MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Aeronautics, Electronics and Telecommunications, Department of Aviation Computer-Integrated Complexes

ACCEPT TO PROTECTION
Head of Department
________ Viktor SINEGLAZOV
"___"_______________2023 р.

QUALIFICATION PAPER (EXPLANATORY NOTE)

HIGHER EDUCATION STUDY

"MASTER"

Specialty 151 "Automation and computer-integrated technologies"
Educational and professional program "Information support and engineering of aviation computer systems"

Subject: A system for generating training samples in semi-supervised learning tasks based on generative-competitive networks

Performer: student of the group ІЗ-225М Trotsiuk D.O.
Supervisor: Head of the AKIK department, doctor of technical sciences, professor

Sineglazov V.M.

Consultant of the "Environmental Protection» _______ Radomska M.M.
(підпис)

Consultant of the "Occupational safety and health» ___________ Kazhan К.І.
(підпис)

Norm control: _________ Fylashkin M.K.
(підпис)

Kyiv – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп’ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
________ Віктор СИНЄГЛАЗОВ
“___”_______________2023 р.

КВАЛІФІКАЦІЙНА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТР”

Спеціальність 151 «Автоматизація та компю’терно-інтегровані технології»
Освітньо-професійна програма «Комп’ютерно-інтегровані технологічні процеси і виробництва»

Тема: Система генерування навчальної вибірки в задачах напівкерованого навчання на основі генеративно- змагальних мереж

Виконавець: студент групи ІЗ-225М Троцюк Денис Олександрович

Керівник: Завідувач кафедри АКІК, д.т.н., професор Синєглазов Віктор Михайлович

Консультант розділу «Охорона навколишнього середовища» _______ Радомська М.М.
(підпис)

Консультант розділу «Охорона праці» ___________ Кажан К.І.
(підпис)

Нормоконтролер: _________ Філяшкін М.К.
(підпис)

Київ – 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій

Кафедра авіаційних комп'ютерно-інтегрованих комплексів

Освітній ступінь: Магістр

Спеціальність: 151 " Автоматизація та комп'ютерно-інтегровані технології"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віктор СИНЄГЛАЗОВ

" ____ " __________2023 р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Троцюка Дениса Олександровича

1. **Тема проекту (роботи):** " Система генерування навчальної вибірки в задачах напівкерованого навчання на основі генеративно- змагальних мереж "
2. **Термін виконання проекту (роботи):** з 02.10.2023 р. до 27.12.2023 р.
3. **Вихідні данні до проекту (роботи):** Тип мережі: генеративна змагальна. Склад мережі: генератор та дискримінатор. Подолання колапсу моди. Гіперпараметри.
4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):**
   1. Перспективи штучного інтелекту;
   2. Змагальні генеративні мережі;
   3. Структурний та параметричний синтез WGAN-GP;
   4. Розробка системи генерації зображень. аналіз модифікацій генеративної моделі GAN;
   5. Оцінка життєвого циклу обладнання;
   6. Охорона праці.
5. **Перелік обов'язкового графічного матеріалу:** 1. Схема нейрона;

   2. Логістична крива; 3. Класифікація нейронних мереж; 4. Ілюстрація важливості програшу суперника; 5. Архітектура генератора; 6. Нормалізація за батчами.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Отримання завдання	02.10.2023 – 03.10.2023	
2.	Формування мети та основних завдань дослідження	03.10.2023 – 05.10.2023	
3.	Аналіз існуючих рішень	07.10.2023 – 15.10.2023	
4.	Теоретичний розгляд рішення задачі	17.10.2023 – 01.11.2023	
5.	Структурний та параметричний синтез WGAN-GP	01.11.2023 – 15.11.2023	
6.	Розробка системи генерації зображень.	20.11.2023 – 05.12.2023	
7.	Оформлення пояснювальної записки	07.12.2023 – 10.12.2023	
8.	Підготовка презентації та роздаткового матеріалу	12.12.2023– 17.12.2023	

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	Доцент Катерина КАЖАН		
Охорона навколишнього середовища	Доцент Маргарита РАДОМСЬКА		

7. Дата видачі завдання: “2” жовтня 2023 р.

Керівник дипломної роботи ______________ Віктор СИНЄГЛАЗОВ

Завдання прийняв до виконання _________________ Денис ТРОЦЮК

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи «Система генерування навчальної вибірки в задачах напівкерованого навчання на основі генеративно-змагальних мереж» 76 с., 27 рис., 8 табл, 9 джерел.

ГЕНЕРАТИВНІ ЗМАГАЛЬНІ НЕЙРОННІ МЕРЕЖІ, НЕЗБАЛАНСОВАНІ ДАНІ, ОБ'ЄКТ ВИЯВЛЕННЯ, СЕГМЕНТАЦІЯ, КЛАСИФІКАЦІЯ, ГЛИБОКЕ НАВЧАННЯ, ГЛИБОКА ГЕНЕРАТИВНА МОДЕЛЬ

Об'єкт дослідження - генеративно змагальні мережі.

Предмет дослідження - методи їх структурно-параметричного синтезу

Мета кваліфікаційної роботи - розробка методів структурно-параметричного синтезу для оптимізації ГЗМ, що сприятиме покращенню точності та диверсифікації генерованих даних.

Метод дослідження - аналіз наукової літератури, комп'ютерне моделювання, експериментальні методи, статистична обробка даних.

В результаті роботи очікується отримання моделі ГЗМ з покращеними характеристиками, що дозволить розширити можливості їх практичного використання у різних областях, таких як візуальне мистецтво, автоматизація проектування, розробка ігор, та інших сферах, де необхідно генерувати високоякісний контент.

NATIONAL AVIATION UNIVERSITY

Faculty of aeronavigation, electronics and telecommunications

Department of Aviation Computer Integrated Complexes

Educational level: Master

Specialty: 151 "Automation and computer-integrated technologies"

APPROVED

Head of Department

Viktor SINEGLAZOV

"____" __________2023

TASK
For the student's thesis

Trotsiuk Denys Oleksandrovych

1. **Theme of the project:** "A system for generating training samples in semi-supervised learning tasks based on generative-competitive networks”
2. **The term of the project (work):** from October 02, 2023 until December 27, 2023
3. **Output data to the project (work):** Type of network: generative competitive. Network composition: generator and discriminator. Overcoming the collapse of mode. Hyperparameters.
4. **Contents of the explanatory note (list of questions to be developed):**
   1. Prospects of artificial intelligence;
   2. Competitive generative networks;
   3. Structural and parametric synthesis of WGAN-GP;
   4. Development of an image generation system. analysis of modifications to the generative model of GAN;
   5. Evaluation of the life cycle of the equipment;
   6. Labor protection.
5. **List of compulsory graphic material:** 1. Scheme of the neuron;

   2. Logistic curve; 3. Classification of neural networks; 4. Illustration of the importance of losing an opponent; 5. Architecture of the generator; 6. Normalization by batches.

6. Planned schedule:

№	**Task**	**Execution term**	**Execution mark**
1.	Task	02.10.2023 – 03.10.2023	
2.	Purpose formation and describing the main research tasks	03.10.2023 – 05.10.2023	
3.	Analysis of existing solutions	07.10.2023 – 15.10.2023	
4.	Analysis of existing systems	17.10.2023 – 01.11.2023	
5.	Structural and parametric synthesis of WGAN-GP	01.11.2023 – 15.11.2023	
6.	Development of an image generation system.	20.11.2023 – 05.12.2023	
7.	Making an explanatory note	07.12.2023 – 10.12.2023	
8.	Preparation of presentation and handouts	12.12.2023– 17.12.2023	

6. Consultants from individual sections

Section	Consultant	Date, signature	
		Issued the task	Accepted the task
Occupational safety and health	Associate Professor Kateryna KAZHAN		
Environmental protection	Associate Professor Margarita RADOMSKA		

7. **Date of task receiving:** “2” October 2023

Diploma thesis supervisor Viktor SINEGLAZOV

(signature)

Issued task accepted Denys TROTSIUK

(signature)

ABSTRACT

Explanatory note of qualification work "System of training sample generation in semi-supervised learning tasks based on generative adversarial networks" 76 p., 27 figs., 8 tables, 9 references.

GENERATIVE ADVERSARIAL NEURAL NETWORKS, UNBALANCED DATA, OBJECT OF DETECTION, SEGMENTATION, CLASSIFICATION, DEEP LEARNING, DEEP GENERATIVE MODEL

The object of research is generative adversarial networks.

Subject of research - methods of their structural and parametric synthesis

The purpose of the qualification work is to develop methods of structural and parametric synthesis for optimizing GANs, which will improve the accuracy and diversification of the generated data.

Research methods - analysis of scientific literature, computer modeling, experimental methods, statistical data processing.

As a result of the work, it is expected to obtain a model of GANs with improved characteristics, which will expand the possibilities of their practical use in various fields, such as visual art, design automation, game development, and other areas where it is necessary to generate high-quality content.

CONTENT

INTRODUCTION

In today's world of rapid technological development, artificial intelligence (AI) is taking a special place, being implemented in various areas of human activity, ranging from industrial automation to creative professions. Among the areas of AI research, generative adversarial networks (GANs) have proven to be a promising tool for generating new data that mimics real-world patterns. GANs play a key role in the development of machine learning, in particular, in image synthesis, natural language processing, music creation, and other tasks.

The relevance of the research topic is due to the rapid development of neural networks and the need to create efficient and reliable systems that can adapt to changing production conditions and market needs.

Despite the large number of studies in this area, the structural-parametric synthesis of ANNs still requires in-depth study to optimize the network training process and improve the quality of the generated data.

The purpose of this master's thesis is to develop structural-parametric synthesis methods for optimizing GANs, which will improve the accuracy and diversification of the generated data. As part of the work, the author of the master's thesis investigated the following aspects: analysis of existing GAN architectures, development of new methods of structural synthesis and parametric optimization, experimental verification of the developed methods on real data.

The object of research is generatively adversarial networks, and the subject is methods of their structural and parametric synthesis. In the course of the work, the author used such research methods as analysis of scientific literature, computer modeling, experimental methods, and statistical data processing.

As a result of the work, it is expected to obtain a model of GAN with improved characteristics, which will expand the possibilities of their practical use in various fields, such as visual art, design automation, game development, and other areas where it is necessary to generate high-quality content.

CHAPTER 1
PERSPECTIVES OF ARTIFICIAL INTELLIGENCE

1.1 AREAS OF USE

Artificial intelligence[1] (AI) is intelligence displayed by machines, as opposed to natural intelligence displayed by animals, including humans. Leading AI textbooks define this area as the study of "intelligent agents": any system that perceives its environment and takes actions that increase its chances of achieving its goals. Several popular sources use the term "artificial intelligence" to describe machines that mimic the "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", but this definition is rejected by leading AI researchers.

AI applications include advanced search engines (like Google), recommendation engines (like YouTube, Amazon and Netflix), human speech comprehension (like Siri and Alexa), self-driving cars (like Tesla), automated decision making. and high-level competition in strategic gaming systems (such as chess and go). As machines become more capable, tasks thought to require "intelligence" are often excluded from the definition of AI, a phenomenon known as the AI effect. For example, optical character recognition is often excluded from things that are considered artificial intelligence as it has become a common technology.

Artificial intelligence was founded as an academic discipline in 1956 and has since experienced several waves of optimism, followed by disappointment and loss of funding (known as the "artificial intelligence winter"), which were followed

by new approaches, success and renewed funding. Since its inception, AI research has tried and abandoned many different approaches, including brain modeling, human problem solution modeling, formal logic, large databases of knowledge, and animal behavior simulations[2]. In the early decades of the 21st century, this field was dominated by mathematical statistical machine learning, and this method has proven to be very successful in helping to solve many complex problems in industry and academia.

The various sub-areas of AI research are centered around specific goals and the use of specific tools. Traditional goals of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception, and the ability to move and manipulate objects. General intelligence[1] (the ability to solve arbitrary problems) is one of the long lines of this field - urgent goals. To address these problems, AI researchers have adapted and integrated a wide range of problem-solving techniques, including search and mathematical optimization, formal logic, artificial neural networks, and methods based on statistics, probability and economics. AI also uses computer science, psychology, linguistics, philosophy, and many other fields.

This area was based on the assumption that human intelligence "can be so accurately described that you can create a machine to simulate it." [3] This raises a philosophical argument about the reason and ethics of creating artificial beings endowed with human-like intelligence. These questions have been explored by myths, fiction and philosophy since ancient times. Science fiction and futurology also suggest that with its immense potential and power, AI could pose a threat to humanity's survival.

Today Artificial Intelligence is used in lot areas:

- Internet and e-commerce
- Education
- Finance
- Health
- Manufacturing
- Media
- And many others

Internet and e-commerce

AI is used to target advertisements to the people most likely to click on them. It is also used to encourage people to stay “engaged” on the website by choosing the content that users are most likely to click on. It can predict or summarize customer behavior[4] from their digital footprints in order to target them for personalized promotions or automatically create customer identities.

As social media sites overtake television as a source of news for young people, news organizations increasingly rely on social media platforms to disseminate information.

Boomtrain[1] is another example of AI that is designed to learn how to best interest each individual reader with specific articles posted on the right channel at the right time that will be most relevant to the reader. It's like hiring a personal editor for each individual reader to create the perfect reading experience.

IRIS.TV[5] helps media companies offer an AI-powered video personalization and programming platform. This allows publishers and content owners to offer contextually relevant content to audiences based on consumer viewing patterns.

A documented case reports that online gambling companies have used AI to improve customer targeting. Moreover, the use of personality-computed AI models can help reduce the cost of ad campaigns by adding psychological targeting to more traditional socio-demographic or behavioral targeting. British firm Ubamarket has developed an app that allows users to shop from home using a smartphone. The app will allow users to pay by phone, list and scan food ingredients for allergens. The app is built on an artificial intelligence module and learns from user behavior to empower and offer personalized suggestions.

Education

AI tutors can enable students to receive additional personalized assistance. They can also reduce anxiety and stress in some students, which can be caused by tutoring labs or human tutors. In future classrooms, environmental informatics can play a useful role. Environmental informatics[2] is the idea that information is everywhere in the environment and that technology automatically adjusts to your personal preferences. Teaching devices can create lessons, tasks and games to adapt to the needs of a particular learner and provide immediate feedback.

But AI can also create an unfavorable environment with a revenge effect if technology prevents society from moving forward and causes negative, unintended consequences for society. An example of the revenge effect is that increased use of technology can make it difficult for students to focus and focus on a task instead of helping them learn and grow. In addition, AI is known to result in the loss of both human activity and simultaneity.

Finance

Financial institutions[6] have long used artificial neural network systems to detect allegations or claims that are outside the norm and flag them for human investigation. The use of AI in banking can be traced back to 1987, when the Security Pacific National Bank in the United States created a fraud prevention task force to counter the unauthorized use of debit cards. Programs such as Kasisto and Moneystream[6] use AI in financial services.

Banks today use artificial intelligence systems to organize transactions, maintain accounting records, invest in stocks, and manage real estate. AI can react to changes in the blink of an eye or when no business is running. In August 2001, robots defeated humans in a simulated financial trading competition. AI has also reduced the incidence of fraud and financial crime by monitoring user behavior patterns for any abnormal changes or anomalies.

AI is increasingly being used by corporations. Jack Ma[6] controversially predicted that the AI CEO is 30 years away.

The use of artificial intelligence machines in the marketplace in applications such as online trading and decision making has changed mainstream economic theories. For example, AI buying and selling platforms have changed the law of supply and demand in the sense that individual supply and demand curves can now be easily assessed and thus customized pricing. In addition, AI machines reduce information asymmetries in the market and thus make markets more efficient while reducing trading volume. In addition, AI in markets limits the effects of market behavior, again making markets more efficient. Other theories influenced by AI include rational choice, rational expectations, game theory, the Lewis turning point, portfolio optimization[3], and counterfactual thinking. In August 2019, the AICPA[6] presented an Artificial Intelligence training course for accounting professionals.

Health

AI in healthcare is often used for classification, whether it is to automate the initial assessment of computed tomography or EKG, or to identify patients at high risk to public health. The range of applications is expanding rapidly. For example, AI is being used to solve an expensive dosing problem, which has shown that AI can save $ 16 billion. In 2016, a groundbreaking study in California showed that a mathematical formula developed using artificial intelligence correctly determines the exact dose of immunosuppressants to be given to patients with internal organs.

Artificial intelligence helps doctors. According to Bloomberg Technology[6], Microsoft has developed AI to help doctors find the right cancer treatments. There is a lot of research and development on cancer drugs. In particular, there are over 800 drugs and vaccines for cancer treatment. This negatively affects doctors because there are too many choices, making it difficult for patients to choose the right drugs. Microsoft is working on a project to develop a machine called the Hanover. Its purpose is to memorize all the documents needed to treat cancer and help predict which drug combinations will be most effective for each patient. One project that is currently being worked on is tackling myeloid leukemia[7], a deadly cancer whose treatment has not improved for decades. It was reported that another study found that artificial intelligence was as good as trained doctors in detecting skin cancer. Another study uses artificial intelligence to try to control multiple high-risk patients by asking each patient a variety of questions based on input from a living doctor in patient interactions. One study was conducted with transfer learning, the machine made a diagnosis in the same way as a well-trained ophthalmologist, and could decide within 30 seconds whether a patient should be referred for treatment with more than 95% accuracy.

According to CNN[8], a recent study by surgeons at the Children's National Medical Center in Washington has successfully demonstrated surgery using an autonomous robot. The team watched the robot as it performed soft tissue surgery, stitching a pig intestine in open surgery, and the group claims it did better than a human surgeon. IBM created its own artificial intelligence computer, the IBM Watson[5], which surpassed human intelligence (on some levels). Watson has struggled to achieve success and expansion in healthcare.

Artificial neural networks are used as clinical decision support systems for medical diagnostics, for example, in concept processing technology in EMR[6] software.

Other medical tasks that could potentially be accomplished with artificial intelligence and that are beginning to be developed include:

- Computer interpretation of medical images. Such systems help scan digital images, such as computed tomography, for a typical appearance and highlight prominent areas such as possible diseases. Typical applications are tumor detection.
- Heart tone analysis
- Aged Care Companion Robots
- Collecting medical records for more useful information
- Develop treatment plans
- Help with repetitive work, including taking medication
- Help for the Blind
- We provide consultations
- Drug creation
- Using avatars[2] instead of patients for clinical training
- Predict the likelihood of death from surgery.
- Predict HIV progression

AI can increase the volume of work tasks, when an employee can be removed from a situation that carries hazards such as stress, overwork, and musculoskeletal injuries, forcing the AI to perform tasks instead. This could broaden the range of employment sectors affected beyond traditional automation to employee jobs and services such as medicine, finance and information technology. For example, call center employees face serious health and safety risks due to its repetitive and demanding nature and high level of micro-surveillance. AI chatbots reduce the need for people to perform the most basic call center tasks.

