loss plots every X epochs
```

**Table 2** Key Exemplary Training Parameters

| Method | Image resolution | Number of epochs | Batch size | Learning rate | Optimizer | Noise scheduler/Loss Function | Tokenizer & Prompt encoder/Label smoothing | Framework |
|---|---|---|---|---|---|---|---|---|
| SD based | 512 × 512 | 10 | 4 | $5 \times 10^{-6}$ | AdamW (Weight Decay fix) | DDPMScheduler (1000 steps, scaled_linear) | CLIPTokenizer & CLIP Text Encoder | PyTorch + Accelerate + Hugging Face Diffusers |
| GAN based | 64 × 64 | to 2500 | 32 | 0.0002 | Adam ($\beta_1$ = 0.5, $\beta_2$ = 0.999) | Binary Cross-Entropy Loss (BCE) | No tokenizer, without label smoothing: 1, 0 | PyTorch |
| DCGAN based | 256 × 256 | As above (GAN), with Generator and Discriminator architecture switching from fully connected (MLP) to deconvolutional with ConvTranspose2 d (1024 → 512), and label smoothing: 0.9 (real), 0.1 (fake) | | | | | | |

the generated image closely replicates the input image. In contrast, a value of 1 represents the highest level of creativity; the image becomes entirely dissimilar to the reference, regardless of the textual prompt (Fig. 5).

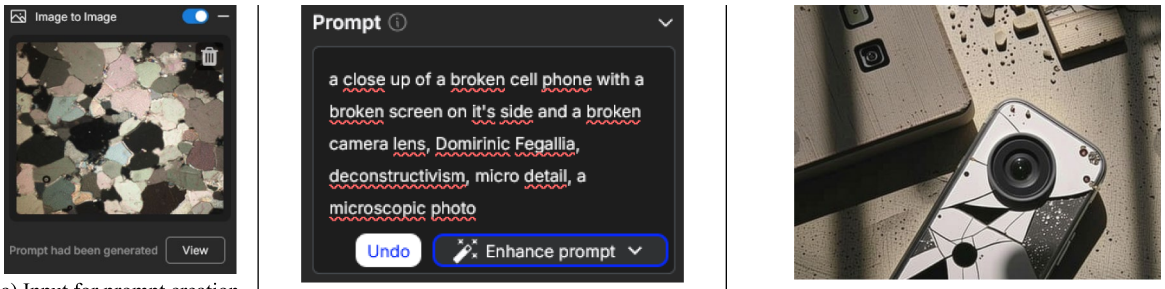
It is important to note that generating an image based on a single input image and a textual prompt (Fig. 5) essentially modifies the given image or creates a new image that visually deviates from the reference. As a result, this approach is unlikely to provide valuable data augmentation. Nevertheless, using textual prompts is a highly influential branch of generative artificial intelligence, gaining widespread popularity, particularly in everyday applications. In the authors'opinion, this area is not yet sufficiently reliable for direct application in expert research within geoengineering and Earth sciences without additional refinement. However, given the rapid advancement of these methods, it is expected that results will soon be updated and improved by both service providers and researchers.



a) DALLE (ChatGPT) b) DALLE-3 (OpenArt) c) Stable Diffusion 3.0 (Open Art) d) Stable Diffusion (stablediffusionweb.com)

e) Stable Diffusion (stablediffusionweb.com) f) Stable Diffusion 3.5 Large Turbo (OpenArt) g) Flux (dev) (OpenArt) h) Flux (realism) (OpenArt)

**Fig. 2** Examples of generated images of Dolomite using models without transfer learning – text prompt:"A microscopic image of Dolomite rock, magnification ×20"




a) Input for prompt creation

b) Prompt generated based on the image input (a)

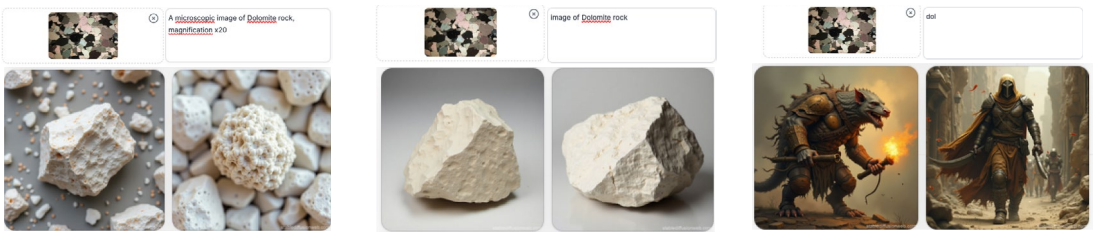
c) Image based on the text prompt (a → b)

**Fig. 3** An example of automatic prompt generation (image content recognition aspect), and the result of image generation is based on the automatically generated prompt (StableDiffusion 3.5 Large Turbo (OpenArt))

## UC2: generation of images based on training a model from scratch in a local environment (Image(s)-to-Image)

Generating images based on a custom training dataset (Fig. 1) using models trained from scratch is highly time-consuming. For example, generating images with a resolution of 64 × 64 pixels in a local environment (e.g., on a laptop) can take several to even dozens of hours, depending on the training dataset size (tens to hundreds of images) and the selected parameters. To achieve a high level of accuracy, models typically require a substantial amount of training data and significant computational power to support numerous training iterations, often spanning hundreds or thousands of epochs. Figure 6 presents the results of an example model training process (based on the Vanilla GAN architecture) using a dataset of 300 training images of Dolomite from Rędziny (Fig. 1). This dataset represents a small number of input samples. Achieving results like the reference images was possible, but required many computational epochs.

