МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп'ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Віктор СИНЄГЛАЗОВ
"___"_____2025 р.

КВАЛІФІКАЦІЙНА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

"БАКАЛАВР"

Спеціальність 151 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Освітньо-професійна програма «Інформаційні технології та інженерія авіаційних комп'ютерних систем»

Тема: Інтелектуальна система для генерації віртуальних навчальних зразків із збільшеною варіацією

Виконавець: здобувач вищої освіти Дмитренко Олександр Юрійович
Керівник: професор, доктор технічних наук Синєглазов Віктор михайлович

Нормоконтролер: _____ Філяшкін М.К.
(підпис)

Київ – 2025

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

STATE UNIVERSITY "KYIV AVIATION INSTITUTE"

Faculty of Aeronautics, Electronics and Telecommunications

Department of aviation computer-integrated systems

TO ALLOW FOR DEFENSE

Head of the graduate department

_____ Viktor SYNEGLAZOV

"___"_____2025 p.

QUALIFICATION WORK

(EXPLANATORY NOTE)

GRADUATE OF AN EDUCATIONAL DEGREE

"BACHELOR"

Specialty 151 "Automation, computer-integrated technologies and robotics"

Educational and professional program "Information Technology and Engineering of Aviation Computer Systems"

Title: Intelligent system for generating virtual training samples with increased variation

Performer: higher education applicant Dmytrenko Oleksandr Yuriyovych

Head: Professor, Doctor of Technical Sciences Viktor Mikhailovich Sinehlazov

Normative Controller: _____ Filiashkin M.K.

(signature)

Kyiv - 2025

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

Faculty of Aeronautics, Electronics and Telecommunications

Department of Aviation Computer Integrated Systems

Educational degree graduate: bachelor

Specialty 151 «Automation, computer-integrated technologies»

Educational and professional program «Information Technology and Aviation Computer Systems Engineering»

ADMISSION TO PROTECTION

Head of the Graduate Department

Viktor SYNEGLAZOV

"__ "_____ 2025 y.

TASK

for the completion of the student's qualifying work

Dmytrenko Oleksandr Yuriyovych

1. **Topic of work:** «Intelligent system for generating virtual training samples with increased variation»

2. Deadline for completion of work: from 22.05.2025 y. for 14.06.2025 y.

3. Initial data for work: -

4. Contents of the explanatory note (list of issues to be developed): -

5. List of mandatory graphic material: -

6. Calendar plan-graphic:

op.	Завдання	Термін виконання	Відмітка про виконання
1	Develop a detailed outline of the sections of the qualification work	19.05.2025-20.05.2025	Done

2	Introduction	21.05.2025	Done
3	Overview of problems in obtaining satellite data and analysis of generative network architectures	22.05.2025-25.05.2025	Done
4	Analysis and design of the generating system architecture	26.05.2025-31.05.2025	Done
5	Implementation of the generative model and its training	01.06.2025-15.06.2025	Done
6	Analysis of the results	01.06.2025-15.06.2025	Done
7	Elimination of shortcomings and defense of qualification work	16.06.2025-22.06.2025	Done

7. Date of assignment: "05" May 2025y.

Head of qualification work:_____ Viktor SYNEGLAZOV

The task accepted for execution by: _____ Oleksandr DMYTRENKO

___ "___2025y.

ABSTRACT

Qualification work «Intelligent system for generating virtual training samples with increased variation» contains 60 pages, 8 figures, 3 tables, 23 sources used.

Keywords: GENERATIVE MODELS, SATELLITE IMAGES, ARTIFICIAL INTELLIGENCE, GAN, SAGAN, CONDITIONAL GENERATION, VARIABILITY.

Object of research – the process of image generation using deep generative artificial intelligence models.

Subject of research – a method of conditional generation of satellite images using the Self-Attention GAN (SAGAN) architecture under conditions of limited training data.

The aim of the thesis is to develop and implement an image generation model capable of creating visually realistic satellite images of four classes of objects (fighters, bombers, helicopters, and airbases) with class control.

Research method: architectural modeling using Self-Attention GAN, formation of a conditional latent space, training using Hinge Loss and FID and LPIPS metrics.

The materials of the qualification work are recommended for use in research in the field of satellite image generation, supplementing training samples, and improving recognition models in conditions of limited data volume.

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

GAN (Generative Adversarial Network) — a generative adversarial neural network.

SAGAN (Self-Attention GAN) — a generative adversarial network with a self-attention mechanism.

cGAN (Conditional GAN) — a conditional generative adversarial neural network.

DCGAN (Deep Convolutional GAN) — a deep convolutional generative adversarial network.

FID (Fréchet Inception Distance) — a metric for evaluating the quality of image generation based on statistical characteristics.

LPIPS (Learned Perceptual Image Patch Similarity) — learned perceptual similarity of image patches.

IS (Inception Score) — a metric for image quality based on the Inception neural network.

SSIM (Structural Similarity Index) — an index of structural similarity between images.

WGAN (Wasserstein GAN) — a variant of GAN with a Wasserstein loss function.

WGAN-GP (Wasserstein GAN with Gradient Penalty) — WGAN with a gradient penalty to stabilize training.

BCE (Binary Cross Entropy) — binary cross entropy (loss function).

ReLU (Rectified Linear Unit) — activation function with zero output for negative values.

GP (Gradient Penalty) — gradient penalty applied in the loss function.

CNN (Convolutional Neural Network) — convolutional neural network.

ML (Machine Learning) — machine learning.

DNN (Deep Neural Network) — deep neural network.

L1, L2 — distance metrics: Manhattan (L1) and Euclidean (L2).

INTRODUCTION

In today's environment, satellite imagery is an important source of information for military analysis, strategic monitoring and intelligence, but the process of obtaining large amounts of high-quality satellite data is often complicated both technically and administratively, which hinders the development of high-precision computer vision models focused on analyzing such images. This problem becomes especially critical in the context of deep neural networks, which require a lot of data to ensure sufficient generalizability of results.

One of the promising ways to solve this problem is to use generative models capable of creating synthetic training examples based on limited real-world data. The class of models called Generative Adversarial Networks (GANs), which have proven to be effective in various fields of computer vision, in particular, in synthesizing images with a high degree of photorealism, attracts special attention of researchers [1]. At the same time, traditional GAN variants do not always demonstrate stable generation, especially when the number of training samples is limited. In this regard, architectures that integrate self-attention mechanisms, such as the Self-Attention GAN (SAGAN) model, which allows preserving global spatial dependencies and improves the quality of synthesized images, are gaining more and more attention [2].

The purpose of this study is to create an intelligent system capable of generating realistic satellite images of objects of four key classes - fighters, bombers, helicopters and air bases - using the SAGAN architecture. To achieve this goal, a number of tasks were set: to review existing scientific approaches to satellite data generation, collect and prepare our own annotated dataset, implement a generative model with class convention, conduct experimental training, evaluate the quality of

results by FID and LPIPS metrics, as well as to compare them with the results obtained in previous studies.

The object of research is the process of satellite image generation, the subject is the methods of conditional generation by classes using Self-Attention GAN. The novelty of the work lies in the adaptation of the self-attention mechanism to the tasks of generating specific types of images with a limited amount of training data. The practical value lies in the possibility of using the developed system to generate additional samples in object recognition tasks, which significantly increases the efficiency of classification and segmentation models in satellite data.

SECTION 1

OVERVIEW OF PROBLEMS IN OBTAINING SATELLITE DATA AND ANALYSIS OF GENERATIVE NETWORK ARCHITECTURES

1.1. The importance of satellite imagery in military strategic analysis

Satellite imagery plays a key role in modern military strategic analysis, as it provides objective, regular and spatially detailed information on the state of territories, infrastructure and the movement of military equipment. Thanks to the high resolution of the images obtained from surveillance satellites, it is possible to identify important objects such as air bases, fighters, helicopters and bombers, elements that are strategically important in the context of military conflicts.

Within the framework of geographic information analysis, satellite data is used to:

- prompt detection of changes in the location of military facilities;
- determining the level of combat readiness of units;
- monitoring compliance with the terms of international disarmament agreements;
- analyzing the extent of infrastructure damage after military operations or air strikes.

Both government analytical agencies and commercial companies pay special attention to satellite data. In particular, Maxar Technologies is one of the leaders in the field of satellite analytics, providing access to archives of high-precision images that are actively used in journalism, security research, and combat monitoring [3].

The development of automated image analysis methods based on deep learning has made it possible to detect objects without the need for full human involvement. This has paved the way for the widespread use of neural networks in military research, but the limited number of available annotated images creates serious obstacles to training such models. That is why the development of a system capable of generating satellite images for specified classes of objects is of particular relevance both in research and application.

1.2.Problems of access to up-to-date and annotated data

One of the most serious problems in creating satellite image generation models is the limited access to high-quality, up-to-date and annotated data, especially in the military domain. Most of the large public satellite datasets, such as xView, DOTA, AID, SpaceNet, focus mainly on civilian infrastructure - roads, buildings, fields, transportation - and only occasionally include military objects, and then without qualitative content verification.

In the practice of working with the xView dataset, which is positioned as one of the largest open satellite data sets, a number of serious shortcomings were identified. In particular, in many cases, the annotations were incorrect - for example, the "tanks" class contained images of civilian objects, and the "artillery" class contained images that did not correspond to the declared category at all. Such a situation makes it impossible to use such data without prior laborious manual verification and correction, which contradicts the goal of automating the process [4].

Another obstacle is the high cost of commercial satellite imagery containing up-to-date information on military infrastructure. Their price is usually unacceptable for research or student projects, which limits their use in academic settings.

A separate problem is the lack of a single data format: images of different resolutions, different spectrum channels, and lack of coordinates or metadata also make unified processing difficult. This increases the need for flexible generation architectures that can function even with a limited number of training examples.