Machine learning, used in HR analytics to predict employee behavior, can be used to improve employee health. For example, mood analysis can be used to detect fatigue and prevent overwork. Decision support systems have a similar ability, for example, to prevent industrial disasters or improve the efficiency of response to them. Manual material handling workers can use predictive analytics and artificial intelligence to reduce musculoskeletal injuries. Wearable sensors may also enable earlier intervention in the management of toxic exposure, and the resulting large datasets can improve workplace health surveillance, risk assessment and research.

AI can also be used to improve the efficiency of the health and safety workflow. One example is the coding of workers' compensation claims. AI-enabled virtual reality systems can be useful for safety training in hazard recognition. Artificial intelligence can be used to more effectively detect potential accidents that are important to reduce accidents, but which are often unreported.

Manufacturing

Robots[2] have become commonplace in many industries and are often assigned jobs that are considered dangerous to humans. Robots have proven to be effective in repetitive tasks that can lead to mistakes or accidents due to loss of concentration, and in other tasks that humans may find demeaning.

In 2014, China, Japan, USA, Republic of Korea and Germany together accounted for 70% of total robot sales. In the automotive industry, a sector with a particularly high degree of automation, Japan has the highest density of industrial robots in the world: 1,414 per 10,000 employees.

Artificial intelligence has been combined with many sensor technologies such as digital spectrometry from IdeaCuria Inc[5]. which allows it to be used in many applications, for example, for monitoring water quality at home.

In the 1990s, some of the first attempts were made to mass-produce domestically oriented types of basic artificial intelligence for learning or leisure. It made significant progress with the digital revolution and helped introduce people, especially children, to life associated with various types of artificial intelligence, in particular in the form of Tamagotchis[8] and Giga Pets, the iPod Touch, the Internet, and the first widespread robot, Furby. Just a year later, an improved type of home robot was released in the form of the Aibo, a robot dog with intelligent functions and autonomy.

Companies like Mattel are creating a range of artificial intelligence toys for kids as young as three. Using proprietary artificial intelligence engines and speech recognition tools, they can understand conversations, give intelligent answers, and learn quickly.

AI has also been applied to video games, such as video game bots, which are designed to act as adversaries where people are not available or desired.

Media

Some AI applications are designed to analyze audiovisual media content such as movies, TV programs, promotional videos, or user-generated content. Solutions often include computer vision, which is the main application for AI.

Typical use cases include image analysis using object or face recognition techniques, or video analysis to identify relevant scenes, objects, or faces. The motivation for using AI-based media analysis may include facilitating media searches, creating a set of descriptive keywords for a media element, monitoring media content policy (e.g., checking the suitability of content for a particular content), among others. TV Time), speech-to-text conversion for archival or other purposes, and detection of celebrity logos, products or faces for relevant advertisements.

AI media analytics companies often provide their services through a REST API, which allows machine automatic access to technology and enables machine reading of the results. For example, IBM, Microsoft and Amazon provide access to their media recognition technology using RESTful APIs[3].

In June 2016, a research team from the Visual Computing Group at TU Munich and Stanford University developed Face2Face, a program that animates a target's face by transposing facial expressions from an external source. The technology of animating the lips of people, including Barack Obama and Vladimir Putin, was demonstrated. Since then, other methods have been demonstrated, based on a deep neural network, from which the name "deepfake" was taken.

In September 2018, US Senator Mark Warner proposed punishing social media companies that allow deepfake documents to be published on their platform.

Vincent Nozick, a researcher at the Gaspard Monge Institute, found a way to detect falsified documents by analyzing the movements of the century. DARPA (a research group affiliated with the US Department of Defense) has committed $ 68 million for deepfake detection work. In Europe, Horizon 2020 funded InVid, a software program designed to help journalists detect fraudulent documents.

Deepfakes can be used for comedic purposes, but they are better known for being used for fake news and pranks. There are also sound deepfakes and artificial intelligence software capable of detecting deepfakes and cloning human voices after 5 seconds of listening.

While the development of music has always been influenced by technology, artificial intelligence, thanks to scientific advances, has made it possible to imitate human composition to some extent.

Among notable early efforts, David Cope created an AI called Emily Howell, which managed to become well known in the field of algorithmic computer music. Emily Howell's algorithm[2] is registered as a US patent.

AI Iamus created in 2012 the first complete classic album written entirely in computer.

Other directions such as AIVA (Virtual Artist with Artificial Intelligence) focus on composing symphonic music, mainly classical music for film scores. He became the first in the world, becoming the first virtual composer recognized by the music professional association.

Artificial intelligence can even create music suitable for use in medical settings, thanks to Melomics'[4] efforts to use computer music to relieve stress and pain.

What's more, initiatives like Google Magenta, run by the Google Brain team, want to find out if artificial intelligence can create compelling works of art.

At Sony's CSL research lab, their Flow Machines software creates pop songs by learning musical styles from a huge database of songs. By analyzing unique combinations of styles and optimization techniques, he can compose any style.

Another AI music composition project, The Watson Beat, written by IBM Research, doesn't need a huge music database like the Google Magenta and Flow Machines projects as it uses Reinforcement Learning and Deep Belief Networks to compose music on simple input. melody and selected style. Since the software was open source, musicians like Taryn Southern collaborated with the project to create music.

South Korean singer Hayeon's debut song "Eyes on You" was written using artificial intelligence and was then directed by real composers, including NUVO.

Company Narrative Science makes computer news and reports commercially available, including summaries of team sporting events based on game statistics in English.

He also creates financial statements and real estate analysis. Similarly, Automated Insights creates personalized headlines and previews for Yahoo Sports Fantasy Football. The company is projected to create one billion stories in 2014, up from 350 million in 2013. The OpenAI organization has also created artificial intelligence capable of writing text.

Another company called Yseop uses artificial intelligence to transform structured data into intelligent natural language comments and recommendations. Yseop can write financial reports, resumes, personalized sales or marketing documents, and more at thousands of pages per second and in multiple languages including English, Spanish, French, and German.

In addition to automating the writing of data-entry tasks, AI has shown the significant potential of computers to engage in higher-level creative work. AI storytelling has become an active area of research since the development of TALESPIN[4] by James Meehan, which consisted of stories similar to the fables of Aesop. The program will begin with a cast of characters who wanted to achieve certain goals, with the story as a narrative of the characters' attempts to fulfill plans to achieve those goals. similar or different approaches. Mark Ridl and Vadim Bulitko argued that the essence of storytelling is an experience management problem, or "how to balance the need for consistent story development with the user agency, which often contradicts each other."

While most research in artificial intelligence storytelling has focused on storytelling (such as characters and plots), there has also been significant research into story communication. In 2002, researchers at North Carolina State University developed an architectural framework for creating narrative prose. Their concrete implementation was able to faithfully reproduce the textual variety and complexity of a number of stories like Little Red Riding Hood[6] with human dexterity. It is this area that continues to generate interest. In 2016, a Japanese AI co-wrote a story and nearly won a literary prize.

Hanteo Global, the organization that operates the only real-time high score table in South Korea, also employs an automated journalist-bot to write articles.

1.2 NEURAL NETWORK AND ITS FEATURES

Neural network (also known as Artificial neural network, ANN) is a mathematical model and its software or hardware implementation based on the organization and function of neural networks (biological neural networks are others). This idea arose while studying the processes occurring in the brain and trying to model these processes. The first experiment was the neural network of W. McCulloch and W. Pitts. After the learning algorithm is developed, the model is used for the following purposes: prediction problem, model analysis, control problem, etc.

ANN is a simple network and interaction network (artificial neurons). Such procedures are usually very simple (especially compared to those used on personal computers). Each processor of such a network processes only the signals it receives regularly and postpones sending to other processors. However, when connected to the control interface on a large enough network, this simple process can become a very complex task.

• From a machine learning perspective, neural networks are useful for pattern recognition, discriminant analysis, clustering, etc.

• From a mathematical perspective, training neural networks is a non-optimization problem.

• From a cybernetic perspective, neural networks are used in adaptive control problems and robotic algorithms.

• From the perspective of technological development and performance, neural networks

• From the perspective of artificial intelligence, ANN is the basis of network language thinking and learning about construction (modeling) in design. The main aspect of being able to use the natural intelligence of computer algorithms.

Neural networks are not programmed in the conventional sense, but are trained. Learning is one of the main benefits of neural networks over algorithms. Technically, training consists of finding the connection coefficients between neurons. During the training process, neural networks can identify path dependencies between input data and output and detail. This means that if training is successful, the network can return results based on missing data in the training sample, as well as missing data and/or “no noise”, and the document will be partially truncated.

Neuron

An artificial neuron[1] is a mathematical function conceived as a model of biological neurons, a neural network. Artificial neurons are elementary units in an artificial neural network. An artificial neuron receives one or more input signals (representing excitatory postsynaptic potentials and inhibitory postsynaptic potentials on neuronal dendrites) and summarizes them to produce an output signal (or activation, representing the action potential of the neuron, which is transmitted along its axon). Typically, each input is weighted separately and the sum is transferred through a non-linear function known as an activation function or transfer function. Transfer functions are usually sigmoid in shape, but they can also take the form of other nonlinear functions, piecewise linear functions, or step functions. They are also often monotonically increasing, continuous, differentiable, and bounded. Also recently investigated were non-monotonic, unbounded and oscillating activation functions with multiple zeros that outperform sigmoidal and ReLU-like activation functions in many applications. Oscillating activation functions can improve gradient flow in neural networks and allow the study of functions with fewer neurons. For example, a bipolar encoded XOR function can be learned with a single GCU neuron. The threshold function inspired the creation of logic gates called threshold logic; useful for constructing logic circuits that resemble brain processing. For example, recently new devices such as memristors have been widely used to develop such logic.

The transfer function[1] of an artificial neuron should not be confused with the transfer function of a linear system.

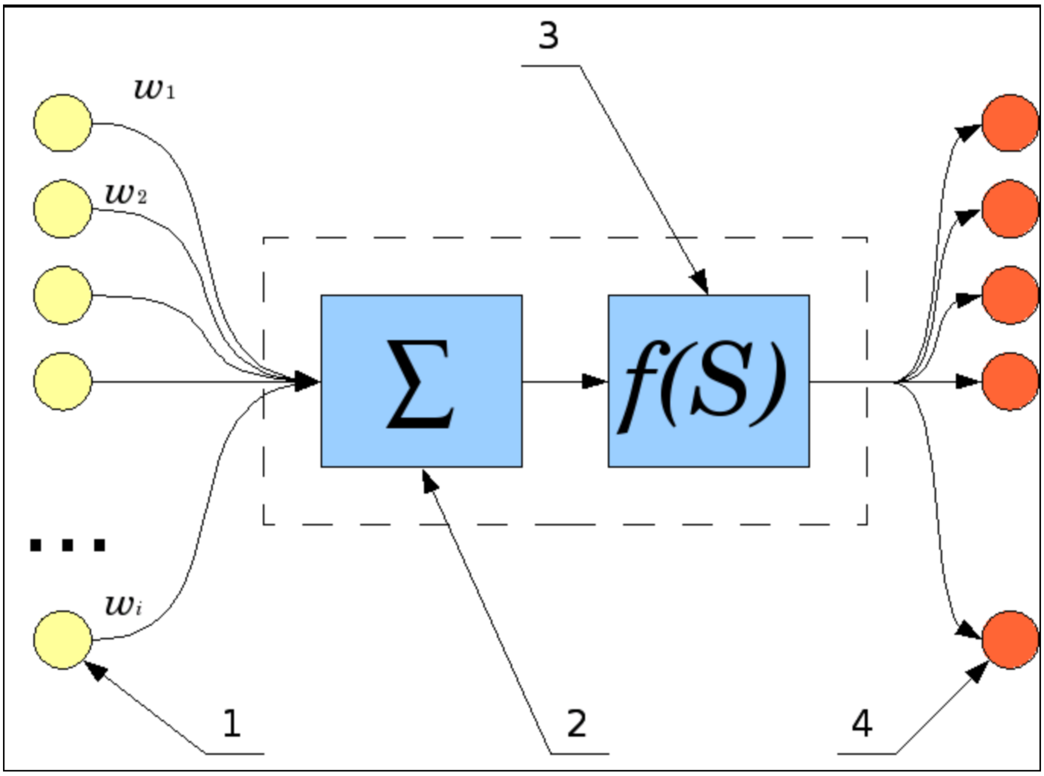


Fig 1.1. Scheme of a neuron

1 – Input signal

2 – Adder

3 – Transfer function

4 – Output signal

$\omega_{1..i}$ – Weights

The output of k-th neuron can be described by following:

$$y_k = \varphi(\sum_{i=0}^{n} \omega_{ki} * x_i) \quad (1.1)$$

where φ is the transfer function.

Types of transfer function

The transfer function (activation function) of a neuron is chosen to have a number of properties that either improve or simplify the network containing the neuron. It is imperative, for example, that any multilayer perceptron using a linear transfer function has an equivalent single layer network; therefore, a non-linear function is required to take advantage of the layered network.

Below u in all cases refers to the weighted sum of all inputs of the neuron, that is, for n inputs

$$u = \sum_{i=1}^{n} \omega_i * x_i \qquad (1.2)$$

where w is the vector of synaptic weights and x is the vector of inputs.

Step function

The output y of this transfer function is binary, depending on whether the input meets a predetermined threshold θ. The "signal" is being sent, that means the output is set to one if the activation meets the threshold.

$$y = \begin{cases} 1 & if\ u \geq \theta \\ 0 & if\ u < \theta \end{cases} \qquad (1.3)$$

This function is used in perceptrons[1] and is often found in many other models. It divides the input space by a hyperplane. This is especially useful in the last layer of the network for binary classification of inputs. It can be approximated from other sigmoidal functions by assigning larger values to the weights.

Linear combination

In this case, the unit of output is simply the weighted sum of its inputs plus the bias term. A number of such linear neurons perform a linear transformation of the input vector. This is usually more useful in the early layers of the network. There are several analysis tools based on linear models, such as harmonic analysis, and all of them can be used in neural networks with this linear neuron. The offset member allows us to perform affine transformations on the data.

Sigmoid

A simple non-linear function, a sigmoid function such as a logistic[1] function also has an easily computable derivative, which can be important when computing weight updates on a network. Thus, it simplifies the mathematical manipulation of the network and was attractive to early computer scientists who needed to minimize the computational load in simulations. This was previously commonly seen in multilayer perceptrons. However, recent work has shown that sigmoid neurons are less efficient than rectified linear neurons. The reason is that the gradients computed by the backpropagation algorithm tend to decrease to zero as activations propagate through the layers of sigmoidal neurons, making it difficult to optimize neural networks using multiple layers of sigmoidal neurons. A typical example of a sigmoid function is the logistic function shown in the figure 2 and defined by the formula (4):

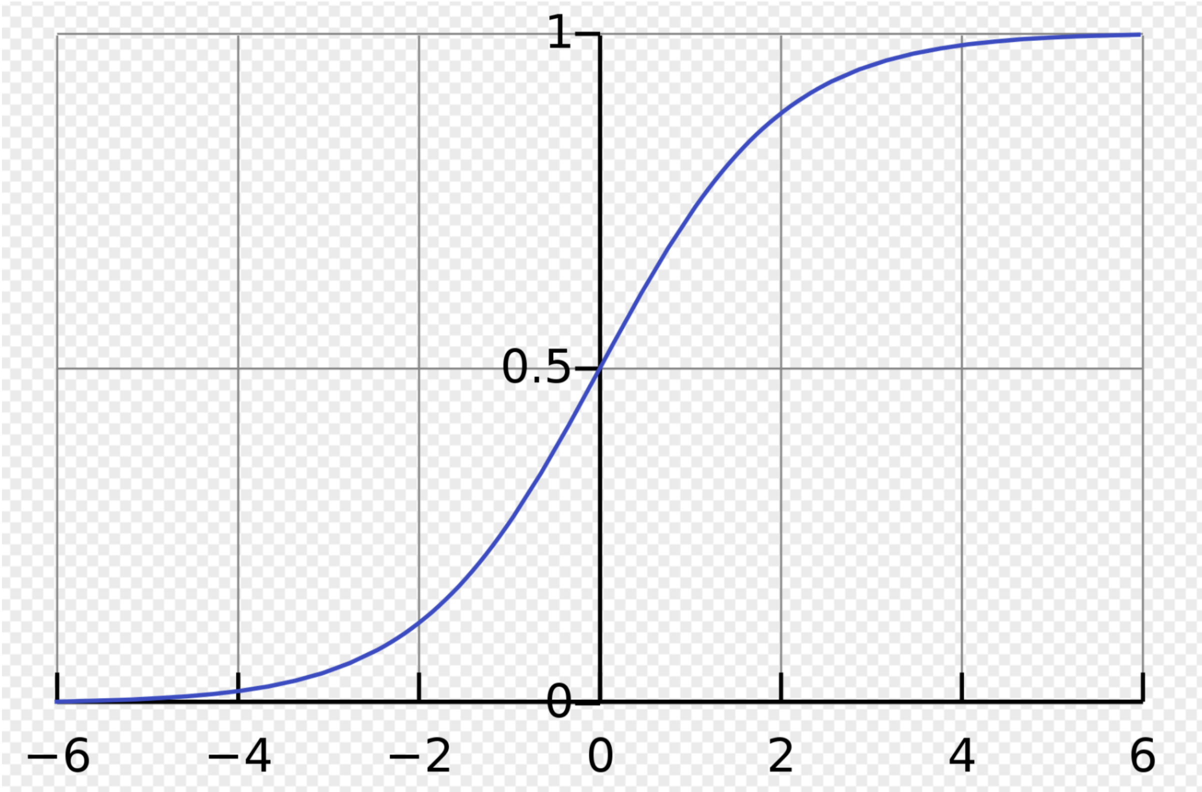


Fig. 1.2 A logistic curve

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \qquad (1.4)$$

Rectifier

In the context of artificial neural networks, a rectifier is an activation function, defined as the positive part of its argument:

$$f(x) = x^+ = \max(0, x) \qquad (1.5)$$

where x is the input of the neuron. It is also known as a ramp function and is similar to half-wave rectification in electrical engineering. This activation function was first introduced into the dynamic network by Hahnloser[4] et al. in a 2000 article in Nature with strong biological motives and mathematical rationale. In 2011, it was first demonstrated that this allows for better training of deeper networks compared to the widely used activation functions before 2011, i.e., the Logistic sigmoid (which is inspired by the theory of probability) and its more practical counterpart, hyperbolic tangent.