It can be observed that as the number of epochs increased, the content of the generated image became progressively clearer. A significant improvement occurred up to approximately the 200 th epoch; beyond this point, the model stabilized the balance between noise and the loss function of both the generator and the discriminator, without introducing substantial changes in output quality. The discriminator, a neural network consisting of several fully connected layers, transformed the image into a high-dimensional feature space in its first layer. The subsequent hidden layers gradually reduced this space, utilizing the Leaky ReLU activation function, which allowed for gradient propagation. The final layer, employing a sigmoid activation function, converted the output into a single value representing the probability of the image being authentic. The generator transformed a random input vector into an image through linear layers. Due to computational complexity, images were generated at a resolution of 64 × 64 pixels. Generating images at a higher resolution would significantly increase computation time (up to several dozen hours), requiring modifications to the network architecture. To generate higher-resolution images (256 × 256 pixels) with a more realistic appearance for this study, the architecture was modified to use a model incorporating convolutional layers, Deep Convolutional GAN (DCGAN). The generator created images based on a random vector, while the discriminator evaluated their authenticity. The generator utilized transposed convolutional layers (ConvTranspose2 d), whereas the discriminator employed classical convolutional layers (Conv2 d), supported by BatchNorm2 d layers and the Leaky ReLU activation function.




a) Example 1, prompt: *A microscopic image of Dolomite rock, magnification x20*

b) Example 2, prompt: *image of Dolomite rock*

c) Example 3 - non-informative text prompt: *dol*