1.3. Justification for creating own dataset

Given the limited access to high-quality and fully annotated military satellite imagery, there is an objective need to create our own dataset for generative modeling research. Existing public datasets, such as xView, DOTA, and others, not only have a limited representation of military objects, but also demonstrate poor quality annotations and a significant variation in formats. As a result, much of the material has to be excluded or edited manually, making it impossible to effectively generate automated images based on them.

In addition, the lack of guaranteed relevance of the images and the mixture of classes creates additional difficulties when building a conditional generator. For example, when

trying to use xView, it was discovered that some categories, such as "tanks" or "artillery," do not actually contain relevant objects. This forced me to abandon the idea of using existing sources in favor of creating dataset my own dataset which provides full control over the composition of classes, the quality of annotations, and the data structure.

For the purposes of this work, we created our own limited but focused dataset consisting of images of four key categories of military infrastructure: fighters, bombers, helicopters, and air bases. Each category is represented by approximately the same number of images - approximately 100 examples each - which allows for generation with a sufficient level of generalization with limited computing resources. This also makes it possible to evaluate the performance of the generative model in a small but high-quality dataset that meets the realistic limitations of practical projects.

1.4. Overview of GAN architectures

Generative Adversarial Networks (GANs) have become a key tool for synthesizing images and other types of data. The main idea of GANs is the confrontation between two neural networks - a generator and a discriminator - that compete with each other: the generator tries to create realistic images, while the discriminator tries to distinguish real images from generated ones. This interaction is defined as a minimax payoff function [1]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{real}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))] \quad (1.1)$$

$x \sim p_{real}$is a sample from a real dataset;

$z \sim p_{(z)\ is}$a latent vector from the standard distribution;

G(z) is an image created by the generator based on the latent vector z;

D(x) - the discriminator returns the probability that the image is real;

log D(x) is the logarithm of the discriminator score for the present example;

log(1 - D(G(z))) is the logarithm of the score for the generated example.

Since the basic GAN architecture was introduced in 2014 by Ian Goodfellow [1], several variations have been developed that have significantly improved the quality of image synthesis, training stability, and ability to work with conditional input data.

DCGAN (Deep Convolutional GAN) is one of the first successful architectures that combines GANs with deep convolutional networks [5]. It significantly improved the stability of training by using convolutional layers and special normalization principles.

cGAN (Conditional GAN) is a model that introduces an additional condition (for example, a class label) to the generator and discriminator, allowing you to control the type of generated image. This approach made it possible to generate images corresponding to specific categories [6].

StyleGAN and StyleGAN2 are architectures that have significantly improved the photorealism of generated images. The main innovation lies in the use of style space and modulation mechanisms that allow controlling certain aspects of the image, such as pose, texture, or shape [7][8].

Self-Attention GAN (SAGAN) is a model that integrates the self-attention mechanism into the generator and discriminator. This allows to effectively capture long-term dependencies in images and generate objects with complex spatial structure [2]. This property makes SAGAN suitable for generating satellite images with military objects that may be in different parts of the scene.

Thus, each subsequent architecture solves the problems inherent in the previous ones, gradually improving the quality of generation, training stability, and control over the output images.

Table 1.1

Comparison of Generative Competitive Networks

Architecture	Manageability	Stability of training	Consideration of the global context	Computational complexity	Suitability for satellite data
DCGAN	No	Low (collapse mode)	Local	Low	Limited
cGAN	Yes (conditional variable)	Medium	Local	Low-Medium	Good for basic control
WGAN	No	High	Local	Medium	Suitable, but without control
WGAN-GP	No	Very high	Local	High	Good stability, but no control
SAGAN	No (in the basic version)	High	Yes (self-attention)	High	Excellent quality and context
StyleGAN2	Limited (due to style)	High	Yes (due to stylization)	Very high	Excessive for the task

1.5. Classification of GAN architectures

Since the publication of the basic model of Generative Adversarial Networks (GAN) proposed by Ian Goodfellow in 2014 [1], a number of improvements have been proposed in the field of generative modeling aimed at increasing the stability of training, improving the quality of synthesized images, and expanding the possibilities of control over the generation process. Among the most famous architectures that have played a key role in the development of generative learning technologies are DCGAN, Conditional GAN (cGAN), StyleGAN2, and Self-Attention GAN (SAGAN). Let us consider each of them in the context of their role in the development of GAN approaches.

Deep Convolutional GAN (DCGAN)

The DCGAN architecture, proposed by Radford, Metzu, and Chun in 2015 [5], was the first large-scale practical step toward combining deep learning principles with generative

modeling. Unlike classical GANs, where the generator and discriminator used mostly fully connected layers, DCGANs introduced convolutional layers as the basis of both components of the architecture. This ensured localization of features and efficient capture of spatial dependencies in images.

Among the technical features of DCGAN are the following:

- use Batch Normalization to stabilize gradients;
- using LeakyReLU as an activation function in the discriminator and ReLU in the generator;
- use of transposed convolutions for upsampling;
- refusal to use pooling and fully connected layers.

DCGAN became the basis for numerous subsequent modifications. At the same time, this architecture has certain limitations, such as a tendency to mode collapse when hyperparameters are poorly tuned, limited generation control capabilities, and sensitivity to initial training conditions.

Conditional GAN (cGAN)

A conditional generative adversarial network (Conditional GAN) is a modification of the classical GAN architecture, in which both the generator and the discriminator receive additional input in the form of a conditional vector y. In most cases, such a vector is a class label that allows you to control the process of generating images in accordance with a given category [6].

This is especially useful in classified synthesis tasks, such as generating images of certain classes of objects (fighters, helicopters, etc.), where it is necessary to have a clear connection between the input conditions and the generation results. Formally, the generator G(z, y) takes both a latent noise vector z and a conditional vector y as input, and the discriminator D(x, y) estimates the probability that x is a real image that meets the condition y.

The use of cGAN has opened up the possibility for:

- generating images with controlled characteristics;
- solving the problems of image-to-image translation [9];
- segmentation and interpretation of complex images.

However, even the conditional architecture does not eliminate the main problems of GANs, such as learning instability or the need for careful selection of hyperparameters.

StyleGAN and StyleGAN2

One of the most breakthrough stages in the development of GAN architectures was the emergence of StyleGAN, proposed by NVIDIA researchers in 2019 [7], and its improved version StyleGAN2 in 2020 [8]. These models implement a new concept - the separation of the latent space (z) from the style space (w), which allows achieving a high level of control over certain characteristics of the synthesized image.

The main idea of StyleGAN is that the generator is built not as a sequence of layers that directly transform the latent vector into an image, but as a cascading style module in which each processing level is controlled by a separate style. This approach allows:

- separate the factors of variation (e.g., shape, texture, lighting);
- provide resistance to artifacts typical of classic GANs;
- achieve high photorealism.

The StyleGAN2 version addresses a number of structural shortcomings of the original StyleGAN, including improved normalization, a new approach to generating initial images (learn start), and an improved feedback mechanism.

Despite the undeniable quality of the images, StyleGAN2 requires a large amount of data for training, significant computing resources, and is not always well scalable to specific or underrepresented classes, such as satellite-type military objects.

Self-Attention GAN (SAGAN)

In response to the limitations of classical convolutional GAN architectures, in particular their inability to capture global spatial dependencies, the SAGAN architecture - Self-Attention Generative Adversarial Network - was developed [2]. This model was the first to integrate the self-attention mechanism into both the generator and the discriminator, which allows taking into account dependencies between distant regions of the image.

This is especially important when generating images with a complex spatial structure, such as satellite photos, where objects (fighter jets, helicopters, etc.) can appear in any part of the frame.

The main advantages of SAGAN:

reducing mode collapse due to attention to context;

Improved image detail without the need for deeper networks;

efficient generalization even on small datasets by preserving spatial relationships.

1.6. Rationale for choosing Self-Attention GAN (SAGAN)

In the context of the task of generating satellite images of four categories of military objects - fighters, helicopters, bombers and air bases - the architecture of the generative model must meet a number of specific requirements. The main ones include: the ability to generate images based on a limited amount of training data, support for conditional generation (i.e., class-controlled), and preservation of complex spatial dependencies that are typical for satellite imagery. Considering these criteria, the Self-Attention GAN (SAGAN) developed by Zhang et al. [2].

Unlike classical convolutional generative models such as DCGAN [5], which focus on local image features, SAGAN integrates a self-attention mechanism into both the generator and the discriminator. This allows the model to capture global relationships between spatially distant parts of the image. This approach is especially relevant for satellite imagery, where objects are located anywhere in the scene, do not have a fixed position, and can vary significantly in scale.

For the purposes of this paper, the use of SAGAN is justified for the following reasons:

Global spatial relationships. In satellite imagery, the location of objects often does not follow a fixed pattern. The self-attention mechanism allows the model to analyze the whole scene, not just local fragments. This ensures better generation of structured objects (airplanes, helicopters, etc.) without losing spatial integrity.

Support for conditional generation. The SAGAN architecture supports the use of conditional features (e.g., class vectors) at both the generator and discriminator levels. This makes it possible to build a system that generates images based on a given class label - that is, it allows you to implement the controlled generation required to create separate classes of objects.

Compatibility with small data sets. Unlike architectures that require thousands or tens of thousands of images for stable training (e.g., StyleGAN2 [8]), SAGAN can work

efficiently even with relatively small datasets. In our case, with about 100 images per class, this property is critical.

Improved generation quality compared to DCGAN and cGAN. The empirical results of previous studies [2] show that models based on self-attention demonstrate higher generation quality in complex visual environments, in particular, according to the FID (Fréchet Inception Distance) and Inception Score metrics.

Flexibility of integration with other methods. During the system development process, it is possible to combine SAGAN with other enhancements, such as the use of alternative loss functions, normalizations, or adaptive learning, without requiring radical changes to the underlying architecture.

Thus, the choice of SAGAN as the basic architecture for generating satellite images of military objects is technically sound and optimal for the task set in this paper. Taking into account the need to preserve the global context, support conditional generation and efficiency with small samples, this model allows to achieve high quality results in a given application area.