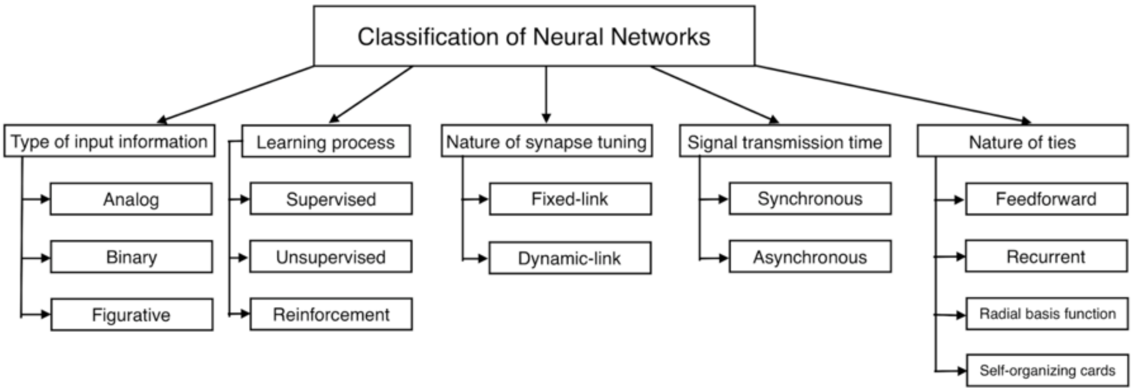


Figure 1.3. Classification of Neural Networks

CHAPTER 2
GENERATIVE ADVERSARIAL NETWORKS

Neural networks (GANs) are designed to learn ground truth classification from a limited number of images (minimum and global classes) and then use the learning data to create synthetic images. This raises an interesting question: whether GANs can be used to create synthetic images for small classes in heterogeneous datasets. In fact, recent advances in GANs show that the ability to represent complex and high-level data can be used as a method for intelligent competition. GANs use the ability of neural networks to learn functions that make the model distribution as close as possible to the real distribution.

They do not rely on prior assumptions, particularly about distribution, and can create synthetic images with high visual fidelity. This important feature allows GANs to be used for all kinds of intractable problems in computer vision. GANs not only create fake images, but also provide a way to replace some elements of the original image. In other words, they can learn to create any number of groups (e.g. objects, individuals, persons, etc.) as well as a variety of variables (e.g. vision, lighting, scale, background, etc.). Many types of GANs have been reported in the literature, and each has its own advantages in solving the challenging problem in computer vision. For example, AttGAN [64], IcGAN [65], ResAttrGAN [66], etc. It is a type of GAN frequently used for face recognition tasks. Not only do they learn to create new facial images with desirable features, they also capture details of unexpected behavior. Recently, GANs have been combined with many existing and explored image segmentation algorithms to overcome the uncertainty problem and improve their performance.

The original GAN architecture [67] has two different roles represented by two networks, namely generator G and discriminator D. The learning process of GAN trains discriminator D and generator G simultaneously. It follows a competitive two-player zero-sum game. An intuitive way to understand GANs is the identification of police officers and fake people. The generator network looks like a group of scammers trying to create fake money and pass it off as real. The police are trying to catch fraudsters using fake money, but they also have to make sure others are using their real money. At the same time, police have seen signs of improvement in detecting counterfeit money, and fraudsters have also made progress in crime. Finally, scammers are forced to create excellent products with real money. High-resolution and true minority images created by learning the classification model can be used to balance class distribution and reduce the effects of overfitting caused by the growth of large datasets. GAN solves the problem of generating data without sufficient data and without the need for human supervision. GANs can provide an effective way to fill gaps in the distribution of training data. In other words, they can transform the distribution of training data into a continuous distribution by providing additional information from the random interaction of random points. Powers et al. [68] argued that GANs provide a way to unlock more information from the dataset. In fact, Facebook VP and Chief AI Researcher Yann LeCun called GANs "the most interesting thing to happen in machine learning in the last 10 years."

To solve the problem in computation, researchers focused on solving the problem of uncertainty in big image data by following a method that will help researchers develop a detailed understanding of GAN-based image generation.

2.1.BRIEF INTRODUCTION TO THE CURRENT STATE OF DEEP GENERATIVE IMAGE MODEL

Adversarial models try to model the distribution of the real data through an adversarial process. Generative adversarial neural networks based on game theory, introduced by Goodfellow et al. [67] in 2014, is arguably one of the best innovations in recent years. The word adversarial in generative adversarial neural networks means that the two neural networks, the generator and the discriminator are in a competition with each other. the learning procedure of GAN is to simultaneously train a discriminator D and a generator G . The generator network takes a noise vector z in a latent space as an input, then runs that noise vector through a differentiable function to transform the noise vector z to create a fake but plausible image $x{:}G(z) \rightarrow x$. At the same time, the discriminator network, which is essentially a binary classifer, tries to distinguish between the real images (label 1) and artificially generated images by generator network (label 0):D(x) → [0, 1]. therefore, the objective function of GANs can be defined as

$$min_G max_D V(D, G) = E_{x\sim p_x(x)}[\log(D\,(x))] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

(2.1)

Given random noise vector z and real image x, the generator attempts to minimize $\log(1 - D(G(z))$ and the discriminator attempts to maximize $\log D(x)$ in Eq. (2.1). For fixed G , the optimal D is given by

$$D*(x) = \frac{p_r(x)}{p_g(x) + p_r(x)}$$

(2.2)

Theoretically, when G is trained to its optimal, the generated data distribution $p_g(x)$ gets closer to the real data distribution $p_r(x)$. If $p_g(x) = p_r(x)$ $D^*(x)$ in Eq. (2.2) becomes 1/z. This means that the discriminator is maximally puzzled and cannot distinguish fake images from real ones. When the discriminator D is optimal, the loss function for the generator G can be visualized by substituting in $D^*(x)$ Eq. (2.1).

$$G^* = max_D V(G, D^*) = E_{x\sim p_r(x)}[\log D^*(x)] + E_{x\sim p_g(x)}\left[\log(1 - D^*(x))\right]$$

$$= E_{x\sim p_r(x)}\left[\log\frac{p_r(x)}{\frac{1}{2}[p_g(x) + p_r(x)]}\right] + E_{x\sim p_g(x)}\left[\log\frac{p_g(x)}{\frac{1}{2}[p_g(x) + p_r(x)]}\right] - 2log2$$

(2.3)

The definition of Jensen-Shannon divergence (D_{JS}) between two probability distributions $p_g(x)$ and $p_r(x)$ is defined as

$$D_{JS}(p_r||p_g) = \frac{1}{2} D_{KL}(p_r|| \frac{p_r + p_g}{2}) + \frac{1}{2} D_{KL}(p_g || \frac{p_r + p_g}{2})$$

(2.4)

Therefore, Eq. (2.3) is equal to

$$G^* = 2D_{JS}(p_r(x)||p_g(x)) - 2log2$$

(2.5)

Essentially, the loss for the generator G minimizes the Jensen-Shannon divergence between the generated data distribution $p_g(x)$ and the real data distribution $p_r(x)$ when discriminator D is optimal. Jensen-Shannon divergence is a smooth, symmetric version of the KL divergence. Huszar [110] believes that the main reason behind the great success of GANs is replacing asymmetric KL divergence loss function in the classical approach to symmetric JS divergence.

Mean squared error used in latent variable models such as autoencoder, averages all the possible features in an image and generate blurry images. In contrast, adversarial loss preserves the features using discriminator networks that detect an absence of any features as an unrealistic image. An example of this is the study carried out by Lotter et al. [111], in which models trained using mean square loss and adversarial loss to predict the next image frame in a video sequence are compared. A model trained using mean square loss generates blurry images as shown in Fig.2.1, where ear and eyes are not sharply defined as they could be. Using an additional adversarial loss, features like the eyes and ear remain preserved very well, because an ear is the recognizable pattern, and the discriminator network would not accept any sample that is missing an ear.

This section has attempted to provide readers a brief introduction to the current state of deep generative image models. A quick summary of this section is depicted below in Fig.2.2

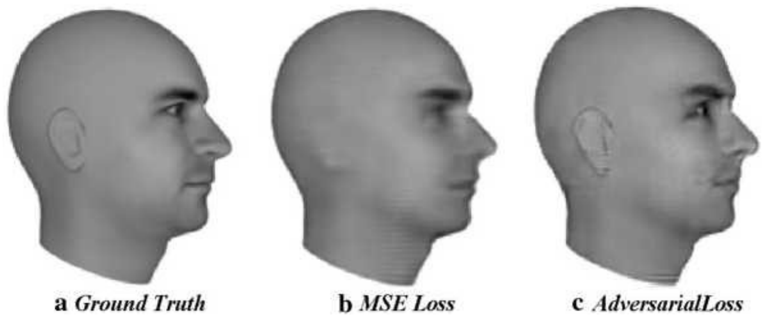


Fig. 2.1 An illustration of the importance of an adversarial loss [111]

Despite remarkable achievements in generating sharp and realistic images, GANs suffer from certain drawbacks.

• *Non convergence* Both generator and discriminator networks in GANs are trained simultaneously using gradient descent in a zero-sum game. As a result, improvement of the generator network comes at the expense of discriminator and vice versa. Hence there is no guarantee of GANs convergence.

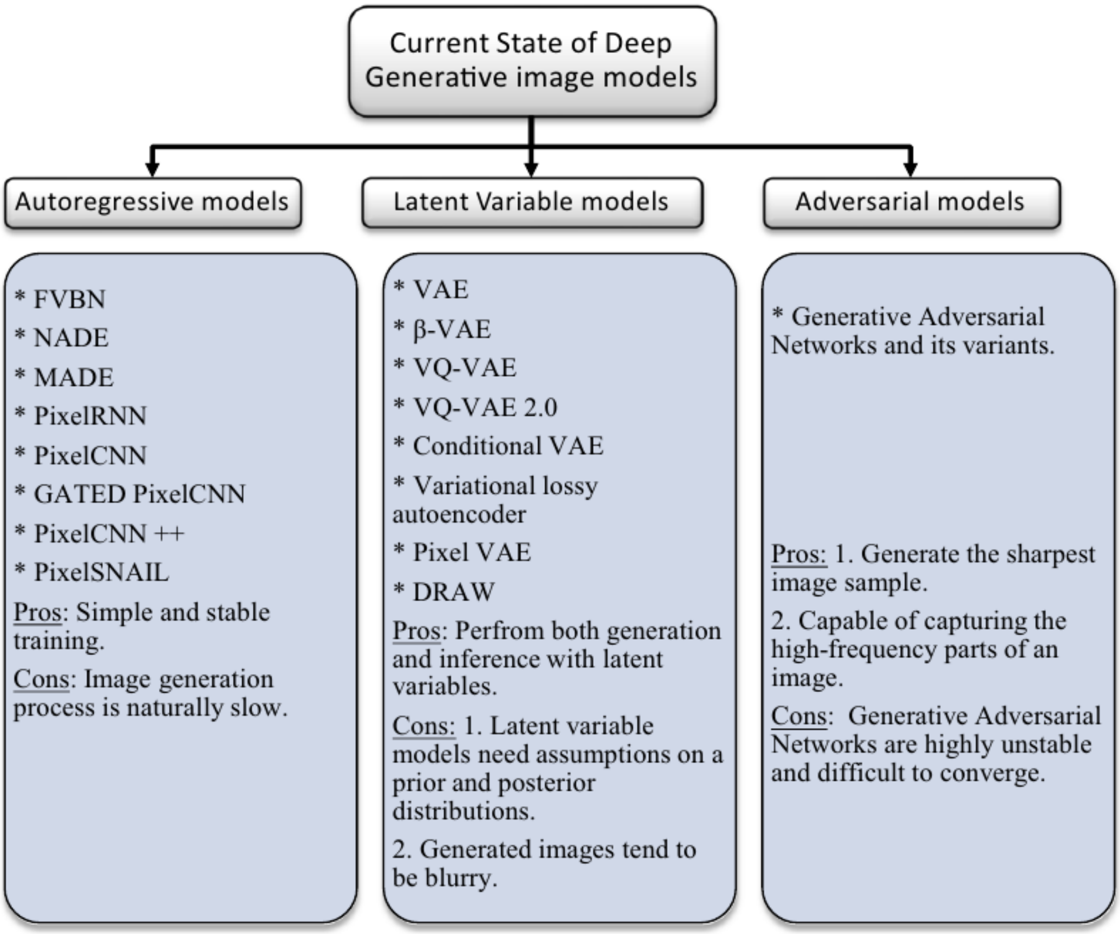


Fig. 2.2. Comparative summary of Deep generative models discussed in “Deep Generative image models” section

• *Mode collapse* Generator network achieves a state where it continues to generate samples with little variety, although trained on diverse datasets. This form of failure is referred to as mode collapse.

• *Vanishing gradient problems* If the discriminator is perfectly trained early in the training process, then there would be no gradients left to train the generator due to vanishing gradients.

Therefore, many GAN-variants have been proposed to overcome these drawbacks. These GAN-variants can be grouped into three categories:

1. *Architecture variants* In terms of architecture of generator and discriminator networks, the first proposed GANs use the Multi-layer perceptron (MLP). Owing to the fact that ConvNets work well with high resolution image data taking into account of the spatial structure of data, a Deep Convolutional GAN (DCGAN) [112] replaced the MLP with the deconvolutional and convolutional layers in generator and discriminator networks respectively.

Autoencoder based GANs such as AAE, BiGAN , VAE-GAN, DEGAN [116], VEEGAN [117] etc., have been proposed to combine their construction power of autoencoders with the sampling power of GANs.

Conditional based GANs like Conditional GAN (CGAN) [118], Auxiliary Classifier GAN (ACGAN) [119], VACGAN [120], infoGAN [121], and SCGAN [122] focused on controlling mode of data being generated by conditioning model on conditional variable.

1. *Training tricks* GANs are difficult to train. Improved trainings tricks such as feature matching, minibatch discrimination, historical averaging, one-sided label smoothing, and Two Time-Scale Update Rule have been suggested to ensure that GANs converge to achieve Nash equilibrium.

2. *Objective variants* In order to improve the stability and overcome vanishing gradient problems, different objective functions have been explored in .

The following section of this review moves on to describe in greater detail the selected GAN variants.

2.2 GENERATIVE ADVERSARIAL NEURAL NETWORKS ARCHITECTURE VARIANTS

The performance and training stability of GANs are highly influenced by the architecture of the generator and the discriminator networks. Various architecture variants of GANs have been proposed that adopt several techniques to improve performance and stability.

2.2.1 CONDITIONAL BASED GAN VARIANTS

The standard GAN [67] architecture does not have any control on the modes of data being generated. Van den Oord et al. [89] argue that the class conditioned image generation can significantly enhance the quality of generated images. Several conditional based GANs have been proposed that learn to sample from a conditional distribution $p(x|y)$ instead of marginal $p(x)$. Conditional based GANs variants (Fig.2.3) can be classified into two groups: 1. Supervised and 2. Unsupervised conditional GANs.

Supervised conditional GANs variants require a pair of images and corresponding prior information such as class label. The prior information could be class labels, textual descriptions, or data from other modalities.

cGAN Mirza and Osindero [118] proposed conditional Generative Adversarial Network (cGAN), to have a control on kind of data being generated by conditioning the model on prior information y. Both discriminator and generator in cGAN are conditioned by feeding y as additional input. Using this prior information, cGAN is guided to generate output images with desired properties during the generation process.

ACGAN Auxiliary classifier Generative Adversarial Network (ACGAN) [119] is an extension of the cGAN architecture. The discriminator in the ACGAN receives only the image, unlike the cGAN that gets both the image and the class label as input. It is modified to distinguish real and fake data as well as reconstruct class labels. Therefore, in addition to real fake discrimination, the discriminator also predicts class label of the image using an auxiliary decoder network.

VACGAN The major problem with ACGAN is that it will affect the training convergence because of mixing the loss of classifier and discriminator into a single loss. Versatile Auxiliary Generative Adversarial Network (VACGAN) [120] separates out classifier loss by introducing a classifier network in parallel to the discriminator.

Unsupervised conditional GAN variants do not use prior information to control the structure of the generated image. Instead, specific information such as hair color, age, and gender is learned during training. Therefore, they need an additional algorithm to decompose the latent space into a deconvolved latent vector c (containing the features) and the standard input noise vector z. The content and representation of the images are then controlled by the noise vector z and the decoded latent vector c, respectively.

Info-GAN Information maximizing Generative Adversarial Network (Info-GAN) [121] splits an input latent space into the standard noise vector **z** and additional latent vector **c**. The latent vector c is then made meaningful disentangled representation by maximizing the mutual information between latent vector **c** and generated images $G(z, c)$ using additional Q network. *SC-GAN* Similarity constraint Generative Adversarial Network (SC-GAN)[122] attempts to learn disentangled latent representation by adding the similarity constraint between latent vector **c** and generated images $G(z, c)$. Info-GAN uses an extra network to learn disentangle representation, while SC-GAN only adds an additional constraint to a standard GAN. Therefore, SCGAN simplifies the architecture of Info-GAN.