**Fig. 4** Examples of textual prompts enriched with an image of a real rock and the results generated based on the **Image-to-Image** function (using the input image and textual prompt)

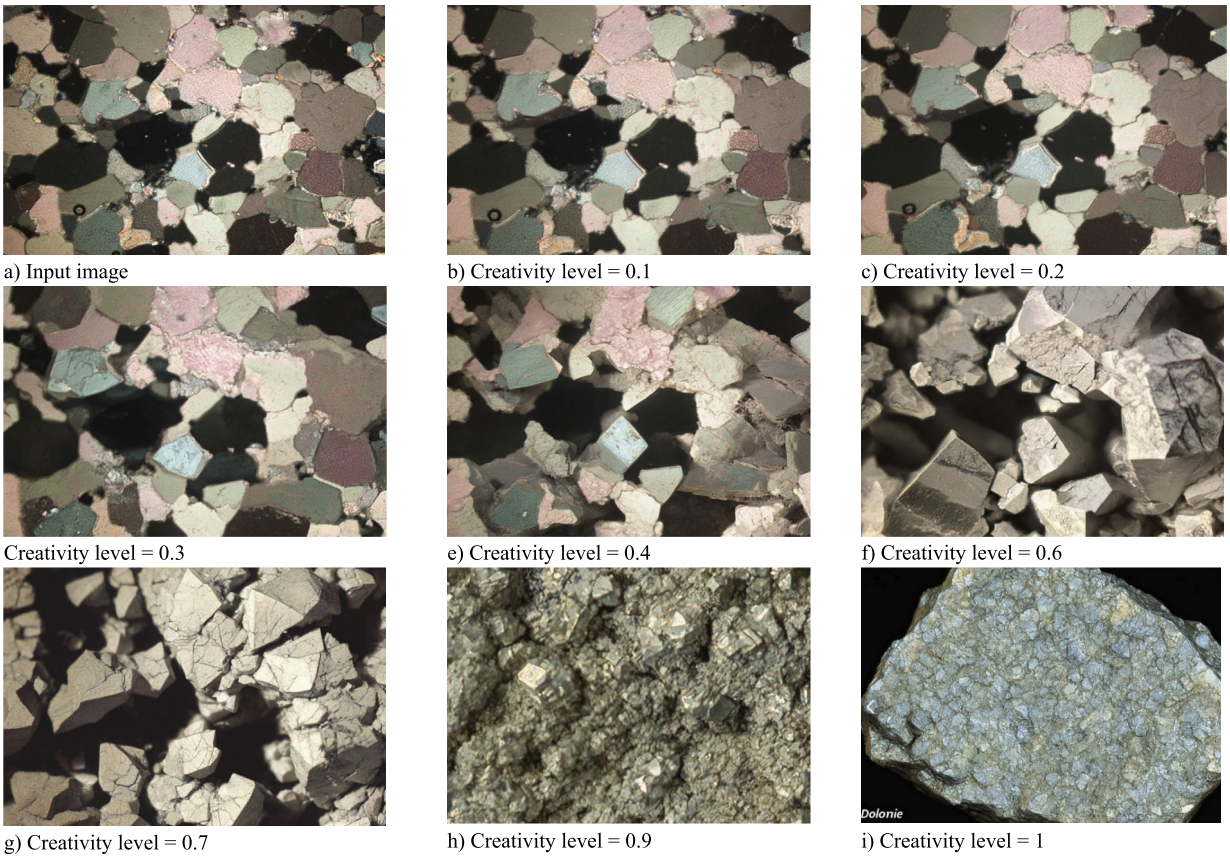



**Fig. 5** Examples of the Creativity Level parameter's influence on image generation results in the Text & Image-to-Image function (OpenArt)

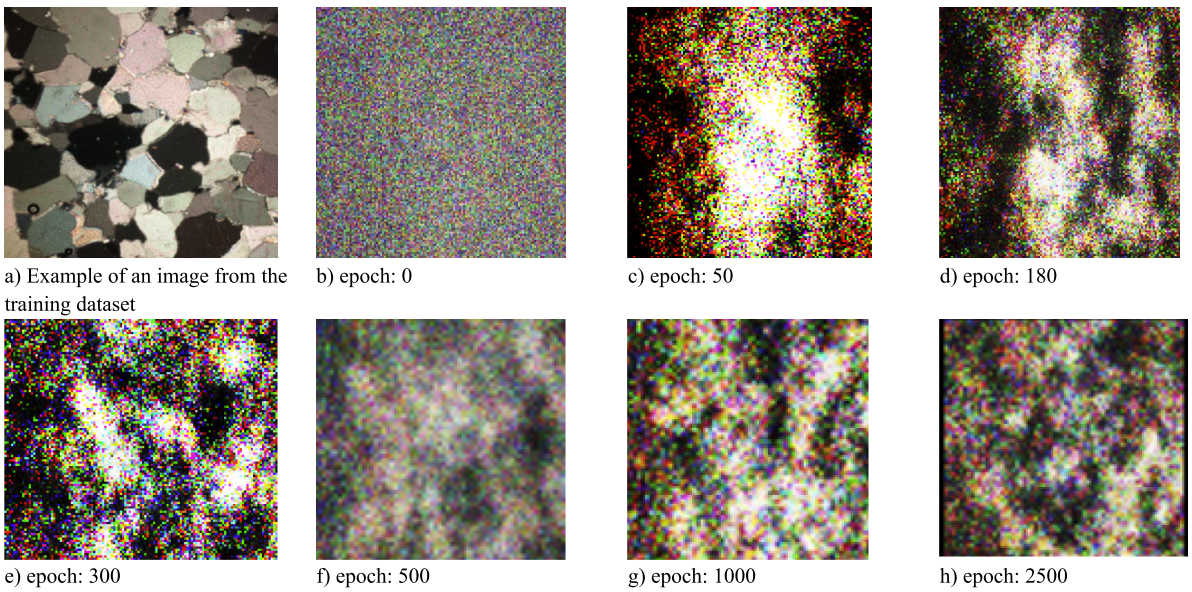



**Fig. 6** Examples of generated images using the **GAN approach**, expected output size **64 × 64 pixels** (local Python environment)

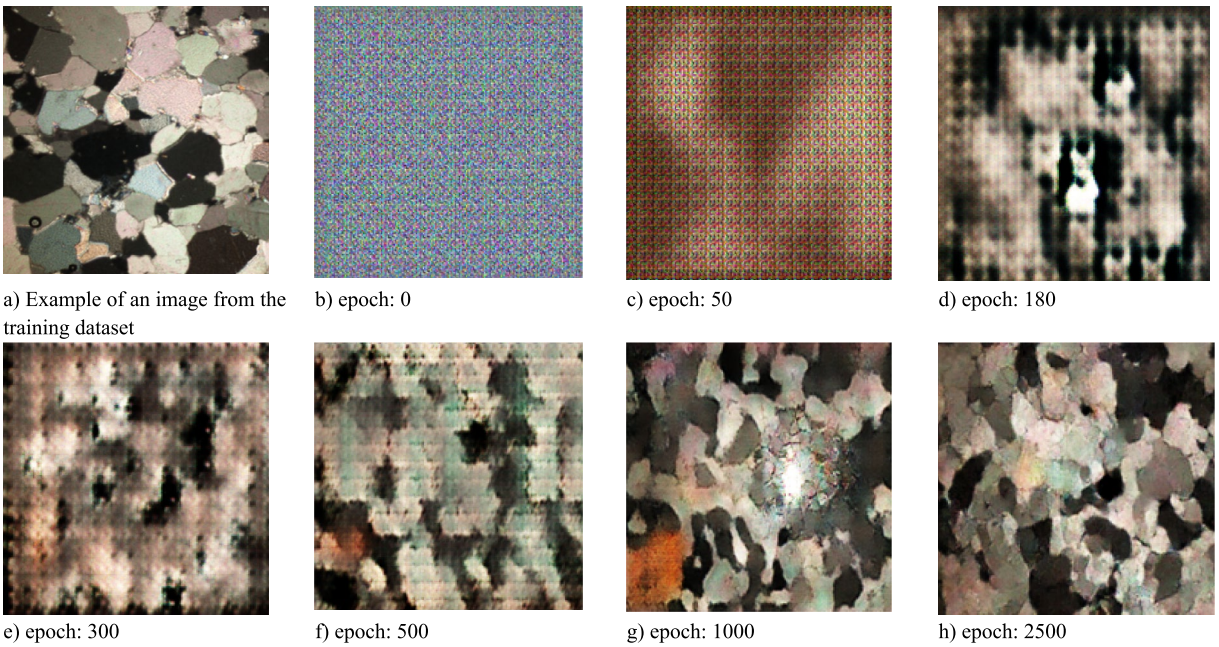
The generation results, depending on the number of epochs, are shown in Fig. 7. The obtained outputs demonstrated significantly better results, and despite the smaller output size compared to the input data (input: 1200 × 800 pixels, output: 256 × 256 pixels), they resembled the input structures.

In this case, positive results with high representativeness were achieved starting from approximately the 900 th computational epoch. The correctness of learning can be observed, apart from empirical evaluation, by analyzing the loss functions of the Discriminator and Generator. An example plot of the loss function values for the case from Fig. 7 is shown in Fig. 8a. The loss function values should be interpreted jointly for the discriminator and generator. Relatively good generation results are indicated when (in this case) both the generator and discriminator loss functions are in the range of 0.5–1.5. If Loss G increases while Loss D decreases, this can be interpreted as the generator producing more realistic images (e.g., epoch 1030). If Loss D increases, the discriminator may be strong, while the generator is too weak (e.g., epochs 630, 860). In the analyzed example, training begins to stabilize around epoch 900. However, even in this case, individual deviations may occur (Fig. 8b, c, d, e, f – an example of relatively incorrect generations, although still of very high quality).