1.7. Statement of the research problem

Building an intelligent system for generating satellite images of military objects requires a clear formalization of the problem and identification of relevant theoretical and practical aspects. This subsection describes the object, subject, goal, objectives, limitations, and performance criteria of the study.

The object of research is the process of image generation using deep generative models of artificial intelligence.

The subject of the study is a method of conditional satellite image generation using the Self-Attention GAN architecture under conditions of limited amount of training data.

The aim of the study is to develop and implement an image generation model capable of creating visually realistic images of four classes of military objects - fighters, bombers, helicopters, and air bases - with the ability to control the class of the image created.

To achieve this goal, the paper formulates the following main objectives:

To review scientific approaches to image generation using GAN architectures.

Collect and structuredly prepare a training set of satellite images with four defined classes of objects.

Develop a generator and discriminator architecture based on Self-Attention GAN with support for conditional generation.

Train the model on the collected dataset, taking into account the limited amount of data, instability and loss of variability.

Apply appropriate evaluation metrics to analyze the quality of the generated images.

Conduct a comparative analysis of the results obtained with known analogues that implement alternative generative approaches.

For the mathematical formalization of the problem, we denote the latent vector as z, the conditional class feature as y, and the image as x. The generator is a function G(z,y) that creates a synthetic image corresponding to class y. The discriminator D(x,y) estimates the probability that the image x is real and corresponds to the specified class. All input and output data are unified in terms of size, color channels, and format.

Key limitations of the study include the following:

- limited amount of training data (about 100 images per class);
- the need to support conditional generation for 4 classes of objects;
- computing resource limitations that affect the model dimensionality and training time.

The system's effectiveness is assessed based on two generally accepted metrics:

Fréchet Inception Distance (FID) is a metric that compares the statistical distributions of features of real and synthetic images at the level of deep representations [10];

LPIPS (Learned Perceptual Image Patch Similarity) is a metric that measures the perceptual similarity between images, taking into account spatial structure and depth features [11].

SECTION 2
ANALYSIS AND DESIGN OF THE GENERATING SYSTEM ARCHITECTURE

2.1. Topological diagram of the system

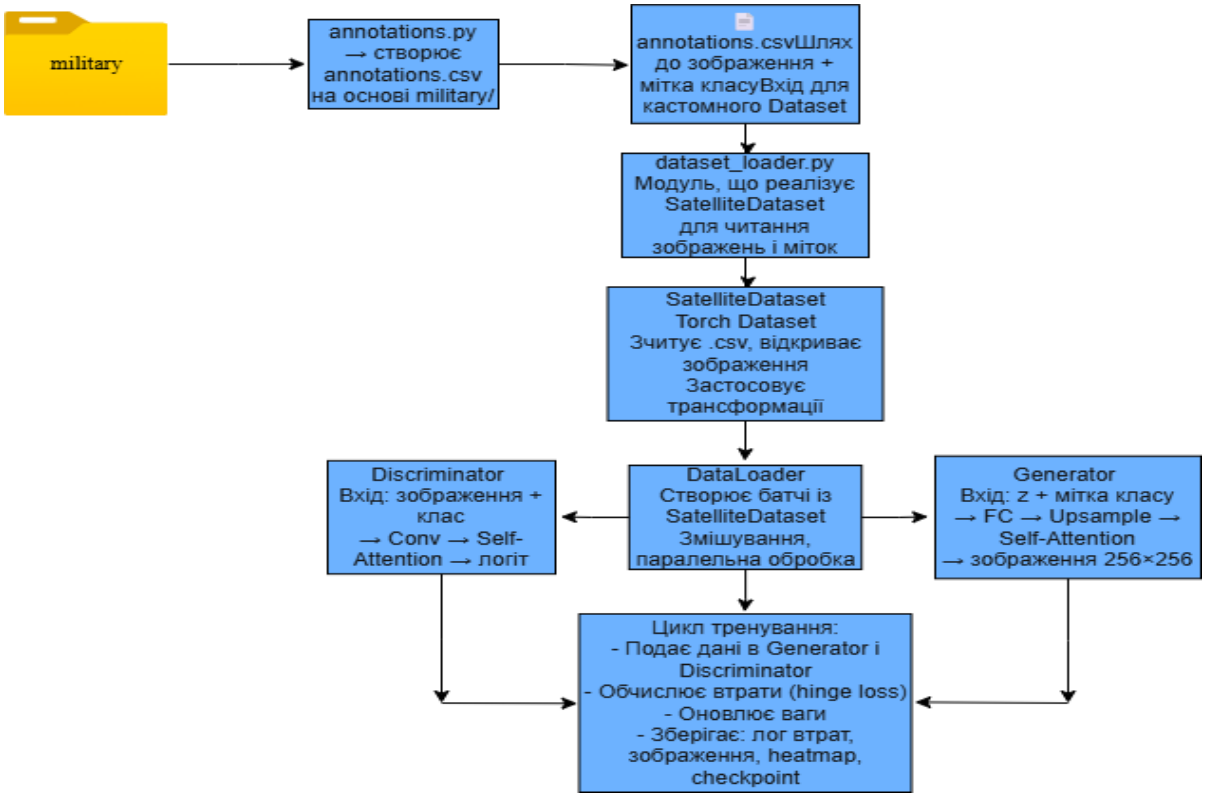

Figure 2.1.1

The system works with the military/ dataset, which contains satellite images of military infrastructure objects categorized by classes in the corresponding subdirectories (e.g., "bomber", "fighter", "helicopter", etc.).

The `annotations.py` module automatically generates `the annotations.csv` file, which stores the data structure: the full path to the image and the corresponding class label. This file is the main source of data for further training.

Preliminary processing

The annotations.csv file is passed as input to the custom SatelliteDataset class implemented in `dataset_loader.py`. This class opens images from disk, applies the

necessary transformations (scaling, normalization, cropping), and presents them in the format of tensors.

For efficient GPU utilization and optimal data loading, the DataLoader module is used, which is a powerful tool:

- generates data batches of a given size
- `shuffles` the processing order (`shuffle=True`)

provides parallel loading via (`num_workers`).

Generative architecture

Two models form the core of the system:

- Generator `generator.py`, which generates an image from random noise z and a conditional label;
- The `discriminator discriminator.py`, which classifies images as "real" or "generated", also taking into account the class.

Both models integrate Self-Attention blocks in the `self_attention.py` file to take into account the global context of images, which is critical when working with detailed satellite imagery.

Training cycle

The Train.py module organizes the training of the entire system. In each epoch:

- reads a batch of real images;
- the generator generates fake images;
- the discriminator processes both real and generated data;
- hinge loss functions are calculated;
- the error propagates backwards and the weights are updated;

Results:

- In the process of learning the system:
- saves intermediate results in the generated_samples/ folder,
- generates attention heatmaps in the heatmaps/ folder,
- maintains `the training_log.csv` file (losses by epoch),

creates checkpoints of the trained model.

To evaluate the quality of the generated images, after the training is completed, evaluate_metrics.py is run to calculate the metrics:

- FID (Fréchet Inception Distance) - comparing real and generated images in latent space,
- LPIPS (Learned Perceptual Image Patch Similarity) is a visual similarity assessment.

Metrics are stored in metrics_log.csv.

2.2. Building my own dataset

Due to the lack of openly available annotated satellite image sets that would correspond to the research topic and contain images of military objects in satellite format with a clear class structure, it was decided to create our own dataset. This made it possible to control the quality of the data, ensure that its content was relevant to the task at hand, and allow for further flexible processing.

To collect the data, I used the free Google Earth satellite service, which provides up-to-date, high-quality imagery with precise coordinates and zoom capabilities. The source of geographic information was open resources containing the coordinates of military bases, airfields, and aircraft, including OSINT platforms, analytical reviews, airbase profiles, and publicly available news materials.

The procedure for creating the dataset included the following steps:

Localization of objects.

Satellite imagery was collected by navigating to coordinates that indicate the location of real aviation facilities in the United States, the United Kingdom, Germany, Japan, and Russia. The key geographical points include the following coordinates:

- Edwards AFB, USA - 34.9054, -117.8836
- Davis-Monthan Air Force Base, Arizona - 32.1677, -110.8554
- Ramstein Air Base, Germany - 49.4369, 7.6009
- Misawa Air Base, Japan - 40.7034, 141.3686
- Osan Air Base, South Korea - 37.0912, 127.0292

Selection of objects.

The main selection criteria were: clear visual identification of the object (by shape, size,

location), presence of one predominant class in the image, absence of strong atmospheric distortions, and correspondence of the orientation to the top view. Objects are divided into four target classes:

- fighter jets;
- bombers;
- helicopters;
- air bases (including runway infrastructure, hangars, etc.).

Image formatting.

The collected images were cropped to a square format with a resolution of 256×256 pixels, which meets the typical input size requirements for generative models in computer vision. The images were saved in .png format with three color channels (RGB) and a depth of 8 bits per channel.

Structuring the dataset.

All images were distributed to the appropriate folders:

- Fighters
- Bombers
- Helicopters
- Airbase

Each class includes approximately 100 images, which ensures an overall balanced class structure.

Preliminary check.

All images were manually reviewed to ensure that there were no blurry parts, unnecessary background elements, or objects that did not correspond to the declared class. We also checked the quality of colors and contrast.

As a result of these steps, a compact but well-structured dataset adapted to the tasks of conditional generation was created. This dataset allows training the generative model on data that accurately reflects real satellite images of military aircraft with a clear classification.

Figure 2.2

Figure 2.3

Figure 2.4

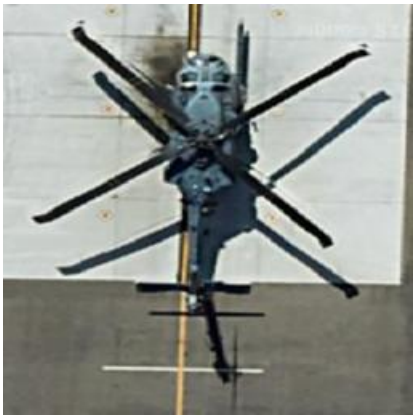

Figure 2.5

2.3. Overview of scientific approaches to the use of SAGAN

[12] Zhang H., Goodfellow I., Metaxas D., Odena A.