2.2.2 CONVOLUTIONAL BASED GAN

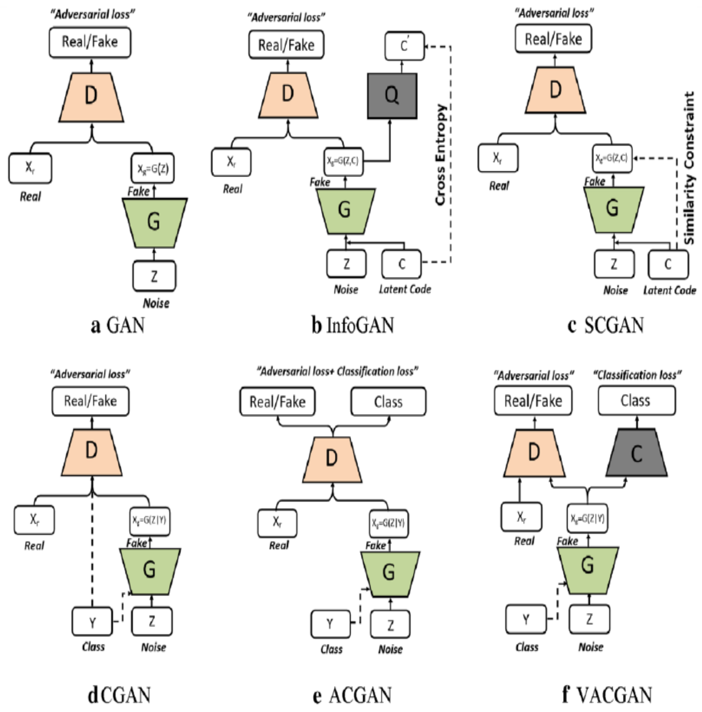
DCGAN Deep Convolutional Generative Adversarial Network (DCGAN) [112] is the first work that deploys convolutional and transpose-convolutional layers in the discriminator and generator, respectively. The salient features of the DCGAN architecture are enumerated as follows:

• First, the generator in DCGAN consists of fractional convolutional layers, batch normalization layers and ReLU activation functions.

• Second, the discriminator is composed of strided convolutional layers, batch normalization layers and Leaky ReLU activation functions.

• Third, uses Adaptive Moment Estimation (ADAM) optimizer instead of stochastic gradient descent with momentum.

2.2.3 MULTIPLE GANS



Where, ***G*** - Generator, ***D*** - Discriminator, ***Q*** - Q-network to learn disentangled representation, ***C*** - Classifier Network, X_r - Original Image, X_g - Generated Image

Fig. 2.3. A schematic view of the vanilla GAN and variants of Conditional GANs

In order to accomplish more than one goal, several frameworks extend the standard GAN to either multiple discriminators, generators, or both (Fig.2.4).

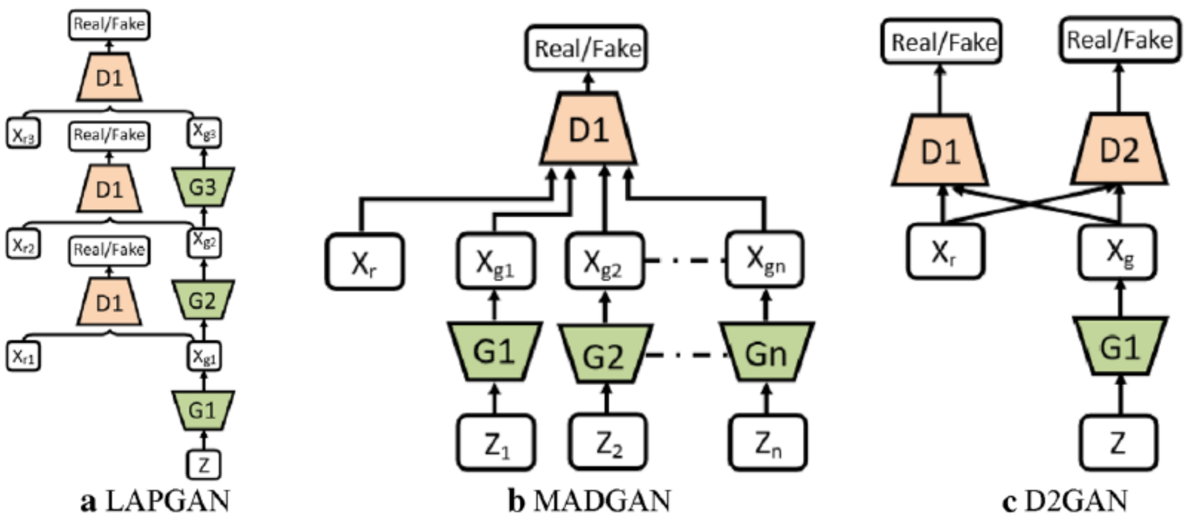
ProGAN In an attempt to synthesize higher resolution images Progressive Growing of Generative Adversarial Network (ProGAN) [131] stacks each layer of the generator and discriminator in a progressive manner as training progresses.

LAPGAN Laplacian Generative Adversarial Network (LAPGAN) [132] is proposed for the generation of high quality images. This architecture uses a cascade of Con-vNets within a Laplacian pyramid framework. LAPGAN utilizes several Generator-Discriminator networks at multiple levels of a Laplacian Pyramid for an image detail enhancement. Motivated by the success of sequential generation, Im et al. [133] introduced Generative Recurrent Adversarial Networks (GRAN) based on recurrent network that generate high quality images in a sequential process, rather than in one shot.

D2GAN Dual discriminator Generative Adversarial Network (D2GAN) [134] employs two discriminators and one generator to address the problem of mode collapse. Unlike GANs, D2GAN formulates a three-player game that utilizes two discriminators to minimize the KL and reverse KL divergences between true data and the generated data distribution.

MADGAN Multi-agent diverse Generative Adversarial Network (MADGAN) [135] incorporates multiple generators that discover diverse modes of the data while maintaining high quality of generated images. To ensure that different generators learn to generate images from different modes of the data, the objective of discriminator is modified to detect the generator which generated the given fake image along with discriminating the real and fake images.

G : Generator, D: Discriminator, X_r: Original Image, X_g: Generated Image



a LAPGAN ***b*** MADGAN ***c*** D2GAN

Fig.2.4. A schematic view of Variants of GANs with multiple discriminators and generators

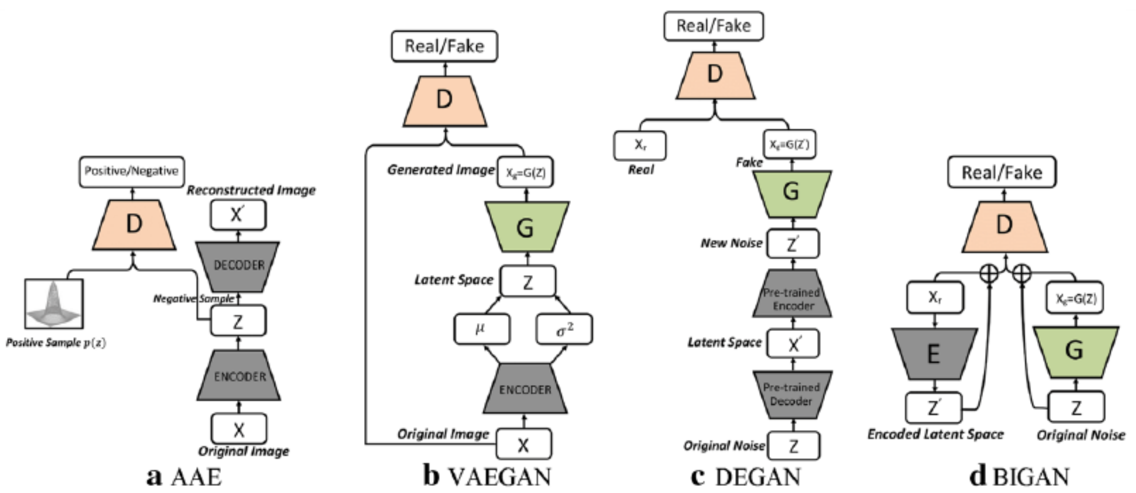
CoGAN Coupled GAN(CoGAN) [136] is used for generating pair of like images in two different domains. CoGAN is composed of a set of GANs-GAN1 and GAN2, each accountable for synthesizing images in one domain. It leans a joint distribution from two-domain images which are drawn individually from the marginal distributions.

CycleGAN and DiscoGAN [137] use two generators and two discriminators to accomplish unpaired image to image translation tasks. CycleGAN [138] adopts the concept of cycle consistency from machine translation, where a sentence translated from English to Spanish and translate it back from Spanish to English should be identical.

2.2.3 AUTOENCODER BASED GAN VARIANTS

The standard GANs architecture is unidirectional and can only map from latent space *z* to data space *x*, while autoencoders are bidirectional. The latent space learned by encoders is the distribution that contains compressed representation of the real images. Several variants of GANs that combine GAN and encoder architecture are proposed to make use of the distribution learned by encoders (Fig.2.5). Attributes editing of an image directly on data space *x* is complex as image distributions are highly structured and high dimensional. Interpolation on latent space can facilitate to render complicated adjustments in the data space *x*.

DEGAN In standard GANs architecture, the input to the generator network is the noise vector that is randomly sampled from a Gaussian distribution N (0, 1), which may create a deviation from the true distribution of real images. Decoder Encoder Generative adversarial Network (DEGAN) [116] adopt decoder and encoder structure from VAE, pretrained on the real images. The pretrained decoder and encoder structure transform random Gaussian noise to distribution that contains intrinsic information of the images which is used as input of the generator network.



Where, ***G*** - Generator, ***D*** - Discriminator, ***E*** - Encoder, $\boldsymbol{X_r}$ - Original Image, $\boldsymbol{X_g}$ - Generated Image

a AAE ***b*** VAEGAN ***c*** DEGAN ***d*** BIGAN

Fig.2.5 A schematic view of Variants of GANs based on Encoder and decoder architecture

VAEGAN Variational autoencoder Generative Adversarial Network (VAEGAN) [115] jointly trains VAE and GAN by replacing the decoder of VAE with GAN framework. VAEGAN employs feature wise adversarial loss of GAN in lieu of element wise reconstruction loss of VAE to improve quality of image generated by VAE. In addition to latent loss and adversarial loss, VAEGAN uses content loss, also known as perceptual loss, which compares two images based on high level feature representation from pretrained VGG Network [11].

AAE Unlike VAEGAN that discriminates in data space, adversarial autoencoders (AAE) [113] imposes a discriminator on the latent space as learning the latent code distribution is simpler than data distribution. The discriminator network discriminates between a sample drawn from latent space and from the distribution $p(z)$ that we are trying to model.

ALI and BiGAN In addition to generator network, Adversarially Learned Inference (ALI) [114] model and Bidirectional Generative Adversarial Network (BiGAN) contain an encoder component E that simultaneously learn inverse mapping of the input data x to the latent code z. Unlike other variants of GAN where the discriminator network receives only real or artificially generated images, in the BiGAN and ALI model, the discriminator network receives both image and latent code pair.

VEEGAN [117]: addresses the problem of mode collapse through addition of a reconstruction network that reverses the action of the generator network. Reconstruction network takes in synthetic images then transforms them to noise, while generator network takes noise as an input and reconstructs them into synthetic image. In addition to adversarial loss, difference between the reconstructed noise and initial noise is used to train the network. Both generator and reconstruction networks are jointly trained, which encourages generator network to learn true distribution, hence solving the mode collapse problem.

Several other GANs have been proposed for image super resolution. The goal of super resolution is to upsample low resolution images to a high resolution one. Ledig et al.

proposed Super-Resolution GAN (SRGAN) [139] for image super resolution,which takes poor quality image as input, and generates high quality image with 4 × resolution. The generator of the SRGAN uses very deep convolutional layers with residual blocks. In addition to an adversarial loss, SRGAN includes a content loss. The content loss is computed as the euclidean distance between the feature maps of the generated high quality image and the ground truth image, where feature maps are obtained from a pretrained VGG19 [140] network. Zhang et al. [141] combined a self attention mechanism with GANs (SAGAN) to handle long range dependencies that make the generated image look more globally coherent. Image-to-image translation GANs such as Pix2Pix GAN [142], Pix2pix HD GAN [143], and CycleGAN [137] learn to map an input image from a source domain to an output image from a target domain. A summary of architectural variants of GANs are summarized in Table 1.

2.3 OBJECTIVE VARIANTS

The main objective of GAN is to approximate the real data distribution. Hence, minimizing distance between the real data distribution (p_r) and the GAN generated data distribution (p_g) is a vital part of training GAN. As stated in "Deep Generative image models" section, standard GAN [67] uses Jensen Shannon divergence to measure similarity between real and generated data distributions $D_{JS}(p_r \| p_g)$. However, JS divergence fails to measure distance between two distributions with negligible or no overlap. To improve performance and achieve stable training of GAN, several distances or divergence measures have been proposed instead of JS divergence.

WGAN Wasserstein Generative Adversarial Network (WGAN) [123] replaces JSD from the standard GAN with the Earth mover Distance (EMD). EMD also known as Wasserstein Distance (WD) can be interpreted informally as minimum amount of work to move earth (quantity of mass) from the shape of one distribution $p(x)$ to that of another distribution $q(x)$ so as to match shape of both the distributions. WD is smooth and can provide meaningful distance measure between distributions with negligible or no overlap. WGAN imposes an additional Lipchitz constraint to use WD as the loss in the discriminator, where it deploys weight clipping to enforce weights of the discriminator to satisfy Lipchitz constraint after each training batch.

WGAN-GP Weight clipping in the discriminator of a WGAN greatly diminishes its capacity to learn and often fails to converge. WGAN-GP [124] is an extension of WGAN that replaces weight clipping with gradient penalty to enforce discriminator to satisfy Lipchitz constraint. Furthermore, Petzka et al. [125] proposed a new regularization method, also known as WGAN-LP, that enforces the Lipschitz constraint.

LSGAN Least squares Generative Adversarial Network (LSGAN) [126] deploys least square loss instead of the cross entropy loss in discriminator of the standard GAN to overcome the problem of Vanishing gradient as well as improving quality of generated image.

EBGAN Energy Based GAN (EBGAN) [127] uses auto-encoder architecture to construct the discriminator as an energy function instead of a classifier. The Energy of EBGAN is the mean squared reconstruction error of an autoencoder, providing lower energy to the real images and high energy to generated images. EBGAN exhibits faster and more stable behavior than standard GAN during training.

Table 2.1 An overview of GANs variants discussed in "Architecture variants" section

Table 2.1

Categories	**GAN Type**	**Main Architectural Contributions to GAN**
Basic GAN	GAN [67]	Use Multilayer perceptron in the generator and discriminator
Convolutional Based GAN	DCGAN [112]	Employ Convolutional and transpose-convolutional layers in the discriminator and generator respectively

	PROGAN [131]	Progressively grow layers of GAN as training progresses
Condition based GANs	cGAN [118]	Control kind of image being generated using prior information
	AC GAN [119]	Add a classifier loss in addition to adversarial loss to reconstruct class labels
	VACGAN [120]	Separate out classifier loss of ACGAN by introducing separate classifier network parallel to the discriminator
	info GAN [121]	Learn disentangled latent representation by maximizing mutual information between latent vector and generated images
	SCGAN [122]	Learn disentangled latent representation by adding the similarity constraint on the generator
Latent representation based GANs	DE GAN [116]	Utilize the pretrained decoder and encoder structure from VAE to transform random Gaussian

		noise to distribution that contains intrinsic information of the real images
	VAEGAN [115]	Combine VAE and GAN
	AAE [113]	Impose discriminator on the latent space of the autoencoder architecture
	VEEGAN [117]	Add reconstruction network that reverse the action of generator network to address the problem of mode collapse
	BiGAN [114]	Attach encoder component to learn inverse mapping of data space to latent space
Stack of GANs	LAPGAN [132]	Introduce Laplacian pyramid framework for an image detail enhancement
	MADGAN [135]	Use multiple generators to discover diverse modes of the data distribution

	D2GAN [134]	Employ two discriminators to address the problem of mode collapse
	CycleGAN [137]	Use two generators and two discriminators to accomplish unpaired image to image translation task
	CoGAN [136]	Use two GANs to learn a joint distribution from two-domain images
Other variants	SA GAN [141]	Incorporate self-attention mechanism to model long range dependencies
	GRAN [133]	Recurrent generative model trained using adversarial process
	SRGAN [139]	Use very deep convolutional layers with residual blocks for image super resolution

Same as EBGAN, Boundary Equilibrium GAN (BEGAN) [128], Margin adaptation GAN [129] and dual agent GAN [130] also deploy an auto-encoder architecture as the discriminator. The discriminator loss of BEGAN uses Wasserstein distance to match the distributions of the reconstruction losses of real images with the generated images.

There are also several other objective functions based on Cramer distance [144], Mean/covariance Minimization [145], Maximum mean discrepancy [146], Chi-square have been proposed to improve performance and achieve stable training of GAN.

2.4 TRAINING TRICKS

Although research on different architectures and target functions of GANs continues to improve learning stability, several techniques have been proposed in the literature to achieve excellent learning results. Radford et al. [112] showed that the use of leaky rectified activation functions in both the generator and discriminator layers gives higher performance than the use of other activation functions. Salimans et al. proposed several heuristic approaches that can improve the performance and stability of ANN training. First, feature mapping changes the goal of the generator to minimize the statistical difference between the features of the generated and real images. Thus, the discriminator is trained to learn important features of real data. Second, mini-packet discrimination, where the discriminator processes sample packets rather than in isolation, helps prevent mode collapse because the discriminator can identify if the generator continues to generate a sample with little diversity. Third, historical averaging, which takes the running average of the parameters in the past and penalizes if there is a large difference between the parameters, which can help the model approach equilibrium. Finally, one-way label smoothing provides smoothed discriminator labels instead of 0 or 1, which can smooth the classification boundary of the discriminator.

Sonderby and others. proposed the idea of crippling the discriminator by introducing noise to the samples instead of the labels, which prevents the discriminator from overfitting. Heusel et al. used a separate learning rate for the generator and the discriminator, and trained the GAN using the two-time-update rule (TTUR) to ensure the model converged to a stationary local Nash equilibrium. To stabilize the learning of the discriminator, Miyato et al. proposed a normalization technique called spectral normalization.

2.5 TAXONOMY OF CLASS IMBALANCE IN VISUAL RECOGNITION TASKS

This section introduces different GANs applied to discrete problems in various vision techniques. We divide the problem of uneven distribution into three types: 1. Imbalance of the image level in the distribution; 2. Imbalance of product level in product search; 3. Pixel-level instability in segmentation tasks. Understanding this fuzzy distribution will provide an important basis for further research on the use of GANs to create synthetic images.

Class imbalances in classification

Image classification is the task of separating the input image into possible groups. Classification can be divided into two different problems: binary classification and multivariate classification. Binary classification involves assigning an input image to one of two categories, whereas multiclassing involves two or more categories. A classic example of a binary image classification problem is identifying cats or dogs in each input image. High inequality figures [152], including class inequality and class inequality, lead to poor performance.

