ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Підходи до побудови нейронних мереж для зображень»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 – Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Сергій ОВДІЄНКО*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КІД-41
Сергій ОВДІЄНКО
Ім'я, ПРІЗВИЩЕ

Керівник: Ігор БУЧЕНКО
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 – Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
_____ Наталія ЛАЩЕВСЬКА
_____
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Овдієнко Сергій Дмитрович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Підходи до побудови нейронних мереж для зображень керівник кваліфікаційної роботи Ігор БУЧЕНКО,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56
2. Строк подання кваліфікаційної роботи «5» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література; основи глибокого навчання; теорія нейронних мереж; комп'ютерний зір; архітектури CNN, GAN, Transformer; методи навчання та оптимізації; сервіси Stable Diffusion, ClipDrop, Freepik AI;
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Огляд архітектур нейронних мереж для обробки зображень
2. Методи навчання та оптимізації глибоких моделей
3. Класифікація, сегментація та генерація зображень за допомогою нейронних мереж
4. Порівняльний аналіз результатів на основі різних архітектур
5. Приклад комплексного використання моделей у практичному завданні
6. Висновки
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «24» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.02-24.03.2025	Виконано
2	Обробка матеріалу	02.03-09.04.2025	Виконано
3	Розробка теоретичної частини	09.04-12.04.2025	Виконано
4	Проведення експериментів	12.04-16.04.2025	Виконано
5	Висновки за результатами аналізу	13.04-17.05.2025	Виконано
6	Оформлення пояснювальної записки	17.05-20.05.2025	Виконано
7	Розробка презентації	20.05-25.05.2025	Виконано
8	Передзахист	27.05-29.05.2025	Виконано
9	Здача в деканат	02.06-05.06.2025	Виконано

Здобувач вищої освіти _____ Сергій ОВДІЄНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ігор БУЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 63 стор., 46 рис., 8 табл., 30 джерел.

Мета роботи – Дослідити підходи до побудови, навчання та практичного застосування нейронних мереж для обробки зображень, проаналізувати сучасні архітектури, а також реалізувати експериментальні приклади з використанням популярних AI-сервісів

Об'єкт дослідження – Нейронні мережі глибинного навчання, що використовуються для обробки, аналізу та генерації зображень

Предмет дослідження – Архітектурні підходи (CNN, GAN, Transformer, Diffusion), методи навчання та інструменти застосування нейромереж у задачах комп'ютерного зору

Короткий зміст роботи: У роботі проаналізовано ключові архітектури нейронних мереж, що застосовуються для класифікації, сегментації та генерації зображень. Особливу увагу приділено сучасним моделям: дифузійним (Stable Diffusion), генеративно-змагальним (Artbreeder) та трансформерним (Segment Anything Model), які були протестовані в однакових умовах. Проведено експерименти, що продемонстрували переваги та недоліки кожного підходу, зокрема в задачах стилізації, обробки обличчя, об'єднання сцен та локального редагування. У рамках практичного дослідження створено приклади повноцінної обробки зображень із комбінуванням моделей, що показало доцільність використання гібридного підходу в реальних дизайнерських задачах. На основі аналізу сформульовано рекомендації щодо вибору моделей для різних типів завдань комп'ютерного зору.

КЛЮЧОВІ СЛОВА: НЕЙРОННА МЕРЕЖА, ОБРОБКА ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, STABLE DIFFUSION, GAN, ГЕНЕРАЦІЯ, КЛАСИФІКАЦІЯ, СЕГМЕНТАЦІЯ.

ЗМІСТ

ВСТУП **8**
1 ОГЛЯД АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ **9**
1.1 Основи глибинного навчання 9
1.2 Класичні архітектури (LeNet, AlexNet, VGG) 10
1.3 Сучасні архітектури (ResNet, Inception, EfficientNet) 15
1.4 Диференціація CNN, GAN, Vision Transformers 19
1.5 Порівняння архітектур по точності, швидкості, обчислювальній складності 21
2 МЕТОДИ НАВЧАННЯ ТА ОПТИМІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ **23**
2.1 Навчання з учителем, без учителя, transfer learning 23
2.2 Функції втрат, оптимізатори (SGD, Adam, RMSprop) 25
2.3 Аугментація даних та регуляризація 29
2.4 Пояснення ефекту перенавчання 32
2.5 Підходи до підвищення точності та стабільності моделей 33
3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ **37**
3.1 Класифікація, сегментація, генерація зображень 37
3.2 Експериментальне дослідження: порівняння архітектур у практиці 48
3.3 Аналіз отриманих результатів 57
3.4 Поєднання нейромережевих моделей для реалізації дизайнерського завдання 60
ВИСНОВКИ **69**
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **71**
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) **71**