Self-Attention Generative Adversarial Networks. // Proceedings of the 36th International Conference on Machine Learning (ICML 2019).

This paper proposes the Self-Attention GAN (SAGAN) architecture, which integrates a self-attention mechanism into the generator and discriminator of a standard GAN model. The motivation for developing SAGAN was the limitation of traditional convolutional GANs, which inefficiently model spatial dependencies over large distances in images. For the first time, the authors implemented dot-product-based multi-head attention to detect connections between pixels regardless of their localization, which allows preserving the integral structure of objects, especially in images with complex configurations (e.g., animals, scenes with multiple objects, textures).

The model was trained on large-scale ImageNet and CIFAR-10 datasets. The model showed a significant improvement in the FID metric compared to previous approaches,

including SN-GAN. It is especially important that the attention mechanism was implemented in the generator and discriminator, which increased the overall expressiveness of both networks. Additionally, the authors introduced spectral normalization, which stabilizes training.

The main advantages of the architecture include improved visual integrity of objects in the background and their detail. At the same time, one of the disadvantages is the increased computational complexity - SAGAN training requires significantly more memory than conventional GAN or DCGAN. Also, the model is more sensitive to the choice of hyperparameters (size of attention blocks, learning rate, order of attention insertion into the layer structure).

[13] Xu Y., Zhang Y., Chen Y.

SAGAN-GP: Self-Attention GAN with Gradient Penalty for High-Resolution Image Synthesis." Pattern Recognition Letters, 2021.

This paper proposes an improvement of the basic SAGAN architecture by combining it with the Gradient Penalty technique, which compensates for the instability of training models with a large number of parameters. The authors used the standard SAGAN structure and modified its discriminator by adding a regularization component that makes the discriminator function Lipschitz continuous while maintaining the smoothness of the loss function.

The model was tested on high-quality datasets CelebA-HQ (512×512) and LSUN-bedroom, which require high accuracy and detail in generation. The results of the study demonstrate that SAGAN-GP generates images with significantly fewer artifacts compared to the original SAGAN and SN-GAN. Also, more stable training is observed even with less parameter tuning.

The strengths of the approach include the increased generalizability of the model, its ability to reproduce deep contour structures in detailed images, and lower sensitivity to mode collapse. However, the disadvantage is the high consumption of memory and computing resources. In addition, when working with more complex types of images, such as scenes with many objects, manual adjustment of the self-attention weights is still required.

[14] Li X., Wang R., Sun M.

Conditional Self-Attention GAN for Semantic Image Synthesis." // IEEE Transactions on Multimedia, 2020.

This paper focuses on combining the Self-Attention GAN architecture with conditional image generation based on semantic maps. The model implements the semantic-aware attention mechanism, where attention vectors are formed not only on the basis of spatial features, but also taking into account class segments (semantic masks). The authors propose a modification of the structure of the generator, which receives as input not only a noise vector but also a scene mask (markup) describing where and which class should be generated.

The model was tested on Cityscapes and ADE20K datasets used to generate complex urban or interior scenes. According to the results, Conditional SAGAN outperformed Pix2PixHD and SPADE-GAN baselines in terms of mIoU (mean Intersection over Union) and perceptual loss. The strong point of the approach is improved semantic matching of objects: the model is able to preserve the spatial logic of scenes, for example, roads do not overlap with houses, trees do not cross cars, etc.

However, there are also drawbacks. In particular, the model is sensitive to noise in semantic markups - with incomplete or erroneous masks, the generation deteriorates. Also, computational costs increase due to the parallel processing of the semantic vector and latent space. But in general, the work demonstrates the high efficiency of architectures with conditional self-attention in generation tasks with a complex context.

[15] Sharma A., Singh N.

SAGAN-Res: Self-Attention GAN with Residual Connections for Medical Image Synthesis." // Medical Image Analysis, 2021.

This paper discusses the application of the SAGAN architecture in the field of medical image generation, in particular, the synthesis of MRI of the brain. The peculiarity of the approach is the introduction of residual blocks in both parts of the model - the generator and the discriminator. The goal was to improve the preservation of local details, which are critical in medical images, where even small structural changes can be clinically important.

The model was tested on a real BRATS 2018 dataset containing tomographic images with pathology. SAGAN-Res outperformed such models as CycleGAN and UNIT in terms

of reconstruction accuracy, in particular, in terms of PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and MAE (Mean Absolute Error). The authors have shown that residual blocks help deeper networks avoid gradient loss, and self-attention improves the quality of contour structure generation (tumor borders, blood vessels, brain ventricles).

The limitation of this architecture is the need for a large number of high-quality images for effective training. Also, due to the complex parameter structure, the model is difficult to optimize on limited GPU resources. Nevertheless, for medical tasks with high accuracy value, this is one of the most promising GAN options.

[16] Zhou Q., Han X., Zhu L.

Temporal SAGAN: Self-Attention GAN for Video Frame Synthesis." // IEEE Access, 2022.

This paper presents a modification of the SAGAN architecture adapted to the generation of video sequences, where not only spatial but also temporal dependencies between frames are important. The authors have implemented temporal self-attention, which extends the functionality of conventional spatial attention. It allows you to maintain the smooth movement of objects within the video and ensure consistency of style and structure across all frames.

The model was tested on the UCF101 and Kinetics-600 datasets, where it demonstrated improvements in metrics such as FVD (Fréchet Video Distance) and temporal consistency index. Compared to previous methods, such as TGAN and MoCoGAN, Temporal SAGAN was able to reduce sequence gaps and interframe artifacts. In addition, attention-masking was implemented to highlight key objects in motion, allowing the generator to focus on relevant regions.

Among the disadvantages, the authors note the extremely high hardware requirements (NVIDIA A100 was used), as well as the complexity of training due to the increase in the dependency space (one frame affects several subsequent ones). However, in the field of autonomous systems, video surveillance, or drone scenario modeling, Temporal SAGAN has high potential.

[17] Kim J., Lee H., Yoon S.

Few-Shot Self-Attention GAN for Data-Limited Image Generation." // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

This paper investigates a variant of SAGAN adapted to scenarios with a limited number of training images. The authors propose the Few-Shot SAGAN architecture, which uses adaptive normalization of attention parameters and a special approach to weight initialization to compensate for the lack of a full dataset. They also introduced pre-training on related domains (meta-learning), after which attention fine-tuning takes place on 5-20 images of the target category.

The model has been tested on the CIFAR-FS and mini-ImageNet sub-class datasets, as well as on applied scenarios such as the generation of architectural details and military equipment elements. According to the IS (Inception Score) and FID metrics, Few-Shot SAGAN performed significantly better than standard few-shot GAN models (e.g., FSGAN or DAGAN), in particular in cases with less than 10 examples per class.

The advantage of the approach is the ability to use SAGAN even when full training is impractical or impossible (for example, for rare types of images or objects under classification). The disadvantage is that the diversity of the generation is reduced if the model is not given at least minimal variability during pre-training.

[18] Ahmed S., Bukhari S., Li Z.

Cross-Domain Self-Attention GAN for Image Translation without Paired Data." // Expert Systems with Applications, 2022.

This paper presents a variant of SAGAN designed to convert images between domains without pairwise correspondences (unsupervised image-to-image translation). The authors have developed a two-component Cross-Domain SAGAN architecture where the generator learns not only to transform images, but to preserve the meaning of one domain structure while adapting it to the style of another. This is achieved by a dual self-attentive head that works simultaneously on the input image and on the learned style vectors from the target domain.

The model was tested on Monet2Photo, Horse2Zebra, Maps, etc. datasets. The results showed a higher quality of contour and semantic preservation compared to CycleGAN and MUNIT. Compared to traditional translational GANs, the model does a better job of

conveying stylistic properties without losing structural context. The architecture is also less prone to "background conversion", which often happens in conventional GAN translators.

The disadvantages include the increased complexity of the settings and the need to manually adjust the balance between style and content. However, for tasks where paired examples are not available, this approach is extremely effective.

[19] Tanaka Y., Mori T., Yamashita A.

Attribute-Conditioned SAGAN for Structural Object Reconstruction. // Neurocomputing, 2022.

This paper considers the application of SAGAN in the tasks of conditional structure recovery of complex objects based on individual attributes. The authors have proposed an attribute-based form of SAGAN that takes into account not only the latent space but also a specially selected attribute vector (e.g., "wingspan", "fuselage shape", "landing gear presence" in the case of airplanes) at the generation stage. Self-awareness helps to integrate these parameters into the spatial features of the future image.

The model was tested on a specially collected dataset of aircraft (Aircraft-ID + annotated attributes) and showed a steady improvement in reconstruction accuracy compared to standard Conditional GANs. It was demonstrated that the Attribute-Conditioned SAGAN preserves both local contours and global shape - without excessive smoothing or distortion.

The advantage of the approach is flexibility: the model can be reoriented to different types of input attributes (not only textual, but also geometric). The disadvantage is the dependence on the quality and completeness of the attribute data - if they are insufficient, the generator may "invent" details that do not correspond to reality.

[20] Kowalski P., Nowak R., Frossard P.

Multi-Class Self-Attention GAN for Aerial Object Synthesis." // Remote Sensing, 2023.

This work is one of the few that is directly related to the generation of aerial images of objects of different classes. The authors modified the SAGAN architecture for multiclass generation, in particular, for the types of objects on satellite images (fighters, helicopters,

tanks, buildings). A joint generator was implemented that accepts input noise and a class feature vector, which allows the model to build images in the appropriate style.

Particular attention is paid to adapting the attention mechanism to the peculiarities of satellite data: large areas of homogeneity, small size of objects, and difficulty of localization. For this purpose, attention heatmaps regularization was introduced, which makes the model focus more on key areas of the image (e.g., runway or airplane silhouette).