CHAPTER 3
STRUCTURAL AND PARAMETRIC SYNTHESIS OF WGAN-GP

Generatively adversarial networks (GANs) are a powerful class of generative models that allow to reconstruct the distribution of any set of images, but they suffer from poor convergence due to the attenuation of gradients vanishing) and the instability of the training process, which leads to the so-called mode collapse (mode collapse), i.e. the problems of the impossibility of further training of the network (generator-discriminator) due to the fact that one of the components, i.e. the generator or the discriminator, intercepts the initiative.

This problem is especially inherent in early versions of GANs , namely Vanilla GANs (2014) and DCGAN (2015). Wasserstein GAN (WGAN) was proposed to solve this problem , but it could not completely eliminate the 3.1 mode collapse problem, so its modification WGAN-GP was proposed and disclosed in this paper.

3.1 WGAN-GP TOPOLOGY

WGAN-GP extends the idea of WGAN and adds a gradient penalty to the loss function in order to stabilize the training process. In turn, WGAN uses the Wasserstein distance as a metric without changing the generator and discriminator architecture itself.

Thus, WGAN-GP can use DCGAN and even Vanilla GAN topologies, i.e., a sweeping algorithm can be used as a generator neural network , or multilayer perceptron , and as a discriminator you can use convolutional neural network , or multilayer perceptron.

The sweep network for the generator and the convolutional network for the discriminator was chosen as the basis , because they are better adapted to work with images and are much less prone to retraining than the multilayer perceptron . Thus, in order to be able to compare the results of WGAN-GP and DCGAN, it was decided to choose the architecture of the latter [3] . The architecture of the generator and discriminator is discussed in detail below:

3.1.1 GENERATOR ARCHITECTURE

For example, let's consider a generator that generates a 3-channel image with a size of 32 by 32 pixels:

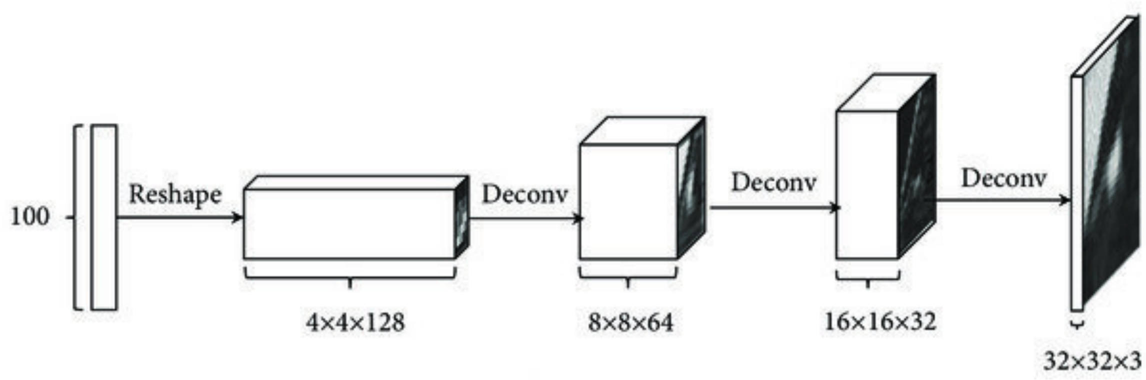


Fig. 3.1 Generator architecture

- At the input, the noise vector with dimension 100 is taken from a uniform distribution (on the segment [0, 1]).
- The next layer contains 4 ·4 ·128 = 2048 neurons, which are reshaped in such a way that 128 4x4 feature maps are formed from them.
- The following layers are deconv (Deconv). All of them contain 4x4 kernels (kernel - K), stride 2 (stride - S) and padding 1 (padding - P). Thus, using the formula, $\mathrm{H}_{out} = (H_{in} - 1) \cdot S - 2P + K$ we get the size of the original feature maps:

for the 1st scanning layer$\mathrm{H}_{out} = (4 - 1) \cdot 2 - 2 \cdot 1 + 4 = 8$

for the 2nd scanning layer$\mathrm{H}_{out} = (8 - 1) \cdot 2 - 2 \cdot 1 + 4 = 16$

for the 3rd scanning layer$\mathrm{H}_{out} = (16 - 1) \cdot 2 - 2 \cdot 1 + 4 = 32$

- For each scanning layer, the number of output feature maps is equal to the number of cores:

The 1st scanning layer contains 64 cores and receives 128 feature maps at the input;

The 2nd deconvolution layer contains 32 cores and receives 64 feature maps as an input, that is, each core is applied to two different input feature maps and the two resulting feature maps are added, so the output has half the number of feature maps as the input for 1st and 2nd scanning layers;

The 3rd sweep layer contains 3 cores and receives 32 feature maps as input, that is, two cores are applied to 11 feature maps and another to 10, since $11 \cdot 2 + 10 = 32$. Feature maps obtained for each core are added - this allows to reduce the number of original feature maps.

- Thus, at the output we get a 3-channel image with a size of 32x32.

Bias (bias – b) in the generator were not used. Taking this fact into account, the total number of parameters (weights) that the generator must adjust is calculated below:

$$100 \cdot 4 \cdot 4 \cdot 128 + \ 64 \cdot 4 \cdot 4 + 32 \ \cdot 4 \ \cdot 4 + 3 \cdot 4 \cdot 4 = 206384$$

3.1.2 GENERATOR ACTIVATION FUNCTION

As activation functions for the generator, it was proposed to use ReLU (rectified linear unit), as it is easy to calculate and is less prone to gradient damping. ReLU is calculated as follows (x is the signal of the neuron):

$$ReLU(x) = \max{(x, 0)} \tag{3.1}$$

ReLU is placed after each scan layer.

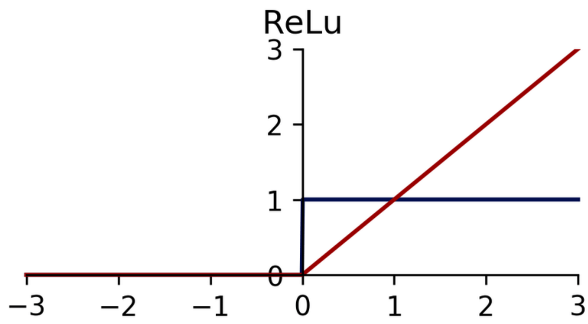


Fig3.2 ReLU function (in red) and its derivative (in blue)

The hyperbolic tangent (Tanh) was used as the activation function of the last layer, since it takes on values in the interval [-1, 1], which are then easily displayed to restore the brightness of each pixel and thus obtain an image. Below is the formula for calculating Tanh (x), where x is the signal of the neuron:

$$Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.2}$$

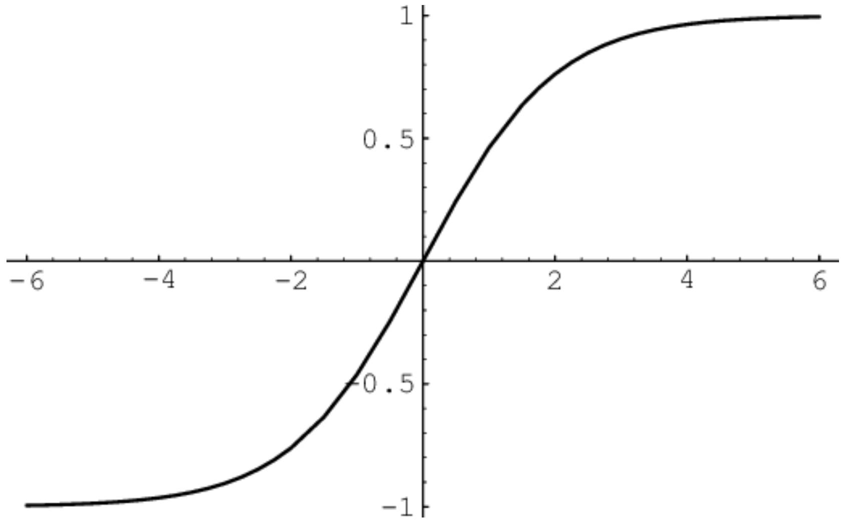


Fig. 3. 3 Tahn function

3.1.3 NORMALIZATION OF THE GENERATOR

To increase training stability and avoid the problem of decaying gradients, batch normalization (BatchNorm) was also used. Batch normalization layers are placed immediately after the sweep layers and before the activation function. Normalization by batches consists in reducing the data to the segment [0, 1].

It is worth noting that during training, the generator can generate several images at the same time (batch) - for this, it receives not one vector as an input, but a whole set of vectors. Thus, each such vector after passing through the generator will correspond to one image.

To carry out normalization by batch , it is necessary for each element of the input feature maps along the batch to determine the minimum value, which is subtracted from all other values at similar positions for other objects from this batch .

In the figure below, for example, the lower left element of the first (blue) feature map is marked in red.

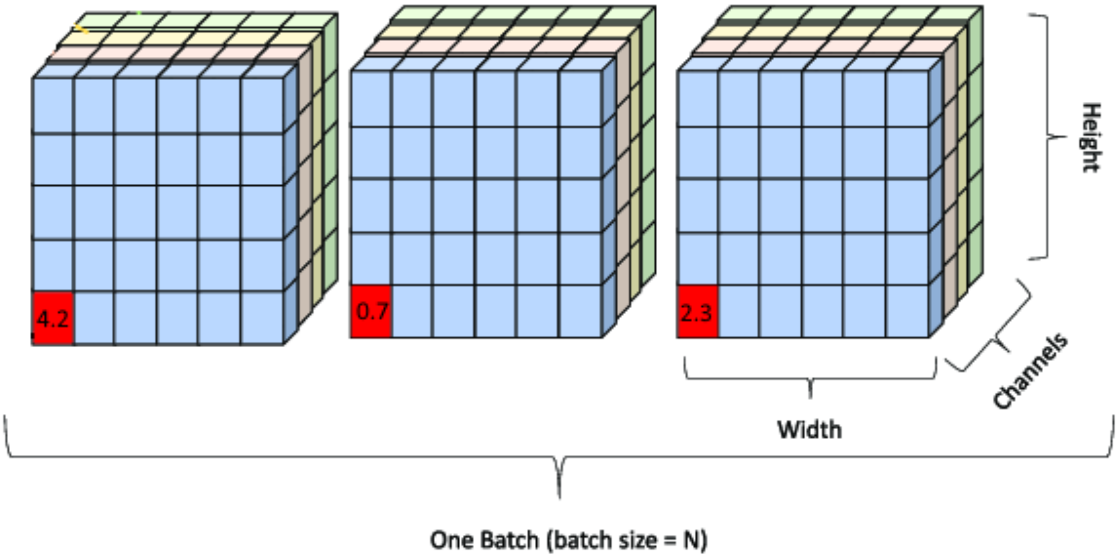


Fig. 3.4 Normalization by batches

The batch is subtracted from each corresponding element of the feature maps and divided by the current maximum. Below is an example for Fig. 3.4 (a batch of three objects)

Step 1. Subtract the minimum element along the batch:

$$4.2 - 0.7 = 3.5; \quad 0.7 - 0.7 = 0; \quad 2.3 - 0.7 = 1.6$$

Step 2. Divide by the current maximum element along the batch:

$$\frac{3.5}{3.5} = 1; \quad \frac{0}{3.5} = 0; \quad \frac{1.6}{3.5} = 0.457$$

A similar procedure is carried out for all positions of all feature maps along the batch.

Thus, after the batch normalization layer , all numbers will be in the interval [0, 1], which to a certain extent allows you to avoid fading gradients and numerical inaccuracies when working with large numbers. Batch normalization also reduces the sensitivity of the model to the way the weights are initialized.

3.1.4 GENERATOR OPTIMIZER

As a gradient descent optimizer, one of the most effective modern methods - Adam was chosen . It involves setting three hyperparameters , namely damping coefficients $\beta_1 and\ \beta_2$, as well as the initial learning rate $\Lambda\,(learning\ rate)$. Adam is an adaptive algorithm of gradient descent, i.e. for each direction (each component of the gradient vector) the update occurs separately. So, according to Adam , the scales $\theta_g$must be updated as follows:

$$\theta_g := \theta_g - \frac{\Lambda}{\sqrt{S}} v$$

(3.3)

Where $L_g$is the loss function of the generator, $v$is the inertia (cumulative speed), and Sis the cumulative sum of squares. They are defined as follows (loss function of the generator below in the learning metrics section):

$$\mathrm{v} := \beta_1 \cdot v + (1 - \beta_1) \cdot \nabla_\theta L_g$$

(3.4)

$$\mathrm{S} := \beta_2 \cdot \mathrm{S} + (1 - \beta_2) \cdot \left(\nabla_\theta L_g\right)^2$$

(3.5)

S and v are calculated element by element , that is, for each component of the gradient separately, which allows better balancing of convergence along different directions.

3.2.1 ARCHITECTURE OF THE DISCRIMINATOR

As an example, let's consider a discriminator that recognizes an image with a size of 32 by 32 pixels:

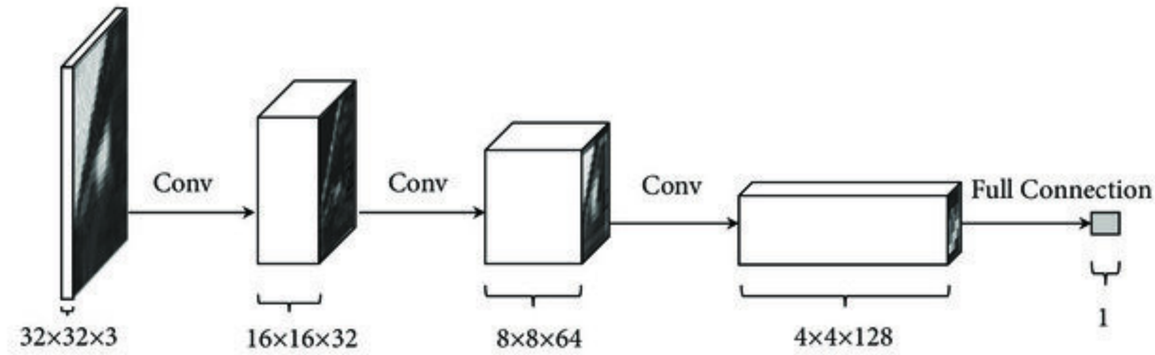


Fig. 3.5 Architecture of the discriminator

- At the entrance, there are 3 character cards measuring 32x32.
- Next are three convolutional layers (Conv). All of them contain 4x4 kernels (kernel - K), stride 2 (stride - S) and padding 1 (padding - P). Thus, according to the formula $\mathrm{H}_{out} = \frac{H_{in}+2P-K}{\mathrm{S}} + 1$, we get the size of the original feature maps:

for the 1st convolutional layer $\mathrm{H}_{out} = \frac{32+2\cdot1-4}{2} + 1 = 16$

for the 2nd convolutional layer $H_{out} = \frac{16+2\cdot1-4}{2} + 1 = 8$

for the 3rd convolutional layer $H_{out} = \frac{8+2\cdot1-4}{2} + 1 = 4$

- flattened feature maps as input , i.e. 128 4x4 feature maps are extracted into a vector of length $4 \cdot 4 \cdot 128 = 2048$. So, the last neuron contains 2048 weights. It weights the input flattened feature maps and returns a probability.
- In order to increase the number of feature maps when passing through the discriminator to the same input feature map, several kernels are used:

The 1st convolutional layer contains 32 kernels and receives 3 feature maps as an input, that is, 11 different kernels are applied to two input feature maps, and 10 kernels are applied to the other input feature map, since$11 \cdot 2 + 10 = 32$

The 2nd convolutional layer contains 64 cores and receives 32 feature maps as an input, that is, 2 different cores are used for each input feature map;

The 3rd convolutional layer contains 128 cores and receives 64 feature maps as an input, that is, 2 different cores are used for each input feature map;

- Thus, at the output we get a number from the interval [0, 1], that is, the probability that the input image is real.

Bias (bias – b) were not used in the discriminator. Taking this fact into account, the total number of parameters (weights) that the discriminator should adjust is calculated below:

$$32 \cdot 4 \cdot 4 + 64 \cdot 4 \cdot 4 + 128 \cdot 4 \cdot 4 + 4 \cdot 4 \cdot 128 = 5632$$

3.2.2 THE DISCRIMINATOR ACTIVATION FUNCTION

As activation functions for the discriminator, it was proposed to use LeakyReLU with a hyperparameter $\alpha = 0.2$, because it, like ReLU , is also quite simple to calculate and is even more resistant to gradient attenuation than ReLU itself .

LeakyReLU is calculated as follows (x is the signal of the neuron, $\alpha - hyperparameter$):

$$\text{LeakyReLU}(x, \alpha) = \max(x, \alpha \cdot x) \tag{3.6}$$

LeakyReLU is placed after each convolutional layer.