ВСТУП

У сучасному цифровому світі обробка зображень стала важливою складовою численних сфер діяльності – від медицини до дизайну, від автомобільної промисловості до соціальних мереж. Зі зростанням обсягу візуальної інформації виникла потреба в автоматизованих системах, здатних розпізнавати, класифікувати та генерувати зображення з високою точністю. Одним із найефективніших підходів до реалізації таких систем є використання штучних нейронних мереж.

Нейронні мережі, особливо ті, що належать до класу глибокого навчання (deep learning), демонструють значні успіхи в розв’язанні задач комп’ютерного зору. Відомі архітектури, такі як LeNet, AlexNet, VGG, ResNet, EfficientNet та новітні трансформерні моделі, стали основою сучасних систем машинного зору. Разом із розвитком архітектур активно удосконалюються й методи навчання та оптимізації моделей, що дає змогу досягати ще кращих результатів.

Особливої популярності набули генеративні моделі, серед яких варто відзначити GAN та дифузійні нейронні мережі (Stable Diffusion). Такі моделі дозволяють створювати нові зображення на основі текстових описів або покращувати якість уже існуючих, що широко застосовується, зокрема, у сфері цифрового дизайну, включаючи сервіси на кшталт Freepik.

1 ОГЛЯД АРХІТЕКТУР НЕЙРОННИХ МЕРЕЖ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ

1.1 Основи глибинного навчання

Глибинне навчання (deep learning) – це підхід у сфері штучного інтелекту, що базується на використанні багаторівневих нейронних мереж, які автоматично витягують ознаки з даних. В обробці зображень найбільше застосування отримали згорткові нейронні мережі (Convolutional Neural Networks, CNN), які дозволяють виявляти локальні візуальні шаблони в піксельній структурі зображення[1][2].

Перевага CNN полягає в зменшенні кількості параметрів завдяки використанню згорток і підвибірок (pooling). Ці мережі здатні навчатися виявляти контури, текстури, форми, об’єкти на різних рівнях абстракції[3].

Загальна структура штучної нейронної мережі, як правило, містить три типи шарів: вхідний, приховані та вихідний. На рисунку 1.1 зображено спрощену модель такої мережі.

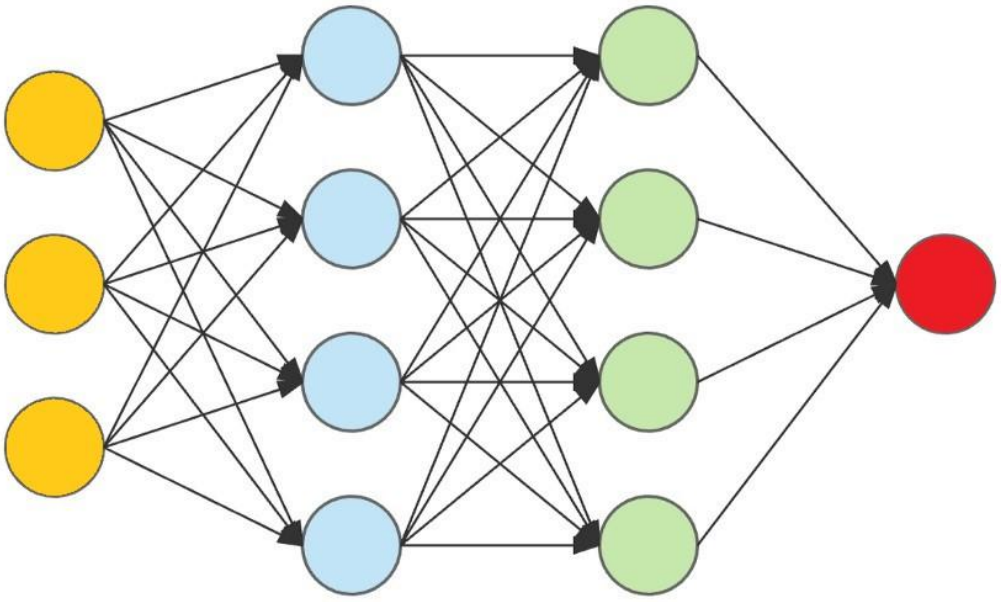


Рисунок 1.1 – Схематична структура згорткової нейронної мережі (CNN)