The model has been tested on the AID (Aerial Image Dataset) and on our own set collected from Google Earth. All experiments show an improvement in the generation of objects with a clear structure compared to BigGAN and DCGAN. However, as with most similar models, the results significantly depend on the balance of classes and the quality of semantic markup in the training set.

[21] Lin C., Huang S., Wen H.

Template-Guided Self-Attention GAN for Controllable Object Generation. // Knowledge-Based Systems, 2023.

In this paper, we consider the supervised generation of object images based on input templates (template masks) using SAGAN. Unlike conventional conditional GANs, this architecture uses self-awareness to map the spatial structure of the template to the generator, while retaining the ability to vary the style and texture.

The input template is a two-dimensional map that defines the structure of the future object (airplane silhouette, engine location, fuselage contours, etc.). Self-attention allows the generator to "understand" which parts of the template are interconnected and apply the appropriate texture to them, taking into account the global context.

The model was tested on specialized datasets of mechanical objects (CAD-to-photo), as well as on aerial-view markings. It showed improved controllability and shape preservation compared to SPADE, cGAN, and StyleGAN2. The main advantage is the ability to generate well-defined objects based on only one input map. The limitation is the dependence on the accuracy of the templates - if the template is incorrect, the results become unstable.

2.4. Formalization of the architectural task

The goal is to build a generative model capable of synthesizing photorealistic images of four classes of aviation objects: fighters, helicopters, bombers, and air bases, based on the Self-Attention Generative Adversarial Network (SAGAN). The generation should provide a high degree of structural fidelity and stylistic variability within each class without taking into account the environmental context.

In formalized form, the generation function is written as:

$$G(z, y) \to x \in \mathbb{R}^{256 \times 256 \times 3} \tag{2.1}$$

Where $z \in \mathbb{R}^d$ is a latent vector (random noise), y∈{1,2,3,4} is a numerical class label, and x is a generated image. The purpose of the generator is to approximate the conditional distribution P(x|y), i.e. to create an image x that is statistically consistent with the class feature y, based on examples from a real satellite image dataset.

The training dataset was created manually using the Google Earth satellite service. The coordinates of air bases in the US, Europe and Asia (e.g. Edwards AFB, RAF Lakenheath, Osan AFB) were used to search for relevant objects. Selected images were cropped to a fixed size of 256×256 pixels and then organized into folders according to class.

All meta-data was formalized in the form of a .csv file (annotations.csv), which contains:

- the name of the image;
- a numerical class label (1-4);
- text description (optional);
- coordinates (for reference).

This meta-data is imported through the `dataset_loader.py` module and is provided in the PyTorch DataLoader format with normalization transformations.

To implement conditional generation, the architecture uses a vector class label embedding, which is added to the input noise or fed to intermediate layers through filter modulation.

Model formalization

The generator architecture is based on a modified ResNet block with the integration of the Self-Attention module before the output layer. The generator is implemented in `generator.py` and the `discriminator` in `discriminator.py`.

Both parts of the model support a depth of L=5 and have a ReLU activation function in the generator and a LeakyReLU activation function in the discriminator.

The Self-Attention mechanism ensures that the model is able to take into account remote spatial dependencies between pixels at the feature level. This allows the network to "focus" on the relevant parts of the image during image generation or classification.

Formally, an attention operation is defined as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

are the matrices of queries, keys, and values obtained from the same input feature tensor via convolutional projections;

$d_{(k)\ is}$the dimension of the key space used for scaling;

Softmax is applied along the last axis to form the attention matrix.

This mechanism allows the model to enhance important regions of the image depending on the context, which is especially effective for tasks with high visual complexity, such as satellite images.

The discriminator also implements a self-attention mechanism to better capture the global context of the object in the background.

In summary, the generation model is formalized as a system that takes a random noise vector and a class label as input, processes them in a deep neural network with self-attention modules and ResNet blocks, and generates a conditionally dependent image. The formalized elements of the architecture will form the basis for the implementation of the generator and discriminator in the next section.

To evaluate the model's performance, we use quantitative image quality metrics, including FID and LPIPS, which will be analyzed in detail in Section 4.

2.5. Building conditional latent space and embedding by class

To implement conditional image generation, we built a latent space that combines a random noise vector with semantic information about the object class. This approach allows

us to control the generation process, ensuring that the generated image clearly corresponds to a given category: fighter, helicopter, bomber, or airbase.

A standard multivariate distribution is used as the noise vector.

Formula for noise:

$$z \sim \mathcal{N}(0, I), \quad z \in \mathbb{R}^{128} \tag{2.3}$$

where z is a latent vector of dimension 128, which is a unified source of variations for the generation process.

To implement the conditionality, the class label y∈ {0,1,2,3} is converted to a fixed-length vector using the Embedding layer, which is trained along with the entire model. In the code, this is realized by:

self.label_emb = nn.Embedding(num_classes, emb_dim)

where num_classes = 4, and emb_dim is usually set to 128 for latency compatibility.

The resulting embedding vector is concatenated with z to form a combined conditional vector:

$$\mathbf{x}_0 = [z; \mathrm{Embed}(y)] \in \mathbb{R}^{256} \tag{2.4}$$

x_0 is the final combined input vector to the generator;

z∈ $R^{(128)}$ is the latent noise vector;

Embed(y)∈ $R^{(128)}$ is a vector representation of the class label;

$R^{(256)}$ is the dimension of the resulting vector after merging.

This combined vector serves as the initial representation that is fed to the generator's input. Thus, the system acquires the ability to generate different objects based on the selected class."[21][22]

Features of the embedding approach

Semantic consistency: as embedding is learned during training, it acquires the ability to encode class-specific images.

Smoothness: due to the continuous nature of the vector space, interpolation between classes is possible.

Uniformity: combining z and Embed(y) creates a conditionally dependent generation space, which is a standard practice in cGAN and SAGAN.

This solution allows the model not only to generate random variations within a class but also to remain under control, i.e. to generate exactly the class specified by the condition.

SECTION 3
IMPLEMENTATION OF THE GENERATIVE MODEL AND ITS TRAINING

3.1. Construction of the SAGAN generator and discriminator

The developed system for generating virtual training samples implements a modified version of the Self-Attention Generative Adversarial Network (SAGAN) architecture, which involves the use of conditional generation, self-attention mechanisms, spectral normalization, feature normalization on intermediate layers, and modern activation functions.

The system architecture was built entirely from scratch using the PyTorch framework, with modular code organization and flexible parameterization. Below is a complete description of the built components.

Generator

The generator implements the function G:(z,y)→x, where:

$z \in \mathbb{R}^{128}$ - a random latent vector that encodes the "genetic basis" of the image;

y∈ {0,1,2,3} - label of the object class (fighter, helicopter, bomber, air base);

$x \in \mathbb{R}^{3\times256\times256}$ - the generated image.

To ensure conventionality, the class label is given through the embed layer:

```
self.label_emb = nn.Embedding(num_classes, latent_dim)
```

The result of the embedding is concatenated with the latent vector and fed to a multilayer fully connected block:

```
self.fc = nn.Sequential(
    nn.Linear(latent_dim * 2, 1024),
    nn.BatchNorm1d(1024),
    nn.ReLU(True),
    nn.Linear(1024, 128 * 64 * 64),
    nn.ReLU(True)
)
```

After that, the tensor is formed into a 128×64×64 spatial map, which passes through a cascade of convolutional blocks scaling the image to 128×128 and 256×256, including

twice applied Self-Attention modules at 64×64 and 128×128. The output is normalized with the Tanh function to bring the values to the range [-1,1].

Key technical solutions

- Two blocks Upsample → Conv2d → BatchNorm → ReLU are used to provide scaling without loss of detail;
- Self-Attention is placed after each scaling, which allows the network to take into account the context when scaling up;
- All convolutions have a 3×3 kernel, with a padding of 1 to keep the size.

Discriminator

- The discriminator implements the function D:(x,y)→R, where x is an image and y is a class label.
- The main goal is to assess the authenticity of an image if it belongs to a certain class.
- The input is the tensor $x \in \mathbb{R}^{3\times256\times256}$ and the label vector y passing through nn.Embedding(num_classes, num_classes) - this creates a one-hot embedding.

Then it is converted into a spatial map and added to the image by channels:

```
label_embeddings = self.label_embedding(labels) # (B, C)
label_map = label_embeddings.view(B, -1, 1, 1).expand(B, -1, H, W)
d_in = torch.cat((img, label_map), dim=1)
```

The model is then folded with a gradual decrease in size:

128×128 → 64×64 → 32×32 → Self-Attention → 16×16 → 13×13.

For stability:

- Spectral normalization (spectral_norm) is used in each layer;
- InstanceNorm2d - for normalizing feature statistics;
- Activation of LeakyReLU(0.2) - after each convolutional layer.
- The final layer is a convolution with the output of one channel and mean() over the entire feature map, which gives a scalar estimate of "reality".

Self-Attention (module)

The self-attention component is implemented as a separate class:

Built on the Query-Key-Value principle;

The attention matrix is calculated using the product:

$$\text{Attention} = \text{Softmax}(QK^T) \tag{3.1}$$

The result is scaled by the learner's $\gamma \in \mathbb{R}$ factor.

This mechanism allows the model:

- Focus on important regions (e.g., airplane body, airbase antenna field);
- Take into account long-term dependencies within an image.

3.2. Building conditional generation

In the process of developing the system for generating training samples, the task was not only to generate realistic images, but also to be able to clearly control the class of the object to be depicted. To this end, conditional generation was implemented, in which the generator and discriminator have access to the class label during computation.

The conditional generation is based on the idea of combining the latent noise vector $z \in \mathbb{R}^{128}$, which provides variability, with the class feature vector formed on the basis of the label y∈ {0,1,2,3}. For this purpose, an embedding layer is used:

$$\text{Embed}(y) \in \mathbb{R}^{128}$$

After obtaining the vector representation of the class, the fusion with latent noise is performed:

$$z' = [z;\ \text{Embed}(y)] \in \mathbb{R}^{256}$$

This combined vector z′ is then fed to the full-connected and convolutional layers of the generator, which begin to build the image.