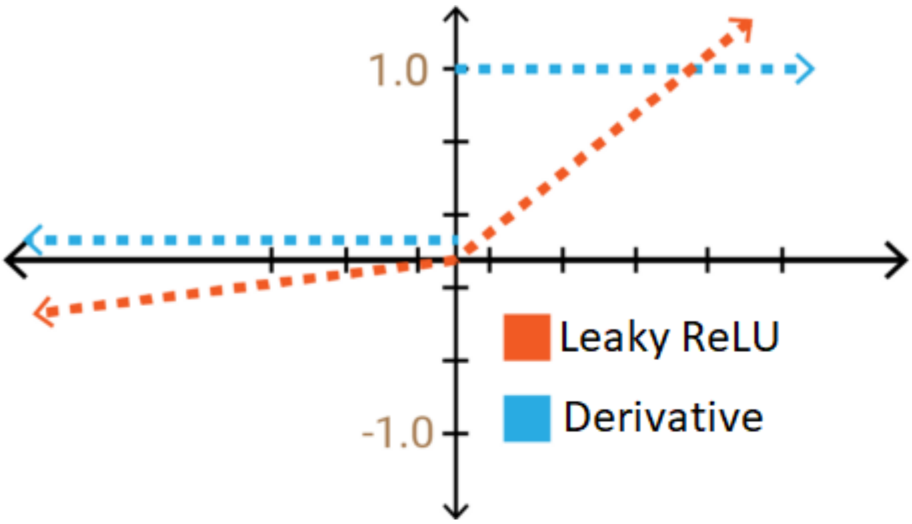
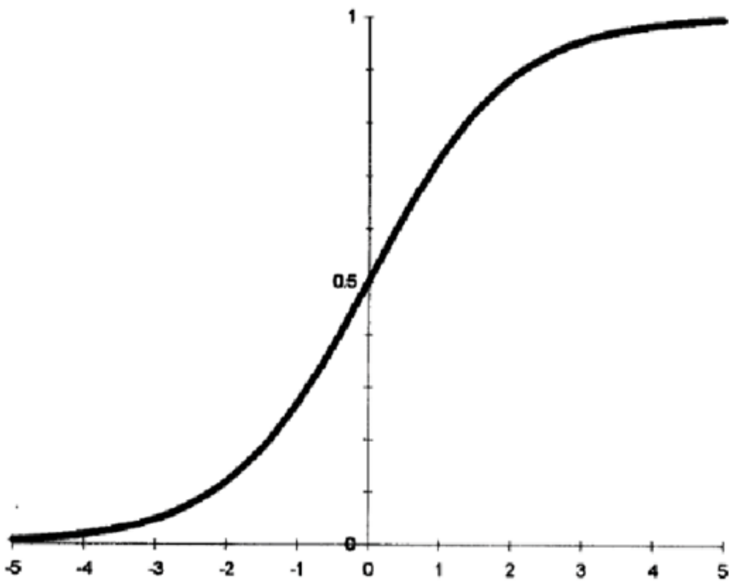


Fig. 3.6 The function LeakyReLU(in orange) and its derivative (in blue)

Sigmoid (Sigmoid) was used as the activation function of the last layer , because it acquires values in the interval [0, 1], that is, it directly reflects the probability. Below is the formula for calculating Sigmoid (x), where x is the signal of the neuron:

$$\mathrm{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{3.7}$$



Sigmoid function

3.2.3 NORMALIZATION OF THE DISCRIMINATOR

batch normalization (BatchNorm) was used . Batch normalization layers are placed immediately after the convolutional layers and before the activation function. More about normalization above (Fig. 4).

3.2.4 THE DISCRIMINATOR OPTIMIZER

As the gradient descent optimizer of the discriminator, as well as for the generator, Adam was chosen . According to Adam, scales $\theta_d$must be updated as follows:

$$\theta_d \coloneqq \theta_d - \frac{\Lambda}{\sqrt{S}} v \tag{3.8}$$

The relevant formulas for finding $v$and Sare above in the generator optimizer section.

3.3. FORMULATION OF THE PROBLEM

Given:

- A set of images of a certain fixed size (training sample) that belong to a certain probability distribution P_r.
- Random noise vector distribution P_z(e.g., uniform)
- Distribution of the generator $\mathrm{P}_g(z)$. Strictly speaking, a generator, like any neural network, is a function, and as you know, a function from a random vector is also a random vector, so the generator represents a certain probability distribution $\mathrm{P}_g(z)$.
- Initial values of the generator $\theta_g$and discriminator weights θ_d.

It is necessary to find:

- Weight values that will generate images from the same distribution as the given images (training sample).

In other words, it is possible to approximate the distribution of the generator $\mathrm{P}_g$to the distribution of the training sample as much as possible P_r.

3.4 METRICS

GANs training is a 0-sum game [1], i.e. a game of two players (generator G and discriminator D), in which the winning of one is equivalent to the losing of the other, so formally solving the above task of training GANs can be represented as finding the minimax point (Nash equilibria) of some expression $\mathrm{V}(G, D)$. In early versions of GANs , namely Vanilla GAN and DCGAN, the following expression was chosen as such :$\mathrm{V}(G, D)$

$$\mathrm{V}(G, D) = E_{x \sim P_r}(\ln D(x)) + E_{z \sim P_z}(\ln(1 - D(G(z))) \tag{3.9}$$

Thus, it is worth finding the point $\min_G \max_D V(G, D)$. The above metric corresponds to the Kullback- Leibler divergence (in its probabilistic sense) , that is, a measure that indicates how much information will be lost when replacing one probability distribution P_r with another P_g.

The generator is a function of the random noise vector z, so it minimizes $E_{z \sim P_z}(\ln(1 - D(G(z)))$,that is, the mathematical expectation of the logarithm of the probability that the images generated by the generator are fake, that is $1 - D(G(z))$, since $D(G(z))$is the probability that the image generated by the generator $G(z)$ is real.

Thus, the generator uses the following gradients:

$$\nabla_{\theta_g} \ln(1 - D(G(z))) \tag{3.10}$$

The discriminator receives both the real images and those generated by the generator, so it optimizes the entire expression at once $\mathrm{V}(G, D)$, i.e. it maximizes the mathematical expectation of the logarithm of the probability that it recognized the real image as real and the one generated by the generator as fake.

Thus, the discriminator uses the following gradients (here xis a real image):

$$\nabla_{\theta_d} \ln D(x) + \nabla_{\theta_d} \ln\left(1 - D(G(z))\right) \tag{3.11}$$

The above metric often leads to the problems of gradient damping and regime collapse [2] , therefore it was proposed to use the Wasserstein distance instead of the Kullback-Leibler divergence when finding the minimax point of the expression $\mathrm{V}(G, D)$, i.e. when finding the optimal values of the weights.

Using this idea, the expression problem can be reduced to:

$$\min_G \max_D V(G, D) = \min_G \max_D E_{x \sim P_r}(\ln D(x)) - E_{z \sim P_z}(\ln D(G(z)))$$

(3.12)

Accordingly, determine the gradients for the generator:

$$-\nabla_{\theta_g} \ln D(G(z))$$

(3.13)

and discriminator:

$$\nabla_{\theta_d} \ln D(x) - \nabla_{\theta_d} \ln \Big(D\big(G(z)\big)\Big)$$

(3.14)

These gradients are characterized by the fact that they "dampen" less often than those given above, but they do not allow to fully solve the problem of mode collapse, since the conditions imposed on the discriminator function often Dlead to the fact that the training process is disturbed and regime collapse occurs.

In order to eliminate this problem, WGAN-GP, i.e. Wasserstein GAN with, was proposed Gradient Penalty [3] is a generative -competitive Wasserstein network with a gradient penalty. Its idea is to modify the loss function of the discriminator by adding a term that is a gradient penalty. So, now the loss function of the discriminator will have the following form:

$$L_d = E_{x \sim P_r}(\ln D(x)) - E_{z \sim P_g}\big(\ln D\big(G(z)\big)\big) + \lambda \cdot E_{z \sim P_z}\Big(\big\|\nabla_{G(z)} D\big(G(z)\big)\big\|_2 - 1\Big)^2$$

(3.15)

Here $\lambda$is the hyperparameter of the gradient penalty, and $\big\|\nabla_{G(z)} D\big(G(z)\big)\big\|_2$ −Euclidean (L_2) is the gradient norm of the discriminator prediction for images generated by the generator.

3.5 DEFINITION OF PARAMETERS

In the corresponding sections above, the number of parameters that the generator and discriminator need to be adjusted during training was calculated. By adding them, we get the total number of parameters of the considered generative-competitive network:$206384 + 5632 = 212016$.

sweep and convolution layers were used in the generator and discriminator , respectively. The size of the image is only 32x32, that is, for higher quality images, the number of parameters will be several million.

3.6 DEFINITION OF HYPERPARAMETERS

Table 3.1

Marking	Name
λ	WGAN-GP gradient penalty factor
Λ	Adam's initial learning rate
β_1	inertia damping factor (cumulative velocity) Adam
β_2	attenuation coefficient of the cumulative sum of squares of the Adam gradients
α	the LeakyReLU activation function
N	batch size for training
n_d	number of iterations for training the discriminator relative to iterations for training the generator

It is worth noting that hyperparameters can also include the number of layers, the number of neurons in these layers, the sizes of convolution kernels, stride and padding , and even the choice of the activation function - all of them are hyperparameters of the network architecture.

3.7 THE PROBLEM OF MODE COLLAPSE

It can be argued that the reason for regime collapse in previous versions of GANs (Vanilla GAN, DCGAN, WGAN) is the lack of rational restrictions on the loss function of the discriminator, which directly leads to the fact that its weights change in such a way that the discriminator begins to make mistakes and misclassify images. That is why WGAN- GP introduces an additional term in the discriminator loss function (gradient penalty), which limits the possible values of the discriminator weights and to a certain extent prevents regime collapse.

3.8. RESULTS

For comparison WGAN - GP and DCGAN was selected a set of images of bedrooms. The pictures below show the corresponding images generated by the generator

Fig. 3.7 The image generated by the DCGAN generator

Fig. 3.8 The image generated by the WGAN - GP generator

Conclusions of chapter

As you can see in the pictures above WGAN - GP capable generate image much better compared to DCGAN . _ The images generated by the DCGAN generator are quite blurred, which can be caused by regime collapse during training: the discriminator is not able to distinguish between the generator images (even though they are blurred) and the real images. That is, the discriminator is unable to provide the generator with informative feedback to propagate the error and adjust its weights accordingly, so the DCGAN generator is unable to improve and generate quality images.

CHAPTER 4
DEVELOPMENT OF SYSTEM FOR GENERATION OF IMAGES. ANALYSIS OF MODIFICATIONS OF THE GAN GENERATIVE MODEL

4.1 FORMALIZATION DYNAMIC REPRESENTATION

For dynamic representation a chart has been created component in:

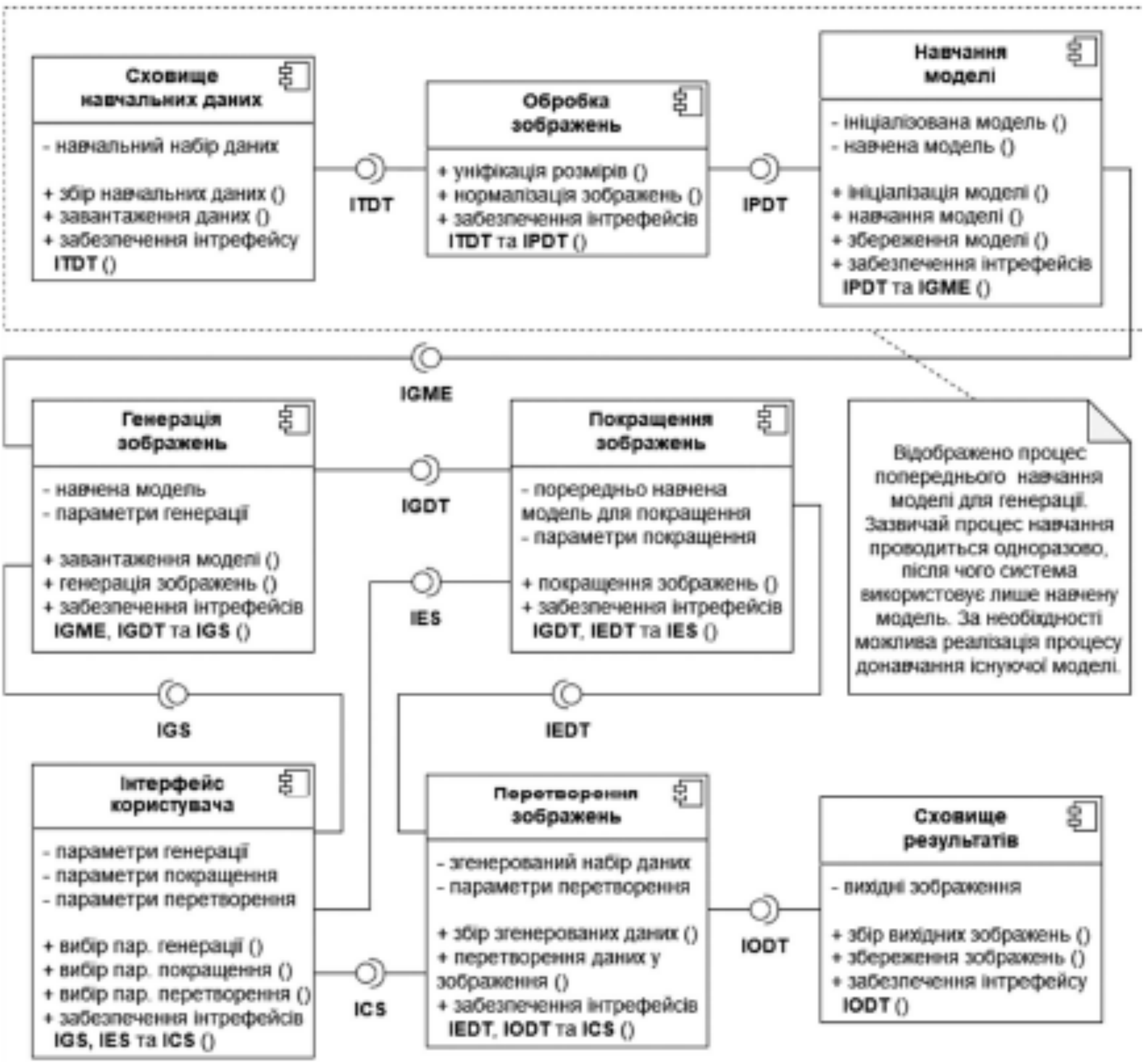


Fig. 4.1 – System model generation images. Chart component in UML notation

Functionality and appointment components systems:

- training data storage: a local data storage that contains the collected set of training images for model training;

- image processing: a set of tools for pre-processing training images and their further use in the model training process;

- model training: a set of tools that allows you to create a suitable model of the required structure and train it;

- image generation: a set of tools that uses a trained generative model and implements the process of creating images;

- image enhancement: a pre-trained model based on the based on the existing Real-ESRGAN method [4], which can be used to to increase the size of generated images;

- image conversion: a set of tools that implements converting data representing images back to image format

- user interface: a component that embodies an elementary graphical user interface for collecting image generation parameters;

- result storage: a local data storage, which will be the resulting generated images will be placed

List of interfaces:

- ITDT (Interface of Training Data Transferring) - interface for transferring training data (from the Training Data Storage module) to the input of the Image processing module;

- IPDT (Interface of Processed Data Transferring) - interface for transferring processed training data (from the Image Processing module) to the input of the Model Training module;

- IGME (Interface of Generative Model Extraction) - interface for receiving the resulting trained generative model (from the Model Training module) to the input of the Image Generation module;

- IGDT (Interface of Generated Data Transferring) - interface for transferring generated data (from the Image Generation module) to the input of the Image Enhancement module;

- IEDT (Interface of Enhance Data Transferring) - interface for transferring generated images of larger size (from the Image Enhancement module) to the the input of the Image Conversion module;

- IODT (Interface of Output Data Transferring) - an interface for transferring data in image format (from the Image Conversion module) to the input of the Result Storage module;

- IGS (Interface of Generating Settings) - interface for transferring generation parameters (from the User Interface module) to the input of the Image Generation module;

- IES (Interface of Enhancing Settings) - interface for transferring enhancement parameters (from the User Interface module) to the input of the Image Enhancement module;

- ICS (Interface of Converting Settings) - interface for transferring conversion parameters (from the User Interface module) to the input of the Image Conversion module.

4.2 PREPARATION DATA

The ArtBench-10 dataset was chosen as the basis [11]. It contains 60'000 images in 10 artistic styles (modern, baroque, expressionism, impressionism, postmodernism, realism, renaissance, romanticism, surrealism, ukiyo-e). Each style contains 6'000 images. The overall dimensions of the images are arbitrary.

Each image has been made square by dropping pixel bands horizontally (or vertically) on both sides, centering the image. The overall size of the images was unified to 64x64 pixels using a bicubic reduction algorithm.

For further use of this set, each image was represented as 3 color channels (red, green, blue), with 8-bit intensity values each (from 0 to 255). After that, each color channel was transformed using linear transformations to the to the interval [-1; 1] to improve the efficiency of model training.

4.3 FEATURES PROCESS TRAINING OF GAN-TYPE MODELS

GAN-type models usually suffer from unstable behavior during during training [1, 2, 6]. The main reason for this instability is the loss of synchronization between the generator and the discriminator, one of the networks temporarily temporarily seizes the initiative, which spoils the results of the learning process. This condition is called "mode collapse" [2, 6]. Since this mode is caused by the dominance of one of the networks, the type of collapse can be identified by intermediate learning outcomes [6]. In general, there are two types of collapse, depending on the network that took the initiative.

The first type, "mode collapse", is determined by the dominance of the generator in the in the learning process. The generator begins to create identical images that were highly were highly rated by the discriminator in the previous training iteration.