На рисунку 1.1 жовтий шар представляє вхідні дані – це пікселі зображення або попередньо оброблені вектори ознак[27]. Синій і зелений шари – це приховані шари, де відбувається витягування ознак і трансформація інформації через ваги та активаційні функції. Нарешті, червоний вихідний шар відображає результат обробки: класифікацію зображення, ймовірнісну оцінку або інші типи відповідей моделі[10].

Таке представлення дає змогу візуально зрозуміти, як дані проходять через мережу, набуваючи дедалі більшої абстрактності на кожному рівні.

1.2 Класичні архітектури (LeNet, AlexNet, VGG)

LeNet-5 – одна з перших згорткових нейронних мереж (Convolutional Neural Networks, CNN), яка була запропонована Яном ЛеКуном у 1998 році. Модель була розроблена для задачі автоматичного розпізнавання рукописних цифр, зокрема для використання в банківській сфері при обробці чеків. LeNet-5 стала першою реальною демонстрацією ефективності згорткових шарів в обробці візуальних даних і стала основою для подальшого розвитку глибинного навчання.

Архітектура LeNet-5 орієнтована на обробку зображень розміром 32×32 пікселі (на відміну від стандартного 28×28 у датасеті MNIST), що дає можливість краще аналізувати просторові зв’язки, особливо по краях. Модель має чітко структуровану послідовність згорткових, pooling та повнозв’язних шарів і складається з 7 навчуваних рівнів.

Структура LeNet-5. Мережа складається з чергування згорткових і pooling-шарів, за якими слідують повнозв’язні шари. Вхідне зображення розміром 32×32 пікселі проходить через два згорткові шари з pooling, один умовно повнозв’язний згортковий шар, шар повного з’єднання та вихідний класифікатор на 10 класів. Загальна кількість параметрів становить приблизно 60 тисяч. У мережі застосовуються функції активації tanh і average pooling[11].

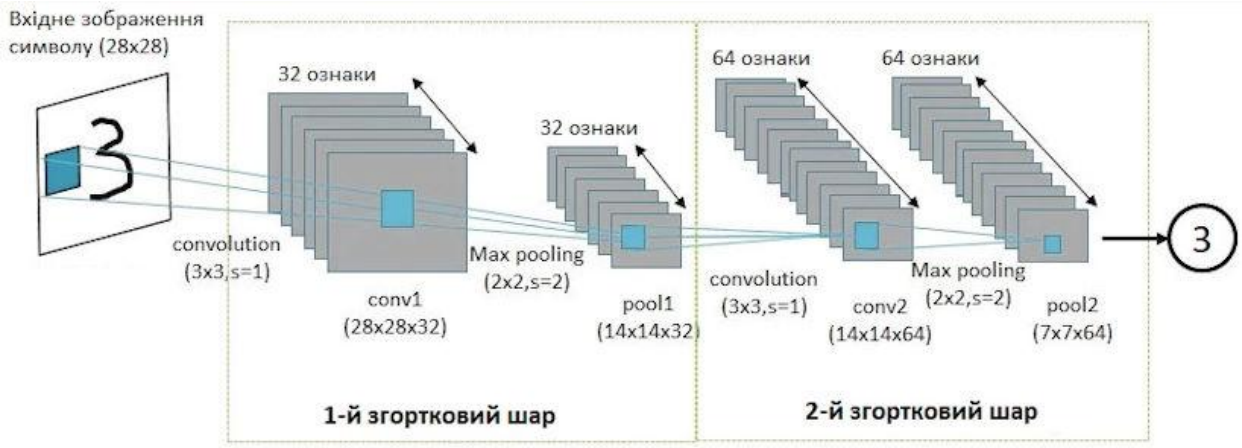


Рисунок 1.2 – Архітектура LeNet-5

На рисунку 1.2 представлено повну структуру моделі LeNet-5. Кожен рівень зображено з урахуванням розмірів вхідного зображення, кількості фільтрів та результатів обробки. Зліва розташовано вхідне зображення, далі – шари згортки та субдискретизації, а наприкінці – повнозв'язні шари з вихідною класифікацією.