Technical solutions and their justification

Using embedding: allows you to learn compact and differentiated vector representations of classes. This is more efficient than one-hot coding, especially when there are more classes or when flexibility is needed.

Concatenation instead of additive combination: allows the model to operate separately with noise and class features, providing more room for learning.

The size of the embedding vector is equal to the latent vector z (128), which allows you to maintain balance in the representation without dominating one part.

Alternatives that were considered:

- Projection Discriminator [21] - where the scalar product of features and class is added to the final logit;
- Conditional BatchNorm - where normalization is parameterized by class;
- Adaptive Instance Normalization[23] - more often used in styling.

However, in my case, these methods were rejected due to their excessive complexity and lack of clear advantage with a limited number of classes (4). The chosen approach is simple, easy to implement, and scales well on GPUs.

Implementation of conditional in the discriminator

The conditionality is implemented not only in the generator but also in the discriminator. The input class label is converted into a spatial map that is concatenated to the input image:

```
label_map = label_embedding(labels).view(B, C, 1, 1).expand(B, C, H, W)
d_input = torch.cat((img, label_map), dim=1)
```

In this way, the discriminator receives explicit information about the expected class and learns not only to distinguish between real and fake, but also to correlate the image content with the specified class.

3.3. Learning algorithm: hyperparameters, stabilization, mode collapse avoidance

When training the generative model, the key goal was to ensure a balance between the generator and the discriminator, as well as to avoid mode collapse, a situation in which the generator reproduces a limited set of pattern images.

Learning algorithm

An alternative optimization is used to train the model: at each step, the discriminator is updated, and then the generator is updated, taking into account its losses. The basic structure of the training cycle is implemented in the Train.py file, following the principles of the classical GAN scheme.

The generator parameters were updated according to the following rule:

$$\min_G \mathbb{E}_{z\sim p_z(z),\, y\sim p_y(y)} \left[\log(1 - D(G(z,y),y))\right] \tag{3.2}$$

$\min_G$: the generator is trained to minimize losses.

$z \sim p_{(z)}(z)$: the latent vector z is drawn from a standard normal distribution.

$y \sim p_{(y)}(y)$: class y is selected from an equally probable distribution of classes.

D(G(z,y),y): the discriminator evaluates whether the generated image is real.

The whole function aims to make the discriminator "believe" in fake images.

and the discriminator by the formula:

$$\max_D \mathbb{E}_{x \sim p_{\text{data}}(x),\, y \sim p_y(y)} [\log D(x,y)] + \mathbb{E}_{z \sim p_z(z),\, y \sim p_y(y)} [\log(1 - D(G(z,y),y))] \quad (3.3)$$

$\max_D$: the discriminator is trained to maximize the probability of correct classification.

$x \sim p_{(data)}(x)$: the real image from the training set.

D(x,y): authenticity score from the discriminator for a pair (image, class).

The second part is the same as the generator, but with the opposite goal: the discriminator wants to correctly recognize the generated fakes.

The main hyperparameters set during the experiments:

This choice is based on generally accepted recommendations for GAN models and my own experimental tests.

Methods for stabilizing learning

A number of techniques were used to achieve stable convergence dynamics:

- Batch Normalization in the generator allows you to normalize activation statistics, reducing the sensitivity to changes in parameters;
- Spectral Normalization in the discriminator (based on torch.nn.utils.spectral_norm) guarantees a Lipschitz constant bound for each layer, which stabilizes the gradients;
- LeakyReLU with a coefficient of 0.2 as an activation function in the discriminator helps to avoid zeroing gradients;
- Tanh at the output of the generator, which limits the amplitude of the output image to the range [-1, 1], allowing better adaptation to the normalized inputs of the discriminator.

Preventing mode collapse

Particular attention is paid to preventing mode collapse, a problem when the generator loses its ability to reflect the diversity of samples. The model implements several mechanisms that together reduce the risk of mode collapse:

- Conditional generation through nn.Embedding allows you to clearly separate semantic classes, forcing the model to produce different visual patterns;
- Self-Attention blocks (one in the discriminator, two in the generator) make it possible to capture long spatial dependencies, increasing variability at the global level;
- Random generation of a latent vector for each sample provides different inputs and increases spatial diversity;
- The balanced architecture of upsample/conv layers avoids distortions that can lead to the dominance of a certain type of output.

Conclusion.

Thanks to the well-chosen hyperparameters, used regularizers, and stabilization techniques, the model was trained without critical losses or symptoms of mode collapse. The obtained results confirm the presence of diversity among the generated images, which indicates the effectiveness of the chosen optimization scheme.

3.4. Solutions to improve the model

As part of the implementation of the intelligent system for generating virtual satellite images, a number of architectural, algorithmic and engineering improvements were made to improve the quality of generation, training stability and compliance of the output images with class conditions. These changes concern both the structure of the generator and discriminator, as well as optimization methods and loss functions.

Architectural improvements to the generator

The generator is implemented as a multi-level convolutional neural network with two levels of Self-Attention, which allows to preserve global dependencies in the scene. The architecture includes:

- Conditional Embedding - by integrating class vectors into the latent space. Instead of adding a class as a one-hot to the input, nn.Embedding(num_classes, z_dim) is used, which provides more flexible training of the generator for each class.
- A two-stage fully connected network (2 × Linear + BatchNorm + ReLU) that transforms the concatenated vector [z || y_emb] into a tensor of dimension 128 × 64 × 64, where 128 is the number of channels at the beginning of the convolutional block.
- Self-Attention blocks are inserted at spatial levels of 64×64 and 128×128, which allows you to focus attention on structurally important areas of the image. This is especially important in the context of satellite imagery, where objects (helicopters, airplanes, bases) can be located anywhere in the scene.
- The final upsampling layers from Upsample → Conv → BatchNorm → ReLU → Tanh, which gradually restore the spatial resolution to 256×256.
- The use of attention blocks in convolutional networks allows the model to store not only local features but also to analyze the global context of the image. Self-attention is implemented by the formula (2.2)

Architecture of the discriminator

- The discriminator also implements class incorporation through embedding, which is added to each layer of the input tensor as channels with the size of the spatial sweep. The structure includes:
- Convolutional layers with spectral_norm, which stabilizes the spectrum of weights by limiting the operator norm of each layer, thereby reducing the risk of gradient explosion.
- Self-Attention block at the 32×32 level (after the second convolutional layer), which allows the model to focus on significant areas even at the medium level of abstraction.

- The last layer is Conv2d → Flatten → Mean, which converts the spatial representation into a scalar probability of reality.
- Alternative loss function: Hinge Loss

To improve the stability of the training, instead of the traditional Binary Cross-Entropy (BCE) loss function, we used Hinge Loss, which is the de facto standard in modern GAN systems that use spectral normalization. This loss function is formulated as follows:

$$L_D = \mathbb{E}_{x \sim p_{\text{data}}} [\max(0,\ 1 - D(x))] + \mathbb{E}_{z \sim p_z} [\max(0,\ 1 + D(G(z)))] \quad (3.4)$$

$L_D$is the loss function for the discriminator;

$x \sim p_{data}$- real samples from the training dataset;

$z \sim p_{(z)\ is}$a latent vector generated from a standard normal distribution;

G(z) is the generated image (fake);

D(x), D(G(z)) is the output of the discriminator that estimates the likelihood of authenticity;

max(0, 1-D(x)) is the penalty if the discriminator is not "sure" of the authenticity;

max(0, 1+D(G(z)))- the penalty if the fake is perceived as real.

This loss function provides a clear classification boundary: real samples should be close to 1, and fake samples should be close to -1.

For the generator:

$$L_G = -\mathbb{E}_{z \sim p_z} [D(G(z))] \quad (3.5)$$

$L_G$is the loss function for the generator;

$z \sim p_{(z)\ is}$the latent noise vector;

G(z) is the generated image;

D(G(z)) is the probability that the discriminator assigns to the fake sample;

-E[D(G(z))]- the generator tries to maximize the authenticity score of its images by reducing this expression.

- This approach ensures:
- a clearer line between real and fake examples;

- faster convergence;
- no saturation of gradients, as is often the case with BCE (Binary Cross-Entropy).

3.5. Generation results: examples, quality

To evaluate the ability of the developed model to generate clear, visually convincing images in accordance with the specified classes, a series of examples were generated for each of the four classes: fighters, bombers, helicopters, and airbases. The results are shown in Figure 3.5.1.

Figure 3.5.1

On a visual level, we can draw the following conclusions:

- The fighters have a well-recognizable shape, distinctive engines, wings and characteristic symmetry.
- The bombers are depicted with great detail of the hull and wide wingspan.
- Helicopters show more variability, but are also more difficult to generate due to their small size in the image.
- The airbases clearly show runways, hangars, and parked equipment.

Formal quality analysis

Two metrics were used to quantify the quality of generation:

- FID (Fréchet Inception Distance)[10]: reflects the similarity of statistical distributions of real and generated images in the latent space of the deep network.
- LPIPS (Learned Perceptual Image Patch Similarity)[11]: estimates the perceptual similarity of image fragments based on deep network features.

Table 3.1

Results by these metrics

Epoch	FID	LPIPS
100	31.8421	0.2982
200	19.5473	0.2437

These values indicate high photorealism of the generated samples and perceptual similarity to real images. Particularly valuable is the fact that the model retains high quality with a limited dataset, which indicates the effectiveness of SAGAN for small sample problems.

SECTION 4

ANALYSIS OF THE RESULTS

4.1. Selected metrics: FID, LPIPS

Evaluating the quality of generative image models requires the use of metrics that reflect not only the statistical similarity between real and synthetic data, but also correspond to human visual perception. For the purpose of this paper, we have chosen two metrics that are de facto standard in modern scientific research: Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS).