Training each epoch for the sample architecture shown in Figs. 7, 8, and 9 (DC-GAN) took approximately 10 s, meaning that with 2500 epochs, the total training time was around 7 h (using a personal laptop). The results presented are for illustration purposes. For generating data with actual research value, further architectural configurations are possible, such as analyzing the impact of the loss function or increasing the dimensionality of the random vector used to initialize the model (e.g., to 256 or more). Such modifications could represent a distinct research topic; however, to maintain the coherence of this study, the authors concentrated on demonstrating the conceptual possibilities. It is important to emphasize that the obtained results indicate that, despite high computational requirements and the need for careful configuration, the proposed architecture can generate images that closely resemble the training dataset. Another key advantage is the ability to perform the entire process in a closed computational environment, without transferring data to external servers. This is a significant benefit in the context of geological and geotechnical research.

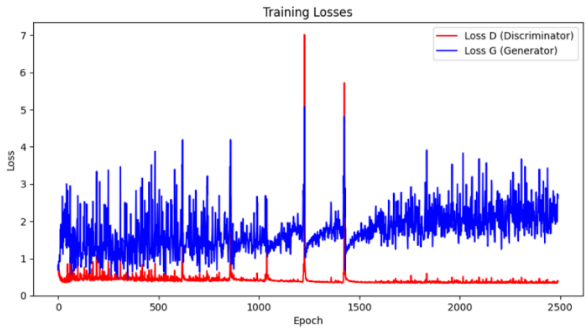
### UC 3: generation of images using transfer learning on pre-trained models, in local and remote environments (Text/Image-to-Image)

Generating images using transfer learning extends the case described in Sect."UC1. Generation of images based on textual prompts using pre-trained models without applying transfer learning (Text-to-Image)."(UC1). The key difference is the application of transfer learning in this scenario. The models used in these applications leverage both textual prompts and image datasets while being fine-tuned, allowing for the simultaneous use of textual prompts and images in fine-tuning the data. As part of the research presented in this study, two scenarios for the application of diffusion



a) Example of an image from the training dataset
b) epoch: 0
c) epoch: 50
d) epoch: 180
e) epoch: 300
f) epoch: 500
g) epoch: 1000
h) epoch: 2500

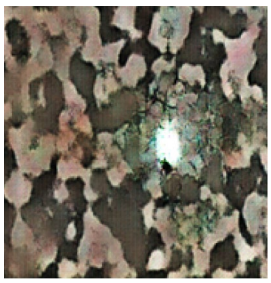
**Fig. 7** Examples of generated images using the **DC-GAN approach**, expected output size **256 × 256 pixels** (local Python environment)




a) Example of the loss function plot for the Generator and Discriminator in DC-GAN

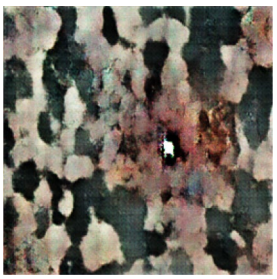
b) epoch: 630

c) epoch:860



d) epoch:1040

e) epoch: 1240



f) epoch: 1430

**Fig. 8** Example of the model loss function plot (a) and image generation results (b-f) showing deviations in the discriminator and generator loss values (examples of relatively incorrect generation)

a) epoch: 1780

b) epoch: 1890

c) epoch: 2350

d) epoch 2480

**Fig. 9** Selected examples of generated images using the DC-GAN approach, output size 256 ×256 pixels (local Python environment)

models were analyzed, depending on the transfer learning environment:

**UC 3.1 Transfer learning of the model in a local environment:** The model was trained in a local environment using Python. The process involved initializing the model by downloading a pre-trained model (e.g., Stable Diffusion from the Hugging Face platform) and then exemplarily adjusting its parameters for this example (configuring the number of epochs (10), mini-batch size (2)). The training stages included loading data, initializing the pre-trained Stable Diffusion model (stable-diffusion-v1-5) from the diffusers library, retrieving key model components, like U-Net (diffusion network), or text encoder based on CLIP, transferring the model to GPU (if available) and configuring the optimizer (AdamW), iterative data processing: processing textual prompts, adding noise and predicting its distribution by U-Net, minimizing the loss function (MSE) and updating model weights, saving the model after each epoch.

**UC 3.2 Use of models in a commercial environment:** This approach involved the application of commercial tools based on diffusion models, which have gained significant popularity within the content creation community. An example of such a solution is the OpenArt platform (or similar services), from which it can be inferred, based on marketing material, that it works on diffusion models. To evaluate the effectiveness of model training in a cloud-based environment, the authors submitted a set of 50 microscopic images of dolomites for external computations to enable fine-tuning of the model for specific geological data. The OpenArt platform operates on a subscription model, and

in the experiment, a package providing 12,000 credits was selected. The cost of training a single model was approximately 2000 to 3000 credits, while each image generation depended on the parameters (e.g., TurboOn mode) and ranged between 2 and 10 credits per generation.

Since Stable Diffusion models are characterized by multimodality, they can process both image and text data, and the generation process can be based on training images and textual prompts. If the prompt used for generation matched those used during training, the generated images were moderately similar to the input data (Fig. 10). The characteristics of the microscopic images were evident; however, they differed strongly in contrast and structure. However, if the prompt deviated from the trained data, the model produced images that were less like the training dataset (Fig. 11).

Interesting and surprisingly good results were obtained using commercial tools that utilize various models for UC 3.2. The results achieved with a defined configuration were sometimes very realistic and accurately reflected input data, demonstrating the potential of these algorithms. This tool allows parameterization of the influence of training data on the generation results (for commercial solution OpenArt from 0 to 1.5, where 0 indicates low influence and results in images dissimilar to the training data, while 1.5 means generating images very similar to the reference) Fig. 12.