Ключові особливості LeNet-5:

– *локальність зв'язків*. Кожен нейрон з'єднаний лише з невеликою частиною нейронів попереднього шару, що дозволяє виявляти локальні просторові патерни;

– *субдискретизація*. Pooling дозволяє зменшити розмірність і підвищити стійкість до зсувів і шуму;

– *ієрархічне витягування ознак*. Мережа послідовно формує більш абстрактні ознаки – від простих контурів до складних форм;

– *обмежена кількість параметрів*. Близько 60 тисяч, що робить її придатною для обчислень навіть на слабких системах.

Значення архітектури LeNet-5. Попри свій вік, LeNet-5 залишається базовою моделлю для вивчення принципів згорткових нейронних мереж. Вона застосовується в навчальних курсах, симуляторах та прикладних задачах з обмеженим обсягом даних. Архітектура заклала фундаментальні принципи, які збереглися в більшості сучасних CNN, зокрема ResNet, EfficientNet, VGG та інших.

AlexNet – це згорткова нейронна мережа, розроблена Алексом Крижевським, Іллею Суцкевером та Джеффрі Гінтоном, яка стала визначальним етапом у

розвитку глибинного навчання. У 2012 році ця модель здобула перемогу в змаганні ImageNet Large Scale Visual Recognition Challenge (ILSVRC), суттєво зменшивши помилку класифікації (top-5 error) з 26 % до 15 %. Її успіх продемонстрував, що використання глибоких нейронних мереж у поєднанні з потужними графічними процесорами (GPU) забезпечує нову якість в обробці зображень.

Модель AlexNet вважається модернізацією попередніх CNN, таких як LeNet-5, однак її глибина, масштаб та технічні інновації були безпрецедентними для свого часу.

Структура AlexNet. Архітектура складається з п’яти згорткових шарів із проміжним застосуванням операцій pooling, після яких йдуть три повнозв’язні шари. На вхід подається зображення розміром 227×227×3. На останньому шарі мережа здійснює класифікацію за 1000 класами. Для активації використовується функція ReLU, а також застосовуються методи нормалізації, dropout та data augmentation. Модель навчалась на двох графічних процесорах, що дозволило обробляти великі обсяги даних у прийнятні строки.