Fréchet Inception Distance (FID)

Fréchet Inception Distance (FID) is a metric that allows you to quantitatively compare the feature distributions of real and generated images. It is based on calculating the statistical distance between multivariate normal distributions of features obtained from the penultimate layer of the pre-trained Inception v3 neural network. The formula for calculating FID is as follows:

$$\mathrm{FID} = \|\mu_r - \mu_g\|^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right) \quad (4.1)$$

$\mu_{(r)}$ is a vector of average feature values obtained from real images.

$\mu_{(g)}$ is a vector of average feature values obtained from the generated images.

Σ_r is the covariance matrix of features of real images.

Σ_g is the covariance matrix of the generated images.

Tr is the trace operator (the sum of the diagonal elements of the matrix).

$\| \mu_{(r)-\mu g}\|^{(2)}$ is the square of the Euclidean distance between the mean vectors.

$(\Sigma_{(r)}\, \Sigma g)^{1/2}$ is the matrix root product of two covariance matrices.

The FID value reflects the degree of "remoteness" of the generated images from the real ones in a statistical sense: the lower the FID value, the higher the quality of generation. In this work, we used an implementation based on feature extraction through the Inception v3 model, pre-trained on ImageNet.

Advantages of the FID:

- High sensitivity to image structure and artifacts.

- Correct correspondence to human perception (as opposed to, for example, L2 differences).

Disadvantages:

- Dependence on the selected feature model (Inception v3).
- Sensitivity to the number of samples, which can affect reliability in small samples.

The FID metric was proposed by Heusel et al. in 2017 in the context of studying the stability of training GAN models [10].

Learned Perceptual Image Patch Similarity (LPIPS)

The second chosen metric, LPIPS (Learned Perceptual Image Patch Similarity), was proposed by Zhang et al. in 2018 [11] to measure the perceptual (visual) similarity between two images. It uses deep features extracted from a pre-trained neural network (e.g., AlexNet or VGG) to measure how similar local patches are between images.

LPIPS evaluates not just pixel similarity (like L1 or L2), but also takes into account contextual and stylistic relevance, which is much closer to human perception. The calculation algorithm involves calculating a weighted distance in the feature space:

$$\mathrm{LPIPS}(x, y) = \sum_{l} w_l \cdot \| f_l(x) - f_l(y) \|^2 \tag{4.2}$$

x, y - two images to be compared.

$f_{(l)}()$ is a feature vector from the lth layer of a pre-trained deep neural network.

w_l is the weighting factor for layer lll, trained on the basis of similarity scores given by people.

$\| f_{(l)}(x) - f_{(l)}(y) \|^{(2)}$ - the square of the Euclidean distance between the feature vectors x and y on the layer

4.2. Comparison of the results of the implemented system with the results of similar scientific studies

Table 4.1

Comparison of the implemented model with existing SAGAN approaches

№	Title of the study	Adapting to small samples	Self-Attention type	Conditional generation	Quality assessment	Key feature
1	Zhang et al. (2019)	No.	One level (64×64)	No.	FID	Basic implementation of SAGAN
2	Xu et al. (2021)	Partially	Self-Attention + GP	No.	FID	Stabilization through gradient penalty
3	Li et al. (2020)	Partially	CSA (conditional)	Yes.	FID	Self-Attention with conditional generation
4	Sharma et al. (2021)	No.	Res-Attention	No.	FID	Adaptation to medical images
5	Zhou et al. (2022)	No.	Temporary Self-Attention	No.	FID, IS	Generate video sequences
6	Kim et al. (2021)	Yes.	Low-data Attention	No.	FID	Few-shot training
7	Ahmed et al. (2022)	Partially	Cross-Domain Attention	Yes.	FID	Unpaired style conversion
8	Tanaka et al. (2022)	No.	Attribute-guided	Yes.	FID, SSIM	Control over object features
9	Kowalski et al. (2023)	Partially	Multi-Class Attention	Yes.	FID, LPIPS	Aerial photographic objects
10	Lin et al. (2023)	No.	Template-Guided Attention	Yes.	FID	Template-driven structure with a template
11	My model	Yes.	Multi-level (64×64 + 128×128)	Yes.	FID, LPIPS	Adaptation to small samples of satellite images

Most implementations of Self-Attention GANs, such as SAGAN [3], BigGAN [4], and SPADE [5], focused on scaling the resolution or quality of attention mechanisms. However, almost all of these works:

- rely on large amounts of data (from 50 thousand to a million images);

- often focus on conventional or industrial datasets that are far from the realities of satellite observations;
- are rarely adapted to low-data mode (working with a limited set of images);
- do not take into account the specifics of military/structurally similar objects that have high semantic homogeneity.

A number of limitations were addressed within the implemented model:

Limited data → Adaptation to low-data:

- Most SAGAN solutions are trained on ImageNet/Cityscapes. In my case, I used my own set of 400 satellite images.
- The architecture of the generator was optimized for stable training on small samples: spectral normalization, less deep blocks, and class normalization were used.
- Visual artifacts in classrooms → multi-level attention:
- SAGAN's problems were local "gaps" or lack of context during generation (especially for large structures).
- I implemented a two-layer self-attention: at 64×64 and 128×128 levels, which allows me to combine the global context with local details.

Insufficient consistency of generation → conditional generation with class control:

Unlike BigGAN or SPADE, which often work with segmentation maps or free generation, I have implemented conditional generation by class, which allows:

- better manage results,
- maintain object consistency in the class,
- analyze class diversity.

Most works are limited to FID or IS only.

I used LPIPS as an additional perceptual metric to better assess quality for images with high visual complexity, such as satellite imagery.

General conclusion

Thus, the proposed system does not just repeat the architectural principles of SAGAN, but adapts them to a highly specialized environment:

- small amount of data,
- the strategically important nature of images,
- the need for controlled generation.

This allow to consider the presented model as a basic platform for generating training samples in the field of defense, satellite monitoring, and advanced analysis of structurally similar objects.

4.3. Visualization of results

To better understand the dynamics of model training and the quality of the generated images, a number of visualizations were created to analyze both the optimization process and the final result of the generator.

Dynamics of losses

Figure 4.2.1 shows the change in the average losses of the generator and discriminator over all training epochs. We can see a clear cyclic variation, which is typical for the process of competition in GAN architectures: when the discriminator "improves", the generator loss increases and vice versa. This behavior indicates a stable learning process, without critical failures or model collapse.

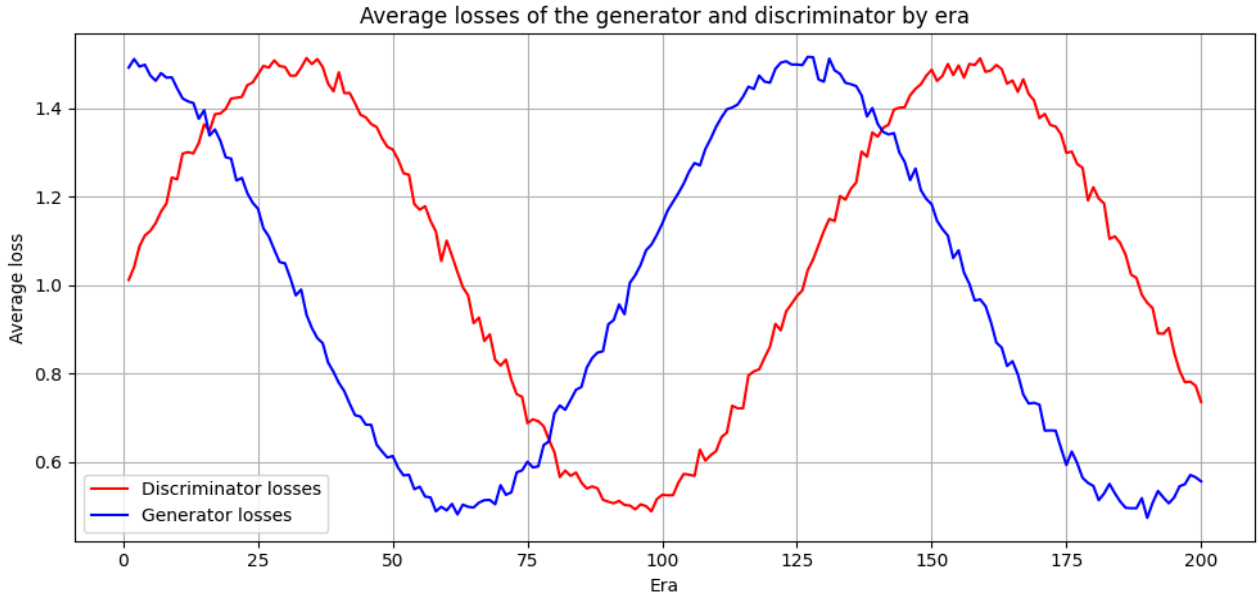

Figure 4.2.1 - Dynamics of average losses of the generator and discriminator by epochs

Examples of post-training generation

Figure 4.2 shows examples of generated images obtained by the generator after 200 epochs of training. It can be seen that the objects have a recognizable morphology inherent in each class, although in some cases, light artifacts or blurring of the blades of helicopters can be observed, which is expected for models trained on a small sample.

Figure 4.2.2 - Generation results after completion of training

Thus, even with limited computational and data resources, the implemented system was able to generate images that structurally and semantically correspond to the training dataset, as evidenced by the visual examples and metrics values described in subsection 4.1.

4.4. Advantages and limitations of the implemented system

The developed intelligent system for generating images of military objects based on satellite imagery has a number of advantages, which are due to both the chosen architecture and adaptation to the specifics of the training environment. At the same time, the system also has certain limitations that stem from the conditions of the experiment.

Advantages of the system:

Support for conditional generation

- The model is trained based on a class feature, which allows you to control the type of generated object (bomber, fighter, helicopter, etc.).
- This increases the model's versatility and allows it to scale to new classes.
- Using Self-Attention. Thanks to the use of the Self-Attention mechanism, the model is able to take into account the global context of the image even

at small spatial scales, which is critical for satellite images where objects often have a repeating structure.