It is also vital that the application generates higher-resolution images in a very short time (the maximum available resolution in the authors'subscription plan is $1536 \times 1536$ pixels), with surprisingly good results, which likely indicates highly optimized algorithms and powerful computational resources used for training the models. The images generated were manually compared with the data used for training, and no exact 1:1 copy was found. However, as seen in Fig. 13, these images are often well-reproduced variations, differing primarily in coloring or the shape of individual objects, indicating that the training dataset performs well.

## Discussion

Microscopic images of rock's thin sections are characterized by high detail, containing numerous details due to the presence of different materials and structures. Acquiring such data is a complex and costly process, and the data itself is often protected by copyright, making it difficult to share with public artificial intelligence models. Generating microscopic geological rock images on the local machines is possible but challenging, requiring appropriate architecture, test data, and

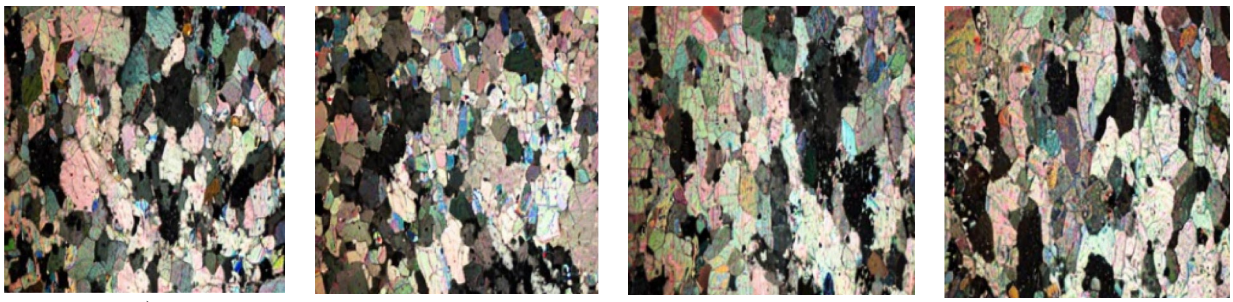



**Fig. 10** Images generated using the Stable Diffusion method, local Python environment, with a textual prompt matching the training data, result after 10 epochs (UC 3.1)

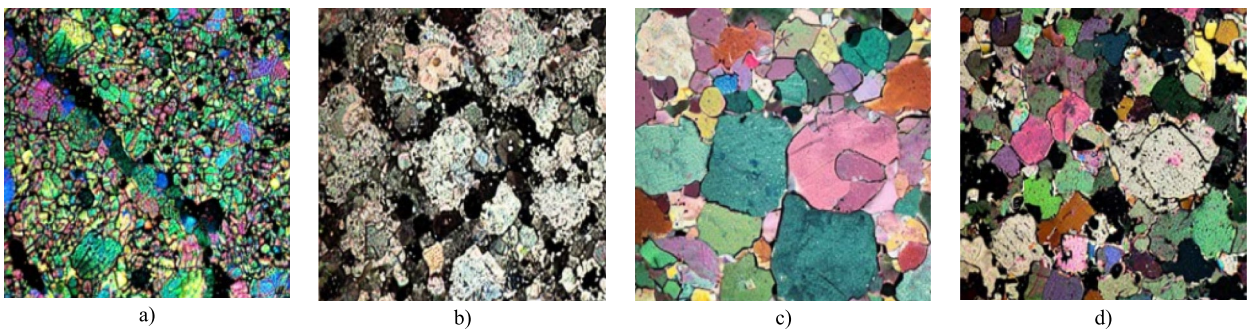
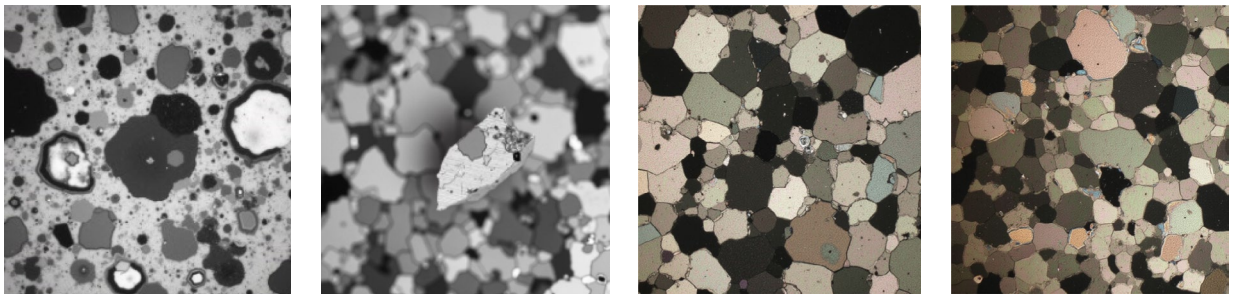



**Fig. 11** Images generated using the Stable Diffusion method, local Python environment, with a textual prompt not matching the training data, result after 10 epochs (UC 3.1)

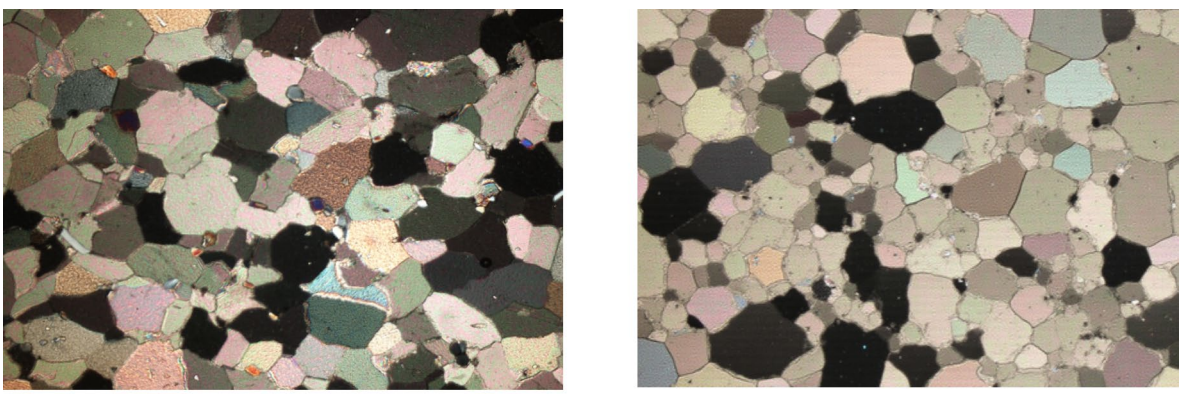


a) Model weight: 0 (min) b) Model weight: 0.4 c) Model weight: 0.8 d) Model weight: 1.5 (max)

**Fig. 12** Images generated using the Stable Diffusion method, commercial environment, with an empty textual prompt (white space), thus not matching the training data (OpenArt, UC 3.2)

computational resources. Significant differences in image quality were observed depending on the GAN-based architecture. Switching from MLP to Conv. layers, adding convolutional layers, and increasing the initial channel depth to 1024 (as in DCGAN) improved detail and realism, highlighting the effectiveness of deeper convolutional networks in modeling complex geological structures. The Stable Diffusion algorithm demonstrates potential in image generation, and its pre-trained versions enable further refinement of results through textual input prompts. Commercial models often result in high-quality images that are difficult for the human (non-expert) eye to distinguish from real photographs (Fig. 13). This highlights the potential of these technologies, when subjected to detailed analysis and expert-level training, to become a powerful tool for computational rock simulations. From the point of view of empirical evaluation, these methods seem to be effective. A popular similarity measure was also calculated between the example input image and the generated results (Table 3).

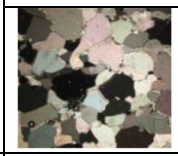
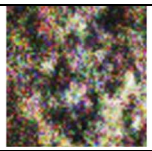
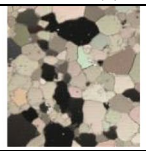
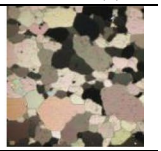