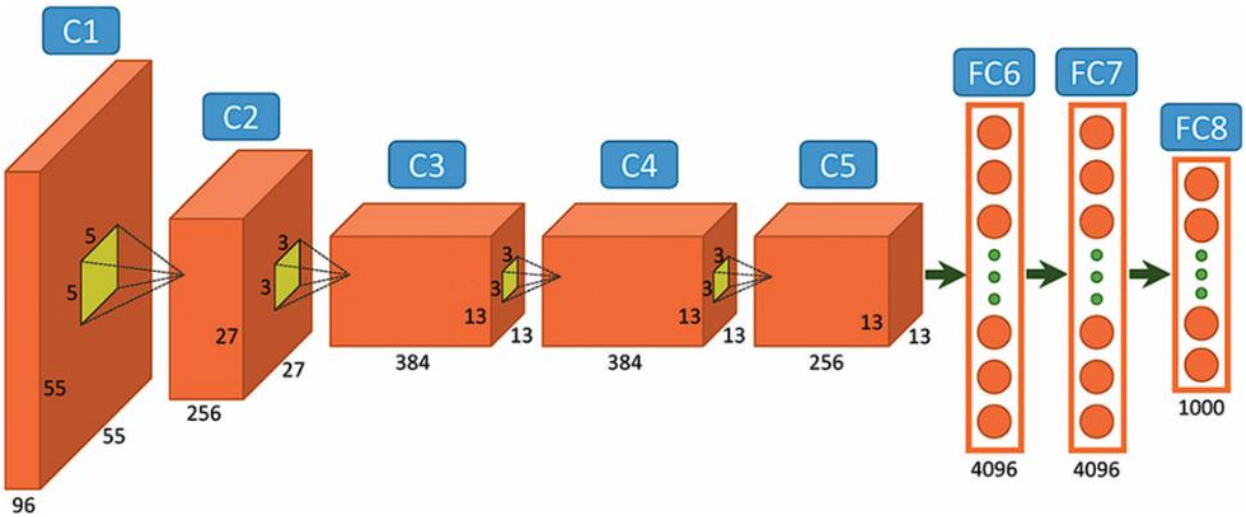


Рисунок 1.3 – Архітектура AlexNet

На рисунку 1.3 показано архітектуру AlexNet. Кожен згортковий шар візуалізовано у вигляді тривимірних блоків, що демонструють просторове зменшення зображення та збільшення кількості каналів. Повнозв’язні шари

зображено як повністю зв’язані нейронні масиви, які виконують фінальну класифікацію.

Ключові особливості AlexNet:

– *використання ReLU*. функція активації Rectified Linear Unit (ReLU) значно прискорює навчання та запобігає проблемі затухаючих градієнтів, яка виникала при використанні sigmoid або tanh;

– *dropout у повнозв’язних шарах*. Випадкове «вимикання» частини нейронів під час навчання допомагає зменшити перенавчання;

– *data augmentation*. Застосовуються дзеркальні відображення, вирізання та випадкове масштабування, що дозволяє штучно збільшити розмір тренувального набору;

– *розділення обчислень між двома GPU*. Оскільки на момент створення обчислювальні ресурси були обмежені, модель була реалізована таким чином, щоб паралельно обробляти дані на двох відеокартах NVIDIA GTX 580.

Значення архітектури AlexNet. Модель AlexNet започаткувала нову хвилю інтересу до глибоких нейронних мереж. Її поява стала поворотним моментом, який започаткував сучасну епоху комп’ютерного зору. Всі основні архітектури, які з’явилися пізніше (VGG, GoogLeNet, ResNet), так чи інакше базуються на концепціях, реалізованих у AlexNet.

AlexNet також продемонструвала, що збільшення глибини мережі в поєднанні з правильною архітектурною організацією (зокрема використанням нормалізації, регуляризації та масштабованості) дозволяє суттєво підвищити точність при розв’язанні задач класифікації зображень.

VGG-16 – згорткова нейронна мережа, розроблена дослідницькою групою Visual Geometry Group з Оксфордського університету у 2014 році. Модель була представлена у межах змагання ImageNet ILSVRC-2014 і показала, що збільшення глибини мережі з використанням малих фільтрів є ефективним підходом до покращення результатів класифікації зображень. Назва VGG-16 означає, що модель містить 16 навчуваних шарів.

Структура VGG-16. Архітектура VGG-16 складається з п’яти блоків згорткових шарів, що чергуються з pooling-шарами, після яких слідують три повнозв’язні шари. Усі згортки мають розмір ядра 3×3 з кроком 1, що забезпечує фіксовану просторову структуру. Вхідне зображення має розмір 224×224×3, а вихідний шар формує ймовірності по 1000 класах. Загальна кількість параметрів перевищує 138 мільйонів.