In this case, the intermediate results are as follows:

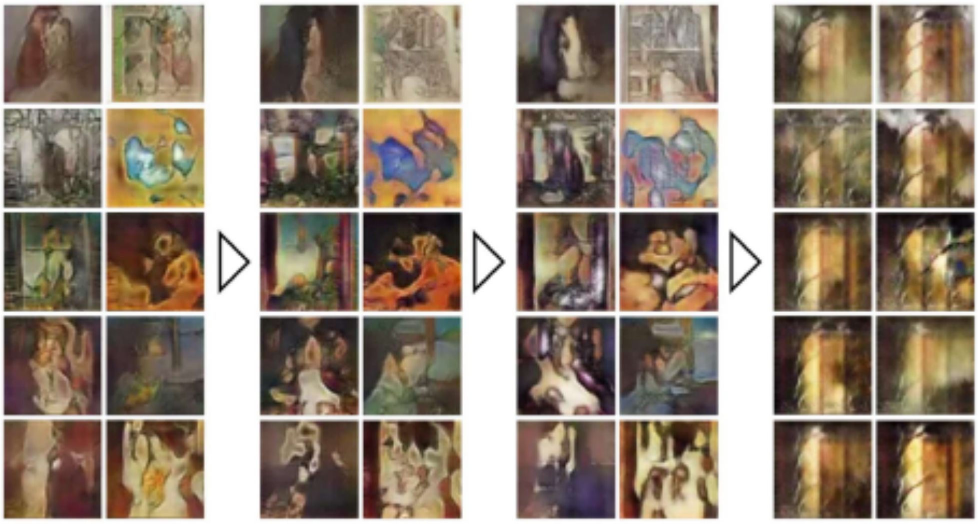


Fig. 4.2 – An example of the result in a collapse due to the dominance of the generator

The second type, "mode collapse," is determined by the dominance of the discriminator in the in the learning process. In this type of collapse, due to the aggressive evaluation of of images by the discriminator, it is difficult for the generator to understand how to improve to improve the current generation results, which leads to a complete loss of of learning progress. In this case, the training results have are as follows:

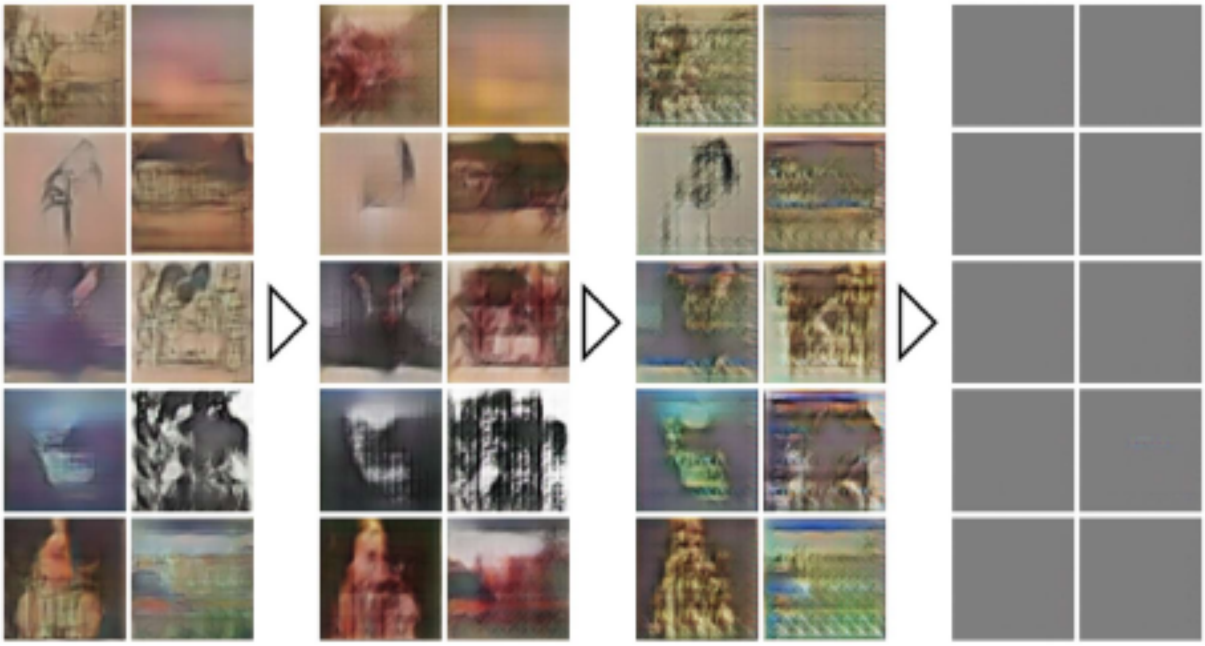


Figure 4.3 – An example of the result in a collapse under dominance discriminator

When analyzing the various modifications, the main idea was to try to overcome the the corresponding collapse mode. For this purpose, a wide range of modifications of the generative GAN model were considered. Most of the trained models showed unsatisfactory results, so only relatively successful modifications were selected for analysis. for analysis.

4.4 MODIFICATIONS OF THE GAN MODEL

In the process of preliminary training and search for modifications of GAN, at least 14 modifications were implemented and trained. Among those considered, most training attempts were unsuccessful. For the analysis, we used three modifications were used for the analysis, for which satisfactory results were obtained, namely: "DCGAN", "DCGAN-WI" and "WGAN-GP" (the names of these modifications are arbitrary and reflect the general idea of model modification and are not are not generally accepted).

When searching for the optimal structure of the modification, we considered a certain list of values of possible hyperparameters of the corresponding generative model of the GAN type. This list is as follows:

Table 4.1

Гіперпараметр	Значення
Розмірність латентного простору для генератора	$[100,\ 128,\ 256]$
Коефіцієнт швидкості навчання	$[1 \cdot 10^{-3},\ 2 \cdot 10^{-4},\ 1 \cdot 10^{-4},\ 1 \cdot 10^{-5}]$
Розмір навчальної вибірки для ітерації навчання	$[64,\ 128,\ 256]$
Функція втрат	$[Binary\ Crossentropy,\ Wasserstein_1\ distance]$
Функція активації	$[Linear,\ ReLU,\ LeakyReLU,\ TanH]$
Оптимізатор	$[SGD,\ Adam]$
Adam β_1 параметр	$[0.0,\ 0.5]$
Adam β_2 параметр	$[0.9,\ 0.999]$
Наявність повнозв'язних шарів	$[False,\ True]$
Вид нормалізації для згорткових шарів	$[None,\ Batch\ Normalization,\ Layer\ Normalization]$
Розмір ядра згортки	$[3,\ 4,\ 5]$
Корекція градієнту	$[None,\ Gradient\ Clipping,\ Gradient\ Penalty]$
Відношення ітерацій навчання моделей	$[1\,/\,1\ ,\ 1\,/\,5\,]$
Початковий розподіл ваг	$[GlorotNormal,\ Normal(0,\ 0.02),\ Uniform(-0.02 \cdot \sqrt{3},\ 0.02 \cdot \sqrt{3})]$
Використання зміщень	$[False,\ True]$

Due to the resource complexity of the process of training generative models, it was decided to transfer the process of pre-training all the models under consideration and calculating the corresponding FID metric to remote servers, namely using the Google Colab service. This service allows to transfer the model training process to the GPU, which significantly speeds up the training process.

To visualize the structure of the generator and discriminator networks of the respective models and their modification, we used the interpretation style used in [3].

4.4.1 TYPE MODEL "DCGAN"

The model, called "DCGAN", is an implementation of the conventional GAN model structure with convolutional layers for the model generator and discriminator [3].

The corresponding model uses only convolutional layers, which should increase the stability of the model [3]. The structure of this model is as follows:

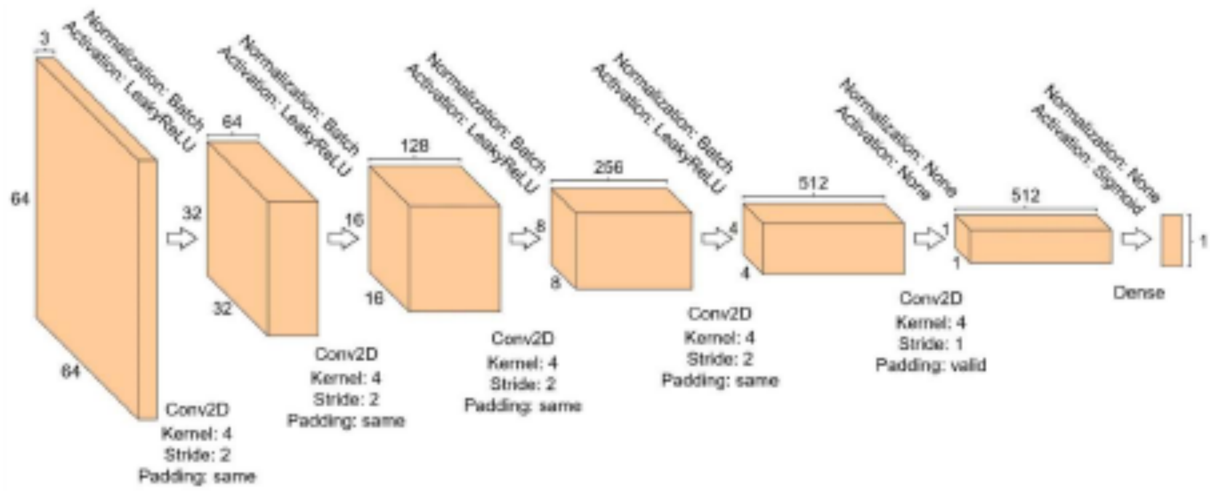


Fig . 4. 4 – “DCGAN” model. Structure of the model discriminator [3]

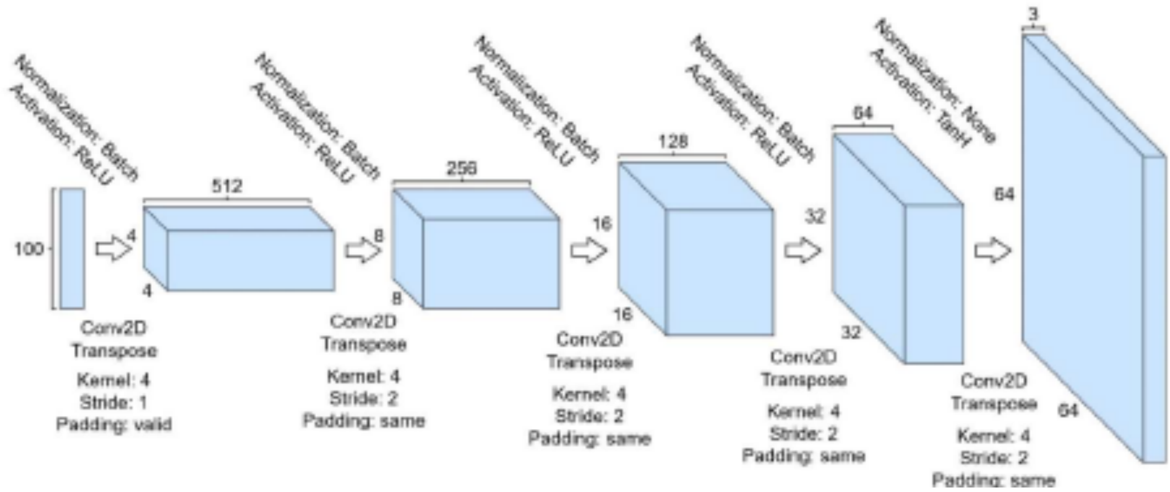


Fig. 4.5 - "DCGAN" model. The structure of the generator model [3]

The following hyperparameters were additionally used for this model:

- Optimizer: Adam ($2 \cdot 10^{-4}$, 0.5, 0.999);
- loss function: Binary Crossentropy;
- activation function: Leaky ReLU: α= 0,2;

- initial values of weights: Glorot Normal;
- use of offsets: False;
- training sample size: 128.

The model was trained for 260 epochs. The estimated training time of training was 280 minutes (Google Colab, GPU). During the training process, the model suffered from constant mode collapse and required constant restarts to continue the training process (on average, every 50 epochs).

The model was saved every 10 epochs of training to allow to restore the model state. After 260 epochs, the value of the FID metric reached value of 102.34405666097047.

4.4.2 "DCGAN-WI" TYPE MODEL

The model called "DCGAN-WI", in general, is a modification of the of the previous model "DCGAN". The main idea of this modification is to increase the size of the kernel of all convolutional layers and the specific values of the initial weights for both networks (discriminator and generator).

Theoretically, the use of odd and larger kernel sizes of convolutional layers and densely distributed initial weights should increase the stability of the of the model training process. The structure of this model is as follows:

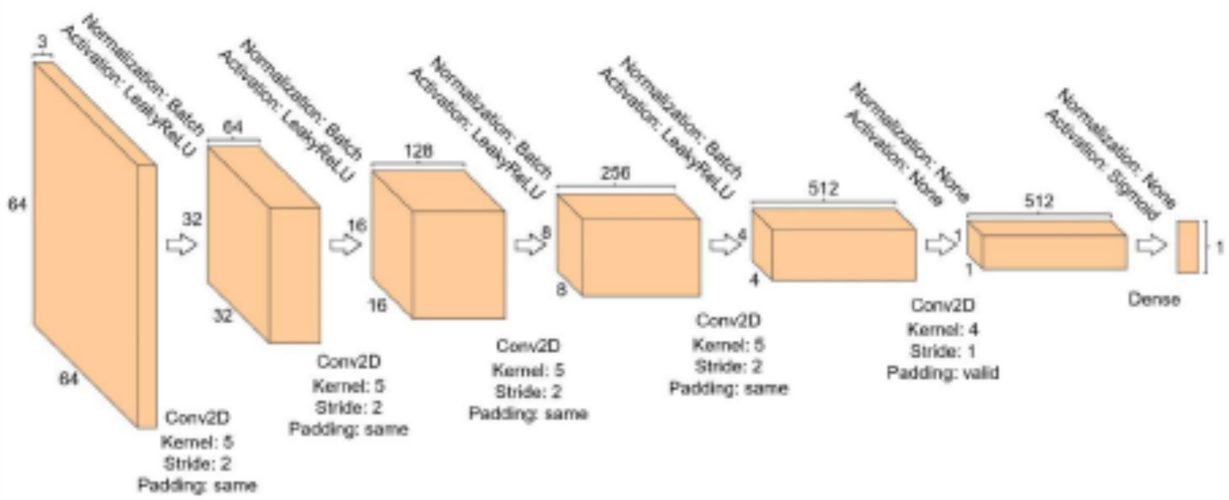


Fig. 4.6 - "DCGAN-WI" model. Structure of the discriminator model [3]

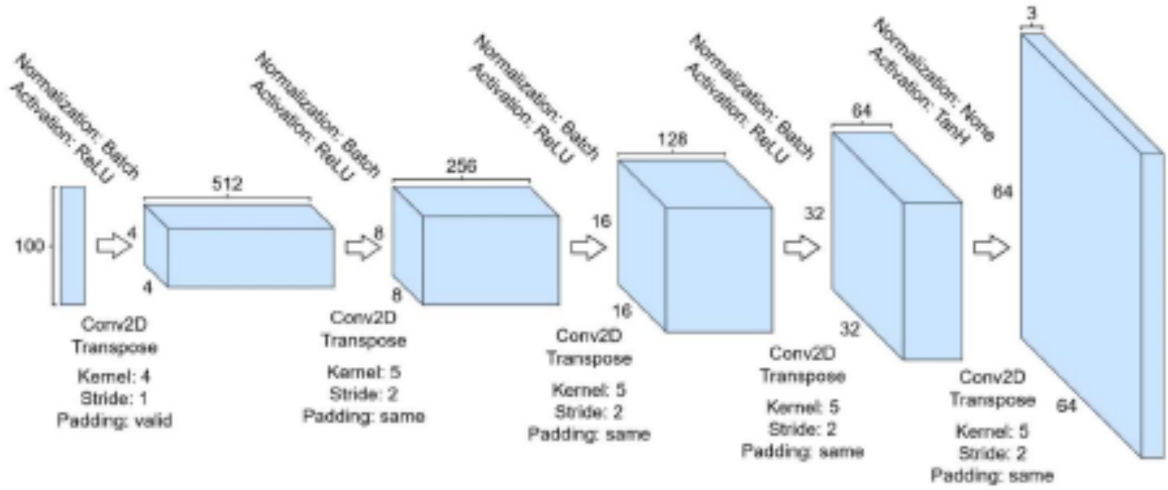


Figure 4.7 - "DCGAN-WI" model. The structure of the generator model [3]

The following hyperparameters were additionally used for this model:

- Optimizer: Adam (2 · 10-4, 0.5, 0.999);
- loss function: Binary Crossentropy;
- activation function: Leaky ReLU: α= 0,2;
- initial values of weights: Glorot Normal;
- use of offsets: False;
- training sample size: 128.

The model was trained for 60 epochs. Reducing the number of epochs is associated with an increase in the computational complexity of this model The estimated time of training of training was 280 minutes (Google Colab, GPU). During the training process, the model suffered from constant mode collapse and required constant restarts to continue the training process (on average, every 50 epochs).

The model was saved every 10 epochs of training to allow to restore the model state. After 260 epochs, the value of the FID metric reached value of 102.34405666097047.

CHAPTER 5
OCCUPATIONAL HEALTH AND SAFETY

5.1 INTRODUCTION

In today's world, the use of computer vision in intelligent image processing systems is becoming a necessary component for the development of various industries, from industry to medicine. Specialized systems that use computer vision play a key role in solving problems of production process automation and improving diagnostic and monitoring systems.

The aim of this research is to develop a system for optimal selection of computer vision methods for efficient image processing. The main focus of the work is not only on improving processing algorithms, but also on studying the aspects of labor protection during the development and implementation of such systems. Ensuring safety and optimizing the use of intelligent technologies are critical conditions for their successful implementation in various industries.

This section will provide a detailed analysis of the occupational health and safety aspects associated with the use of computer vision methods for image processing. Consideration of risks, identification of safety measures, and consideration of modern standards will be an integral part of our research aimed at creating efficient and safe intelligent systems.

5.2 ANALYSIS OF WORKING CONDITIONS

The facilities where computers are to be installed and operated must comply with the building design documentation approved by the authorized state authorities. In addition, the employer must comply with sanitary standards for lighting, microclimate parameters (temperature, humidity), vibration, sound, fire resistance, and electromagnetic field characteristics. The defined indicators of sanitary standards are specified in the State Sanitary Rules and Norms for Working with Visual Display Terminals of Electronic Computers according to SanPIN 3.3.2.007-98, approved by Resolution No. 7 of the Chief State Sanitary Doctor of Ukraine dated December 10, 1998. These rules apply to work with visual display terminals of various types used in electronic computers.

For example, employers are prohibited from installing computers in basements. In order to avoid accidents and short circuits, it is prohibited to perform work that requires the use of excessively humid technological processes near the premises where the computer is operating (above or below them). The relevant room must be equipped with central or individual heating, air conditioning, or ventilation systems. However, when installing such systems, make sure that heating batteries, water pipes, ventilation cables, etc. are securely hidden under protective shields to prevent possible exposure of the employee to voltage.

All rooms where workplaces for employees working on computers will be equipped should have elements of natural and artificial lighting. Windows should be equipped with easily adjustable blinds or curtains that allow employees to adjust the level of light in the room. The location of computers in the room should be such that light falls on the monitor screens from the south or northeast.

To ensure the maximum level of safety and labor protection, the room should have first aid kits, automatic fire alarm systems and fire extinguishers. In a room with 5 or more computers, a service switch should be installed in a visible place to allow for a complete power outage if necessary.

Regarding personal workstations, SanPIN 3.3.2.007-98 stipulates that a workstation should have dimensions of 4.5×2.5×3 meters (length, width, height), an area of 11.25 m2 and a volume of 33.75 m3. At least 6.0 square meters of space and at least 20.0 cubic meters of volume must be allocated for one workstation using a PC. If necessary, workstations can be divided by partitions up to 2 meters high. When determining the size of the room and workplace, you should take into account the furniture and equipment that will be in the room. Employees' desks may be equipped with devices for placing technical devices and places for storing documents, provided that this does not interfere with the visibility of the screen and does not disturb the employee. In case of high noise or vibration, the employer must provide anti-vibration mats.

Employees' chairs should be lift-and-swivel, adjustable in height and provide support for the back and spine. Wet cleaning and dusting of workstations and computer screens should be performed daily. The employer prohibits repairing and maintaining computers at the employee's workplace, repairing or attempting to configure the computer without specialists, and storing unnecessary documents and items at the workplace.