The SSIM (Structural Similarity Index) is dimensionless, where 0 indicates no similarity and 1 represents complete structural similarity. Conversely, the PSNR (Peak Signal-to-Noise Ratio), expressed in decibels (dB), reflects the level of mean squared error (MSE) between images. A greater PSNR value indicates better image quality. The results are relatively low in this case, as the generated images are inspired by the input images rather than being direct copies. However, the metrics can still be compared relative to one another across the generated images. Here, only minor differences are observed. Based on these observations, the DCGAN model achieves the best overall performance, exhibiting the highest SSIM and PSNR values. This suggests a stable balance between structural fidelity and pixel-level quality. In the case of diffusion models (Stable Diffusion), the visual results appear highly convincing; however, the metrics indicate a higher SSIM than DCGAN. This may suggest a better geometric representation of the image, albeit with a slightly lower PSNR, which reflects a higher average pixel error. This outcome aligns with the visual impression that Stable Diffusion images have a natural appearance yet may contain subtle intensity-level distortions. The locally trained Stable Diffusion model (trained on a small data sample and basic parameters) shows a low level of alignment, indicating weak adaptation to the training dataset. Nevertheless, the results achieved by strongly initiated and well-tuned Stable Diffusion models demonstrate the high potential of



a) b)

**Fig. 13** Difference between the sample real image (a) and the generated image (b, weight 0.9, OpenArt, UC 3.2)

**Table 3** Quantitative metric between the real image and the generated image

| | Real | GAN | SD-basic, local | DC-GAN | SD, OA (1) | SD, OA (2) |
|---|---|---|---|---|---|---|
| |  |  |  |  |  |  |
| SSIM | 1 | 0.13537 | 0.12577 | 0.25545 | 0.26513 | 0.23128 |
| PSNR (dB) | ∞ | 9.7501 | 8.8837 | 9.8976 | 8.6618 | 8.4729 |

this technology for practical applications. However, the high PSNR value observed for the GAN model requires cautious interpretation. To calculate this metric, the reference image was downscaled to match the resolution of the generated image ($64 \times 64$), which, in the authors' opinion, does not fully reflect the actual similarity in this case.

Synthetic geological images demonstrate significant potential in analytical and computational applications, particularly in integrating machine learning models and geological data analysis systems. However, in the authors'opinion, using textual prompts in Stable Diffusion without a properly described training dataset requires further refinement for advanced geological research. For instance, the generated images did not reflect the expected effect despite explicitly specifying the exclusion of light polarization in the microscope within the prompts. This is understandable, as preparing a properly curated dataset for transfer learning would be necessary. The best results were achieved using commercial tools, likely due to algorithmic optimization and significant computational resources available in these platforms. However, it is also possible to generate images locally, especially when many images need to be created, and the data is confidential.