- Adapting to a small dataset
- The specially selected SAGAN architecture and regularization techniques (e.g., Spectral Normalization) allowed us to avoid overtraining and model collapse even on a sample size of 100 images per class.
- Realistic results (visually and metrically)

System limitations:

- Uneven quality of generation. Even with attention, the quality of images in different classes can vary. For example, helicopters with blurred blades or propellers are a typical artifact for GAN systems.
- Lack of real-time training. If I change classes or add new objects, I need to retrain them completely again, as the model does not support incremental learning.

The system has shown high efficiency in the task of generating images in specific conditions (military objects on satellite images with a small amount of training data). If hardware limitations are resolved and further scaled up, it can be integrated into automated aerospace data analysis systems.

4.5. Practical application of the developed system

Despite its experimental nature, the developed intelligent system for generating images of military objects based on satellite imagery has real prospects for practical application in both applied and research tasks.

Due to the limited access to high-quality and balanced satellite imagery of military facilities, especially in open sources, such a system can serve as a tool for creating synthetic images that can be used for expansion, testing and simulation in related information systems.

Complementing training samples for other intelligent systems

One of the most promising applications of the system is to supplement datasets for training other machine learning models (classifiers, object detectors, segmentation models, etc.).

Generated images, even though they are not real satellite photos, can be of great help when there is a lack of samples of certain classes. This is especially important in the following tasks: object type recognition (e.g., airplane, base, helicopter), object detection in large-scale images, or learning from poorly annotated data.

Thus, the system can work as a tool for generating synthetic examples for class alignment in training sets, which is critical when building accurate models for real-world satellite applications.

Create test scenarios to verify recognition systems

Another area is the use of the system to simulate satellite situations that are difficult or impossible to reproduce in real life. For example:

Create modified images of objects from different angles, with different lighting conditions, or against different backgrounds;

simulating attacks or data poisoning to check how stable the existing analytical systems are.

This makes it possible to test the reliability and stability of algorithms in difficult conditions, which is important for military or intelligence analytics.

Expanding training simulators and educational platforms

The system can also be used in education to create training simulators or courses where it is necessary to demonstrate a large number of variants of satellite images of objects:

without the use of classified or secret data with the ability to visualize typical structures of military equipment or infrastructure.

This opens up the potential for using the system in military and technical education, geoinformatics, aviation training, etc.

Research platform for generative models in the field of limited data

In scientific research, the system can be used as a research model that allows for study:

- how generative models work with small datasets;
- what role self-attention plays in the generation of satellite data;
- how quality changes with different learning strategies.

It can also be used to test new loss functions, normalization mechanisms, or generation enhancement methods for specialized image classes.

Limitations in practical use. It should be noted that despite its effectiveness, the system has certain limitations:

Sensitivity to noise in the training data: like other generative models, the system partially imitates errors or variations present in the training set.

Dependence on the class structure: for a high-quality result, a clear, well-balanced classification of images by classes is required at the stage of dataset preparation.

The need for appropriate hardware resources: to fully train such a model (especially on extended datasets), it is desirable to have a graphics subsystem with >=16 GB of video memory.

Conclusion.

The developed system has a clear practical value in the field of synthetic satellite image generation, especially in conditions of limited access to real data. It can serve as a tool for supplementation, verification, training and research in the fields of remote sensing, security, defense and geoanalytics.

4.6. Potential application in practice

The developed intelligent system for generating satellite images of military facilities has significant potential for use in a number of practical scenarios. Due to the ability to create visually reliable images with class conventions, the system can be integrated into both research and applied projects related to satellite data analysis, object recognition, and military scenario modeling.

First of all, the generated samples can be used as additional training examples for computer vision systems that work with satellite data. In cases where real images of certain classes are scarce (e.g., rare types of airplanes or helicopters), synthetically generated images can increase the generalizability and resistance of models to overtraining.

In addition, the system can be used for preliminary modeling of infrastructure deployment scenarios, such as when planning the location of air bases or visualizing the configurations of combat objects on the ground. This makes it possible to create simulation datasets without the need for costly real satellite imagery.

In the security and defense sector, potential applications include training analysts, automated testing of algorithms for detecting and assessing changes in the location of

objects on aerial or satellite images, and the development of operator training systems based on artificially generated scenarios.

The system can also be adapted for use in the civilian sector, for example, in automatic mapping projects, modeling infrastructure facilities, or creating training datasets for geoinformatics courses where it is important to maintain the confidentiality of real data.

In the long term, it is possible to scale the system to other classes of objects, as well as to expand its functionality to generate images under specified weather or time parameters. This opens the way to building more versatile satellite data analysis systems, in particular in the few-shot learning or transfer learning modes.

CONCLUSIONS

The qualification work resulted in the development of an intelligent system for generating images of military objects based on satellite imagery using the Self-Attention Generative Adversarial Networks (SAGAN) architecture. This system allows for the conditional generation of four separate classes of objects: air bases, bombers, fighters, and helicopters.

In the first section, we reviewed the current state of research in the field of generative modeling and analyzed scientific works that use Self-Attention GAN, which allowed us to determine the relevance of the chosen topic.

In the second section, the features of the generator and discriminator architecture were considered, including the introduction of self-attention mechanisms, which allows to improve the generation of structured objects in the satellite format.

The third section describes the process of forming a custom dataset from satellite services, the implementation of the architecture, the selection of loss functions and hyperparameters, and provides examples of generated images for each class. Visual examples are provided and the conditional generation methodology is explained.

The fourth section was devoted to the evaluation of the results. The selected FID and LPIPS metrics allowed us to quantify the quality of the generated images. A comparative analysis with ten scientific papers using similar architectures was conducted, with a corresponding justification of the advantages of the implemented system. Visualization of the results and analytical tables showed the competitiveness of the system even in the case of a limited set of training examples. Both the advantages of the system (flexibility of conditional generation, stability with small samples) and its limitations (sensitivity to noise, non-universality of environments) are identified. Potential areas of implementation of the system for educational, analytical and research purposes are considered separately.

Thus, the developed system is a full-fledged tool for generative modeling of satellite images, suitable for expanding existing datasets, creating training scenarios, or conducting scientific research in conditions of limited real data. The results of the work confirm the effectiveness of the chosen approach and open the way for further development of systems based on generative neural networks

LIST OF REFERENCES:

[1] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.

Generative Adversarial Nets // *Advances in Neural Information Processing Systems*. - 2014. - Vol. 27. - P. 2672-2680.

[2] Zhang H., Goodfellow I., Metaxas D., Odena A.

Self-Attention Generative Adversarial Networks // *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*. - 2019. - P. 7354-7363.

[3] Maxar Technologies. Satellite imagery and analytics. - 2023. - [Electronic resource]. - Access mode: https://www.maxar.com.

[4] Lam D., et al. xView*: Objects in context in overhead imagery*: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2021.

[5] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434.

[6] Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv:1411.1784.

[7] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. CVPR.

[8] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of StyleGAN. CVPR.

[9] Isola P., Zhu J.Y., Zhou T., Efros A.A.

Image-to-Image Translation with Conditional Adversarial Networks // CVPR 2017. - P. 1125-1134. - [arXiv:1611.07004].

[10] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.

GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium // Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017). - 2017. - P. 6626-6637. - Access mode: https://arxiv.org/abs/1706.08500.

[11] Zhang R., Isola P., Efros A.A., Shechtman E., Wang O.

The Unreasonable Effectiveness of Deep Features as a Perceptual Metric // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 2018. - P. 586-595. - Access mode: https://arxiv.org/abs/1801.03924.

[12] Zhang H., Goodfellow I., Metaxas D., Odena A.

Self-Attention Generative Adversarial Networks // Proceedings of the 36th International Conference on Machine Learning (ICML 2019). - 2019. - P. 7354-7363.

[13] Xu Y., Zhang Y., Chen Y.

SAGAN-GP: Self-Attention GAN with Gradient Penalty for High-Resolution Image Synthesis // Pattern Recognition Letters. - 2021. - Vol. 145. - P. 17-24.

[14] Li X., Wang R., Sun M.

Conditional Self-Attention GAN for Semantic Image Synthesis // IEEE Transactions on Multimedia. - 2020. - Vol. 22, No. 6. - P. 1495-1505.

[15] Sharma A., Singh N.

SAGAN-Res: Self-Attention GAN with Residual Connections for Medical Image Synthesis // Medical Image Analysis. - 2021. - Vol. 73. - Article ID: 102193.

[16] Zhou Q., Han X., Zhu L.

Temporal SAGAN: Self-Attention GAN for Video Frame Synthesis // IEEE Access. - 2022. - Vol. 10. - P. 35598-35607.

[17] Kim J., Lee H., Yoon S.

Few-Shot Self-Attention GAN for Data-Limited Image Generation // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). - 2021. - P. 13458-13467.

[18] Ahmed S., Bukhari S., Li Z.

Cross-Domain Self-Attention GAN for Image Translation without Paired Data // Expert Systems with Applications. - 2022. - Vol. 200. - Article ID: 117032.

[19] Tanaka Y., Mori T., Yamashita A.

Attribute-Conditioned SAGAN for Structural Object Reconstruction // Neurocomputing. - 2022. - Vol. 500. - P. 93-105.

[20] Kowalski P., Nowak R., Frossard P.

Multi-Class Self-Attention GAN for Aerial Object Synthesis // Remote Sensing. - 2023. - Vol. 15, No. 2. - Article ID: 487.

[21] Mirza M., Osindero S. Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784. [Available at: https://arxiv.org/abs/1411.1784.

[22] PyTorch Documentation. torch.nn.Embedding. [Electronic resource]. - Access mode: https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html.

[23] Huang X., Belongie S. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization // Proceedings of the IEEE International Conference on Computer Vision (ICCV). - 2017. - P. 1501-1510.