In order to preserve the health of employees, each person must undergo a medical examination upon hiring and regularly at least once every 2 years. This includes doctors of such specialties as general practitioners, neurologists and ophthalmologists. Mandatory rest breaks for employees, lasting 10-15 minutes every hour or two, depending on the complexity of the work, should also be clearly established. Working hours of continuous computer work should be limited to 4 hours. It is recommended to allocate a separate room for employees to rest in order to relieve the nervous and emotional stress that may arise when working with a computer.

5.3 DEVELOPING OCCUPATIONAL HEALTH AND SAFETY MEASURES

Occupational health and safety in the workplace is an important component of ensuring the safety of employees and achieving high research results. The following are the health and safety measures that should be implemented in the room:

Table 5.1 - Measures and equipment to prevent and protect against electrical injuries

№	Categories of protection measures	Type of event	Selection criterion
1	Technical	- installation of protective grounds; - use of protective separation of power grids; - maintaining a dry, dust-free room with a humidity of no more than 75%;	Avoid breakdowns, current leaks and prevent contact with live parts.

		- applying an insulating coating to the floor; - ensuring complete and reliable insulation of live parts.	
2	Organizational	- Conducting the necessary briefings on electrical safety rules.	Ensure that personnel have knowledge of the correct and safe operation of the devices.
3	Mode	- checking and troubleshooting the devices only when they are in a disconnected state.	Preventing contact with live elements
4	Operational	regular monitoring of the devices (surface temperature, charge level); regular maintenance; timely replacement of detected damaged elements. -	Ensuring safe work with the object, reducing the risk of contact with electric current, preventing an explosion due to malfunctioning elements.

5.4. FIRE SAFETY

Sources of fire hazard in the laboratory: laptop, wooden desk. The following combustibles are present in the office:

- Wood.
- Fabric (material of blinds, curtains).
- Circuit boards (computer).

Tables 5.2-5.4 define the sources of fire, the category of the room, the class zone and the class of possible fire, and also provide a list of means and measures of protection against explosion and fire and ways to overcome the fire situation in case of its occurrence.

Table 5.2 - Sources of fire hazards

№	Name	Source of danger	Causes of hazards	Consequences of the danger
1	Laptop	Power supply, live parts	Short circuit	Fire and burns, which can lead to serious health consequences for employees and patients. In addition, there may be damage to equipment, tools and personal belongings of employees.
2	Materials and substances prone to ignition	Fire of materials	External fire	

Table 5.3 - Explosion and fire hazard characteristics

№	Name	Meaning	Description
1	Fire class	A,E	Combustion due to the ignition of solids and live electrical installations

2	Fire safety class of the room area	Class II-IIIa	Combustible liquids, solid combustible and flammable substances, materials capable of reacting with water, air oxygen or each other burn only if the premises in which they are located or used do not belong to categories A and B .
3	Fire hazard category	B	

Table 5.4 - Means and measures of protection against fire hazard

№	Events	Realization	Selection criterion
1	Technical	Placement of a VP-5 powder fire extinguisher in the room, installation of air conditioning, and ensuring the availability of a fire hydrant and hose in the corridor.	Eliminating fires and primary sources of ignition, maintaining optimal temperature and humidity in the office.
2	Organizational	Conducting fire drills and fire safety briefings. Create and post evacuation plans in a prominent place.	Training on fire safety, providing an algorithm of actions to prevent human casualties, providing accessible information for quick and safe evacuation.
3	Mode	Prohibiting the use of open fire in the office, preventing unauthorized persons from entering the office, and preventing the	Prevention of fire due to unforeseen factors.

		presence of explosive objects in the office.	
4	Operational	Regular monitoring and timely inspection of the equipment, repair and replacement if necessary.	Prevention of fires that may occur as a result of a technical malfunction of the equipment.

5.5 VERIFICATION CALCULATION OF ARTIFICIAL LIGHTING OF INDUSTRIAL OR OFFICE PREMISES.

The verification calculation for artificial lighting of a workspace will be performed according to the formula:

$$F = \frac{E * S * K * Z}{N * n * \eta} \tag{5.1}$$

F is the luminous flux of the lamp, lm;

E - minimum standardized illumination, lux, for a given category of visual work;

S is the area of the room, m2 ;

N - number of luminaires; n - number of lamps in each luminaire;

K is a safety factor;

Z is the coefficient of unevenness of lighting;

η - luminous flux utilization factor.

Let's calculate the luminous flux coefficient using the formula:

$$\boldsymbol{i = \frac{A * B}{(A + B) * h} = \frac{4.5 * 2.5}{(4.5 + 2.5) * 3} = 0.53 \approx 0.5}$$

The lighting belongs to the first group, and according to the formula, the value of the flow coefficient is 22. After that, we proceed to calculate the luminous flux of the lamp:

$$\boldsymbol{F = \frac{E * S * K * Z}{N * n * \eta} = \frac{300 * 11{,}25 * 5 * 1{,}5}{2 * 4 * 22} = 143 lm}$$

To meet the requirements of artificial lighting in an office with an area of 11.25 m2, it is enough to install 2 luminaires with 4 lamps each.

Conclusions of chapter

This section discusses the layout of the workspace and analyzes possible physical and biological risks. During the analysis of working conditions in the office, it was found that the area and volume of the premises per employee meet the regulatory requirements, and the workplaces are properly organized. The office meets the requirements of electrical safety, fire safety and is equipped to ensure biological safety. In general, the room meets the general safety requirements applicable in this context.

CHAPTER 6

EQUIPMENT LIFE CYCLE ASSESSMENT

6.1EQUIPMENT LIFE CYCLE

Life cycle assessment of equipment is an important tool for making decisions in production and consumption. It allows you to take into account not only the cost and efficiency of equipment at the stage of its use, but also the environmental impact of all stages of the life cycle - from resource extraction to disposal.

In a world where more and more organizations are striving for sustainable development, life cycle assessment of equipment is becoming an integral part of the decision-making process. It allows you to determine the environmental burden associated with the production and use of specific equipment, as well as identify opportunities for optimization and improvement.

Life cycle assessment or LCA (also known as life cycle analysis) is a methodology for assessing the environmental impacts associated with all stages of the life cycle of a commercial product, process or service.

For example, in the case of an industrial product, the environmental impact is assessed from the extraction and processing of raw materials (cradle to grave), through the production, distribution and use of the product for processing or final disposal of the materials it is made of (disposal).

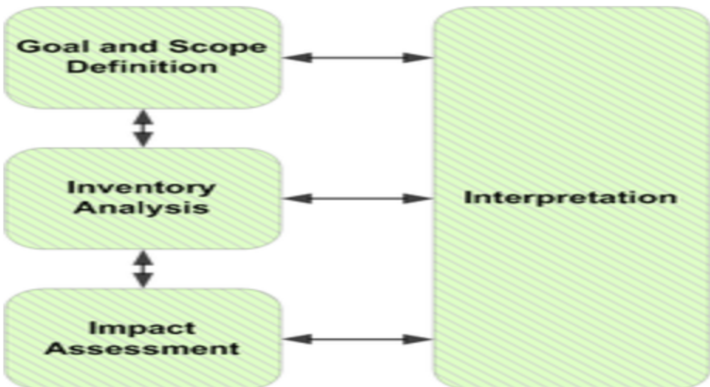


Fig. 6.1: Illustration of the general stages of life cycle assessment as described in ISO 14040

Life cycle assessment (LCA) is one of the tools in the section "Visual Data and Information (4.6.6 Visual Data And Information)" in PMBOK 7, and is defined there as:

"This assessment is a tool used to determine the overall environmental impact of a product, process, or system. It covers all aspects of the production of the project deliverable, from the origin of the materials used in it to its distribution and final disposal."

Widely recognized procedures for conducting LCA are included in the International Organization for Standardization (ISO) 14000 series of environmental management standards, including ISO 14040 and ISO 14044.

ISO 14040 defines the "principles and structure" of the standard, while ISO 14044 provides a general description of the "requirements and guidelines". In general, ISO 14040 was written for a management audience, while ISO 14044 was written for practitioners. The introduction to ISO 14040 defines life cycle assessment as follows: LCA examines the environmental aspects and potential environmental impacts throughout the entire life cycle of a product (i.e., from cradle to grave), from raw material procurement to production, use and disposal. The general categories of environmental impacts that need to be considered include resource use, human health, and ecological impacts.

According to the EPA's National Risk Management Research Laboratory, "LCA is a method of assessing the environmental aspects and potential impacts associated with a product, process, or service by using:

1. Compiling an inventory of relevant energy and material resources and environmental emissions
2. Assessment of potential environmental impacts associated with certain resources and emissions
3. Interpretation of the results to help you make a more informed decision.

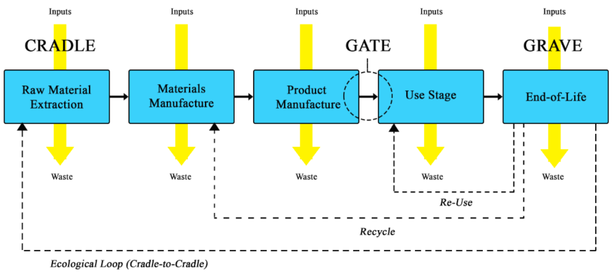


Fig. 6.2. An example of a life cycle assessment (LCA) stage diagram

Thus, it is a method of environmental assessment related to all stages of the product life cycle, from raw material extraction to material processing, production, distribution, use, repair and maintenance, as well as disposal or recycling.

6.2 ENVIRONMENTAL IMPACT OF EACH STAGE OF THE COMPUTER LIFE CYCLE

Life cycle assessment of equipment is an important aspect of making decisions about its use. One of the key aspects of the assessment is the analysis of the environmental impact of each stage of the life cycle of equipment such as a computer.

Fig. 6.3 Phases or stages of the computer life cycle

The first stage of a computer's life cycle is the extraction and production of raw materials. Computers require large amounts of rare and useful metals, such as gold, silver, and platinum. Mining these metals often involves high levels of environmental pollution and health hazards for workers. In addition, the process of manufacturing computer components consumes a large amount of energy and uses various chemicals that can be hazardous to the environment.

To make the components that will make up a computer, you will need the materials mentioned above, for example:

Copper is commonly used as a conductor of electricity, so it is commonly used in both the computer's motherboard and its cabling. In addition, microchips, integrated circuits, and heat sinks are also made from this material.

Silicon, in turn, is also one of the most important, as it is a semiconductor that supports high temperatures. It is one of the most common materials and is also used for computer microchips and integrated circuits.

Plastic will be the most commonly used computer material because it is used in most components. Among them, one of the most commonly used is acrynothrile-butadiene-styrene thermoplastic.

Usually, each component has its own company or company, for example, one may be responsible for manufacturing motherboards; while another manufactures processors.

After each company is responsible for developing the components, they are sent to the company that is responsible for assembling and designing the computer. The design itself takes two to three years to complete.

When researching this stage of the computer life cycle, the following disturbing data was found:

- Engineers and manufacturers are not very aware of the harmfulness of materials in the environment; there is no separate "toxicological counseling" available.
- Women working in component production are more likely to have miscarriages, in particular, 40% more often than other workers.
- Water use is one of the weaknesses of tech companies, as they consume the most water (more than a trillion gallons in semiconductor production annually in the US) and, in turn, need to invest heavily in cleaning up contaminated oils and water.

This means that companies operating in the tech sector should be aware of and find a way to not waste such an important natural resource as water, and not worry about the damage caused by the materials used to its employees and the environment.

Likewise, the regulatory authorities of each country should keep an eye on them, because while technology has been extremely beneficial to humanity, it also has its drawbacks, and a way to strike a balance must be found, as we are the only ones with the planet and we must take care of it.

The second stage is the transportation and delivery of computer equipment to the end user. Sending computers around the world requires the use of transportation, which leads to greenhouse gas emissions and air pollution. In addition, the packaging materials used to protect computer equipment during transportation can be a negative factor for the environment.

The third stage is the use of the computer. During operation, a computer consumes electricity, which leads to greenhouse gas emissions and air pollution. In addition, some computer components contain hazardous chemicals, such as lead and mercury, which can be released if improperly disposed of or damaged.

The fourth stage is the disposal or recycling of computer equipment. Dumping old and faulty equipment in a landfill causes problems with decomposition and contamination of soil and groundwater. In addition, many computer parts can be recycled and reused. However, the recycling process can have a negative impact on the environment, especially if it is not done properly.

Assessing the environmental impact of each of these stages of the life cycle of computer equipment allows you to evaluate its environmental sustainability and take steps to improve its sustainability. For example, energy-efficient materials and technologies can be used in the production of computers, and recycling and disposal of equipment can be encouraged. In addition, developing more durable components can reduce the need for new equipment and reduce waste.

Overall, environmental assessment of each stage of the computer hardware life cycle is now an integral part of sustainable development. Understanding these factors helps us to make informed decisions about the selection and use of computer equipment to minimize negative environmental impact and create a more environmentally sustainable future infrastructure.

6.3 REQUIREMENTS FOR COMPUTER DISPOSAL

Computers are an integral part of modern society and are used in various fields of activity. However, like any other equipment, they have a limited life cycle and eventually become obsolete or fail. In this regard, there is a need to properly dispose of computers.

The requirements for computer recycling are determined by both the efficient use of resources and environmental safety. The following factors should be taken into account when choosing a disposal method:

1. Separation of components: A computer consists of many different components, such as the motherboard, processor, hard disk, etc. When you dispose of your computer, you should separate these components into separate groups to make more efficient use of resources.

2. Prioritize: Some computer components can be reused or recycled more efficiently than others. For example, recycling metal components may be more efficient than recycling plastic parts. Therefore, it is necessary to prioritize the recycling process.

3. Secure data deletion: Computers store a large amount of personal and sensitive data. When recycling, you should ensure that all data is safely deleted from the hard drive or use special programs to completely format the information.

4. Environmental aspect: Dispose of your computer with minimal negative impact on the environment. For this purpose, you can use recycling methods that minimize emissions of harmful substances and prevent soil and water pollution.

5. Compliance with the law: Different countries have rules and laws regarding the disposal of electronic equipment. When choosing a disposal method, you need to consider the requirements and standards to avoid fines or legal problems.

Recycling computers is an important step in their life cycle. Choosing the right recycling method allows you to use resources efficiently and minimize the negative impact on the environment. The requirements for computer recycling include separation of components, prioritization, secure data deletion, environmental aspects, and legal compliance. When choosing a recycling method, you should take these factors into account to ensure that the recycling process is as efficient and safe as possible.

Conclusions of the chapter

This chapter has considered the environmental impact of each stage of the computer life cycle. It has been established that the greatest environmental impact is caused by the production of computers, which consumes a large amount of energy and uses hazardous chemicals. The use of computers also leads to energy consumption and greenhouse gas emissions. Disposing of computers can have a negative impact on the environment if it is not done properly.

To minimize the negative impact on the environment, measures must be taken at all stages of the computer life cycle. In production, energy-efficient materials and technologies should be used, as well as safe methods of extraction and processing of raw materials. During use, the recycling and disposal of computers should be encouraged. Disposal of computers should be carried out with minimal negative impact on the environment, using recycling methods that minimize emissions of harmful substances and prevent soil and water pollution.

’

CONCLUSION

This diploma work surveys various GANs architectures that have been used for addressing the different imbalance problems in computer vision tasks. In this survey, we provided detailed background information on deep generative models and GAN variants from the architecture, algorithm, and training tricks perspective. In order to present a clear roadmap of various imbalance problems in computer vision tasks, we introduced taxonomy of the imbalance problems. Following the proposed taxonomy, we discussed each type of problems separately in detail and presented the GANs based solutions with important features of each approach and their architectures. We focused mainly on the real-world applications where GAN based synthetic images are used to alleviate class imbalance. In addition to the thorough discussion on the imbalance problems and their solutions, we addressed many open issues that are crucial for computer vision applications.

Synthetic but realistic images generated using the methods discussed in this survey have the potential to mitigate the class imbalance problem while preserving the extrinsic distribution. Many of the methods surveyed in this paper tackled the highly complex imbalances by combining GANs architecture with different other deep learning frameworks. Specifically, the use of autoencoders with GANs has offered an effective way to perform feature space manipulations instead of complex pixel space operations.

Synthetic images generated by GANs cannot be used as the complete replacement for real datasets. However, the blend of real and GANs generated images have enormous potential to increase the performance of the deep learning model. Looking into the future, GAN-related research in image as well as non-image data domains to address the problem of imbalances and limited training dataset would continue to expand. We conclude that the future of GANs is promising and there are clearly a lot of opportunities for further research and applications in many fields.

SOURCES

1. Ian Goodfellow NIPS 2016 Tutorial : Generative Adversarial Networks arXiv preprint arXiv:1701.00160, 2016.

2. Martin Arjovsky , Soumith Chintala , Leon Bottou Wasserstein GAN arXiv preprint arXiv:1701.07875, 2017.

3. Is lagyhaan Gulrajani , Faruk Ahmed , Martin Arjovsky , Vincent Dumoulin , Aaron Courville Improved Training of Wasserstein GANs arXiv preprint arXiv:1704.00028, 2017.

4. Nugraha BT, Su SF, Fahmizal. Towards self-driving car using convolutional neural network and road lane detector. Proceedings of the 2nd International Conference on Automation, Cognitive Science, Optics, Micro ElectroMechanical System, and Information Technology, ICACOMIT 2017. 2017;2018-Janua:65

5. Yadav SS, Jadhav SM. Deep convolutional neural network based medical image classification for disease diagnosis.J Big Data. 2019. https ://doi.org/10.1186/s4053 7-019-027

6. Gutierrez A, Ansuategi A, Susperregi L, Tubío C, Rankić I, Lenža L. A Benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. J Sensors. 2019. https ://doi.org/10.1155/2019/52194 71.

7. Santos L, Santos FN, Oliveira PM, Shinde P. Deep learning applications in agriculture: a short review. Advances in intelligent systems and computing. Fourth Ibe. 2020. https ://doi.org/10.1007/978-3-030-35990 -4_12.

8. Wang T, Chen Y, Qiao M, Snoussi H. A fast and robust convolutional neural network-based defect detection model in product quality control. Int J Adv Manufactur Technol. 2018;94:3465–71.

9. Hashemi M. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs interpolation. J Big Data. 2019. https ://doi.org/10.1186/s4053 7-019-0263-7.