A significant advantage is the ability to supplement existing datasets, particularly in cases where acquiring real-world images is costly, complex, or even impossible. Therefore, in the authors' opinion, one of the key motivations behind employing generative models in geological contexts is the ability to augment limited datasets and simulate rare or underrepresented geological patterns. However, synthetic geological images can be integrated into machine learning pipelines in several ways. For example, in mineral deposit location prediction tasks, images generated using GAN networks can be incorporated into data augmentation processes as part of an automated pipeline (e.g., using the ImageDataGenerator in the Keras environment). This may lead to increased diversity in training data and improve the overall generalization capability of the models. Synthetic image data can also be used to evaluate the performance of advanced segmentation models such as U-Net, Mask R-CNN, or Detectron2 in tasks related to identifying tectonic structures, faults, and stratigraphic formations. They can serve as a controlled reference dataset for benchmarking model performance, supporting fine-tuning by systematically monitoring performance metrics (e.g., precision, recall, F1-score). In training machine learning models, synthetic images can be used as input data for tasks such as lithological classification, identification of geological structures, and automated interpretation of seismic sections. For instance, future studies could explore integrating geoscientific images with convolutional models like ResNet-50 or EfficientNet, utilizing specific datasets within popular frameworks such as TensorFlow or PyTorch. Synthetic images may also be used in geological education and training. For example, they can be integrated into interactive simulations or used as teaching tools within platforms like Jupyter Notebook, employing libraries such as Gradio or Streamlit to enhance and popularize the field of microscopic imagery.

## Summary

This paper provides an overview and results of example methods for generating images as tools to support geological research. The results indicate that generating microscopic geological images of rocks is feasible; however, it requires careful selection of models and proper configuration of their architecture and parameters, which significantly impact the outcomes. This process is particularly challenging due to the high level of detail, varied texture, and complex structure of geological images, posing additional challenges in parameter optimization. On the other hand, while using textual prompts in pre-trained models requires further refinement, results obtained with advanced (and often commercial) models suggest that the generated images are of high quality and are frequently difficult to distinguish from real ones. This indicates the significant potential of this technology and the need for further research in this direction. The authors agree with the conclusions of Zhang et al. (2022), stating that classification and assessment of vulnerability to geological hazards play a crucial role in geosciences. Due to geomaterials'spatial variability and anisotropic properties, results obtained using machine learning (ML) or deep learning (DL) methods may carry significant uncertainty. Therefore, despite promising results, the authors, citing Zhang et al. (2022), emphasize the need to combine statistical methods (ML) with approaches based on physical

data and expert knowledge, enabling the proper development of advanced technologies. In the authors'opinion, artificial intelligence can bring significant innovations to geological and geotechnical engineering, but its development should not replace human labor and expertise. Technological progress in this field opens new research and professional opportunities, increasing opportunities among engineers in developing and improving intelligent geological analysis systems.

**Acknowledgements** This work was supported by the AGH University of Krakow, Faculty of Geology, Geophysics and Environmental Protection.

**Author contributions** All authors contributed to the work presented in this manuscript. Author 1 (Młynarczuk Mariusz), and Author 2 (Habrat Magdalena) were involved in the paper preparation. The specific contributions can be detailed as follows: conceptualization: MM and HM; methodology and validation: MM and HM; formal analysis: MM and HM; investigation: MM and HM; data curation: MM and HM; original draft preparation: HM; review and editing: MM, and HM; supervision: MM. All the authors have read and agreed to the published version of the manuscript.

**Funding** This research received no external funding.

**Data availability** No datasets were generated or analysed during the current study.

## Declarations

**Competing interests** The authors declare no competing interests.

**Clinical trial number** Not applicable.



## References

Izadi H, Sadri J, Bayati M (2017) An intelligent system for mineral identification in thin sections based on a cascade approach. Comput Geosci 99:37–49

Marmo R, Amodio S, Tagliaferri R, Ferreri V, Longo G (2005) Textural identification of carbonate rocks by image processing and neural network: Methodology proposal and examples. Comput Geosci 31(5):649–659

Li N, Hao H, Gu Q, Wang D, Hu X (2017) A transfer learning method for automatic identification of sandstone microscopic images. Comput Geosci 103:111–121

Singh N, Singh TN, Tiwary A, Sarkar KM (2010) Textural identification of basaltic rock mass using image processing and neural network. Comput Geosci 14:301–310

Gerard RE, Philipson CA, Manni FM, Marschall DM (1992) Petrographic image analysis: an alternate method for determining petrophysical properties. Automated pattern analysis in petroleum exploration. New York, NY, Springer, New York, pp 249–263

Młynarczuk M (2010) Description and classification of rock surfaces by means of laser profilometry and mathematical morphology. Int J Rock Mech Min Sci 47(1):138–149

Ładniak M, Młynarczuk M (2015) Search of visually similar microscopic rock images. Comput Geosci 19:127–136

Habrat M, Młynarczuk M (2018) Evaluation of local matching methods in image analysis for mineral grain tracking in microscope images of rock sections. Minerals 8(5):182

Long T, Zhou Z, Hancke G, Bai Y, Gao Q (2022) A review of artificial intelligence technologies in mineral identification: classification and visualization. J Sens Actuator Netw 11(3):50

National Academies of Sciences Engineering and Medicine (2020) Characterization, modeling, monitoring, and remediation of fractured rock. Washington, DC: The National Academies Press. https://doi.org/10.17226/21742

Izadi H, Sadri J, Mehran NA (2013) A new approach to apply texture features in minerals identification in petrographic thin sections using ANNs. In 2013 8th Iranian conference on machine vision and image processing (MVIP) (pp. 257-261). IEEE, Zanjan, Iran, 10-12 September 2013. https://doi.org/10.1109/IranianMVIP.2013.6779990

Kaswan KS, Dhatterwal JS, Malik K, Baliyan A (2023) Generative AI: a review on models and applications. In 2023 international conference on communication, security and artificial intelligence (ICCSAI) (pp. 699–704). IEEE, Greater Noida, India, 23-25 November 2023. https://doi.org/10.1109/ICCSAI59793.2023.10421601

Bengesi S, El-Sayed H, Sarker MK, Houkpati Y, Irungu J, Oladunni T (2024) Advancements in generative AI: a comprehensive review of GANs, GPT, autoencoders, diffusion model, and transformers. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3397775

Sengar SS, Hasan AB, Kumar S, Carroll F (2024) Generative artificial intelligence: a systematic review and applications. Multimedia Tools and Applications 1–40. https://doi.org/10.1007/s11042-024-20016-1

Hadid A, Chakraborty T, Busby D (2024) When geoscience meets generative AI and large language models: Foundations, trends, and future challenges. Expert Systems 41(10):e13654

Abdellatif A, Elsheikh AH, Busby D, Berthet P (2022) Generation of non-stationary stochastic fields using generative adversarial networks. Preprint at https://arxiv.org/abs/2205.05469

Qaderi, S., Maghsoudi, A., Pour, A.B., Rajabi, A. & Yousefi, M. (2025). DCGAN-Based Feature Augmentation: A Novel Approach for Efficient Mineralization Prediction Through Data Generation. *Minerals*. 2025; 15(1):71.

Zhang W, Gu X, Tang L, Yin Y, Liu D, Zhang Y (2022) Application of machine learning, deep learning and optimization algorithms in geoengineering and geoscience: comprehensive review and future challenge. Gondwana Res 109:1–17

Pierdicca R, Paolanti M (2022) GeoAI: a review of artificial intelligence approaches for the interpretation of complex geomatics data. Geosci Instrumentation Methods Data Syst Discuss 2022:1–35

Ferreira I, Ochoa L, Koeshidayatullah A (2022) On the generation of realistic synthetic petrographic datasets using a style-based GAN. Sci Rep 12(1):12845

Nathanail A (2023) Geo Fossils-I: a synthetic dataset of 2D fossil images for computer vision applications on geology. Data Brief 48:109188

Saif A, Alnagi E, Ahmad A (2025) Texture-based classification of geo-fossils. In: Delir Haghighi P, Greguš M, Kotsis G, Khalil I (eds) Information Integration and Web Intelligence. iiWAS 2024. Lecture Notes in Computer Science, vol 15343. Springer, Cham. https://doi.org/10.1007/978-3-031-78093-6_20

Kupferschmidt C, Binns AD, Kupferschmidt KL, Taylor GW (2024) Stable rivers: a case study in the application of text-to-image generative models for Earth sciences. Earth Surf Proc Land 49(13):4213–4232

Khanifar J (2025) Evaluating AI-generated responses from different chatbots to soil science-related questions. Soil Adv 100034. https://doi.org/10.1016/j.soilad.2025.100034

Hejducki Z, Podolski M (2012) Harmonogramowanie przedsięwzięć budowlanych z zastosowaniem algorytmów metaheurystycznych. Zeszyty Naukowe WSOWL, Nr 4(166):68–79

Piotrowski AP, Napiorkowski JJ, Kiczko A (2012) Differential evolution algorithm with separated groups for multi-dimensional optimization problems. Eur J Oper Res 216:33–46

Huang, Y. (2020). Deep Q-networks. *Deep reinforcement learning: fundamentals, research and applications*, 135–160.

Bashar A (2019) Survey on evolving deep learning neural network architectures. J Artif Intell 1(02):73–82

He KX, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp 770–778. https://doi.org/10.1109/CVPR.2016.90

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25(2):1097–1105. https://doi.org/10.1145/3065386

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. Preprint at https://arxiv.org/abs/1406.1078

Radford A (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint at https://arxiv.org/abs/1511.06434

Lv Z (2023) Generative artificial intelligence in the metaverse era. Cogn Robot 3:208–217

Uppalapati VK, Nag DS (2024) A comparative analysis of AI models in complex medical decision-making scenarios: evaluating ChatGPT, Claude AI, bard, and perplexity. Cureus 16(1). https://doi.org/10.7759/cureus.52485

Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2023) Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comput Surv 55(9):1–35

Durgadevi M (2021) Generative adversarial network (GAN): a general review on different variants of GAN and applications. In 2021 6th international conference on communication and electronics systems (ICCES) (pp. 1–8). IEEE, Coimbatre, India, 08-10 July 2021. https://doi.org/10.1109/ICCES51350.2021.9489160

Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Adv Neural Inf Process Syst 33:6840–6851

Lee S, Hoover B, Strobelt H, Wang ZJ, Peng S, Wright A, ... Chau DHP (2024) Diffusion explainer: visual explanation for text-to-image stable diffusion. In 2024 IEEE visualization and visual analytics (VIS) (pp. 96–100). IEEE, St. Pete Beach, FL, USA, 13-18 October 2024. https://doi.org/10.1109/VIS55277.2024.00027

Chen M, Mei S, Fan J, Wang M (2024) An overview of diffusion models: applications, guided generation, statistical rates and optimization. Preprint at https://arxiv.org/abs/2404.07771

Yang L, Zhang Z, Song Y, Hong S, Xu R, Zhao Y, ... Yang MH (2023) Diffusion models: a comprehensive survey of methods and applications. ACM Comp Surv 56(4), 1–39

Gallon D, Jentzen A, von Wurstemberger P (2024) An overview of diffusion models for generative artificial intelligence. Preprint at https://arxiv.org/abs/2412.01371

Stable Diffusion (n.d.) https://stablediffusionweb.com/. Accessed 18 Jan 2025

Stable Diffusion API (n.d.) https://stablediffusionapi.com/. Accessed 18 Jan 2025

OpenArt Advanced (n.d.) https://openart.ai/. Accessed 18 Jan 2025