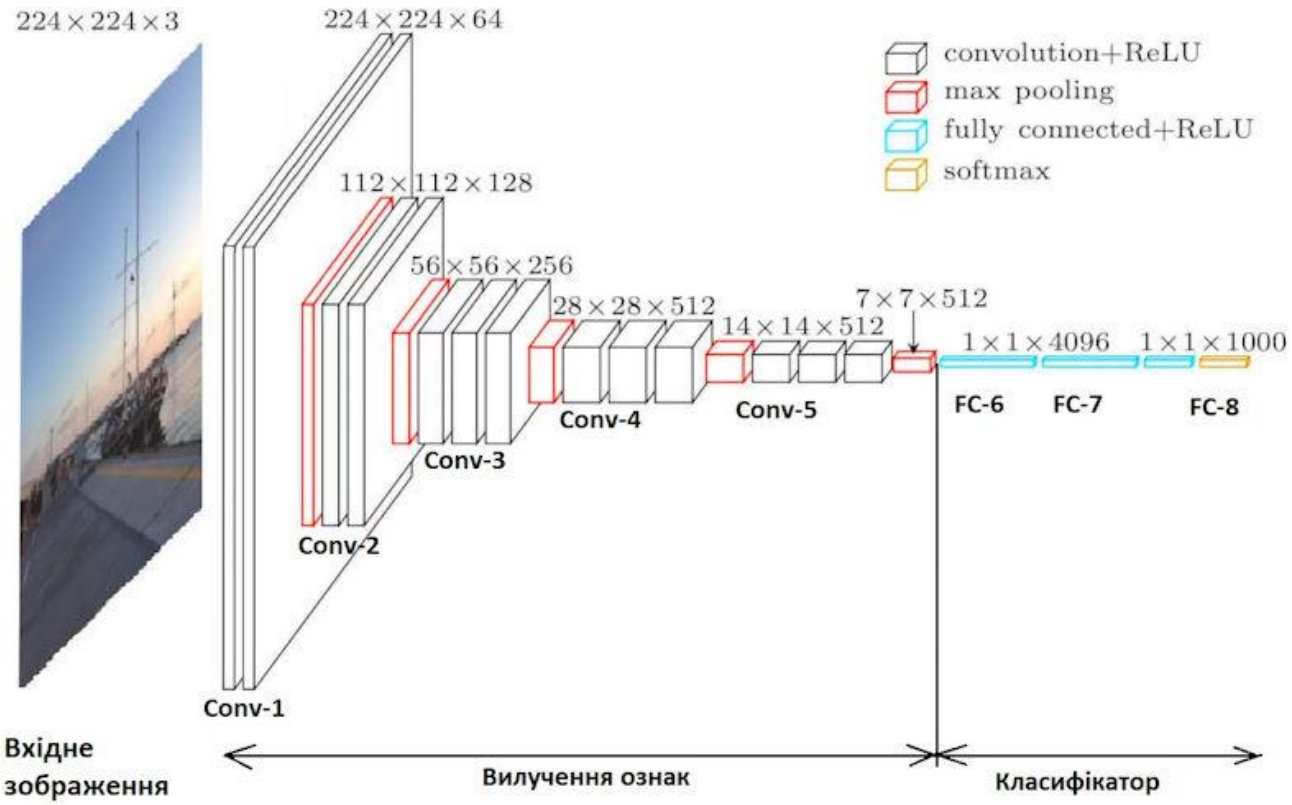


Рисунок 1.4 – Архітектура VGG-16

На рисунку 1.4 показано архітектурну побудову VGG-16. Видно, як на кожному етапі збільшується кількість каналів ознак, а просторові розміри зменшуються за рахунок pooling. Наприкінці подано три щільно з’єднані шари, які формують фінальну класифікацію.

Ключові особливості VGG-16:

– використання однотипних згорток 3×3 по всій мережі забезпечує простоту й масштабованість архітектури;

– maxPooling використовується для поступового зменшення просторових розмірів;

– повнозв’язна частина містить два шари по 4096 нейронів і один вихідний шар;

– незважаючи на обчислювальну складність, модель активно використовується в задачах transfer learning завдяки високій точності.

Значення архітектури VGG-16. Модель VGG-16 стала класичним прикладом глибокої згорткової мережі зі сталою структурою. Її симетричність, простота та якісні результати забезпечили широке використання в академічних дослідженнях і прикладних розробках. Архітектура VGG-16 стала основою для подальших моделей, таких як ResNet і DenseNet, і залишається актуальною як базова модель у задачах комп’ютерного зору.

1.3 Сучасні архітектури (ResNet, Inception, EfficientNet)

З розвитком глибинного навчання потреба в покращенні точності моделей і водночас зменшенні обчислювальних витрат призвела до створення нових архітектур згорткових нейронних мереж. Починаючи з 2015 року, було запропоновано низку ефективних моделей, які враховували обмеження попередніх підходів. Серед них ключовими є ResNet, Inception та EfficientNet – моделі, що заклали основу сучасного машинного зору.

ResNet (Residual Network) – архітектура згорткової нейронної мережі, представлена дослідниками Microsoft у 2015 році. Основна ідея ResNet полягає у введенні залишкових з’єднань (residual connections), які дозволяють інформації передаватися через кілька шарів без змін. Такий підхід усуває проблему затухання градієнтів, яка виникала при навчанні дуже глибоких моделей.

Структура ResNet. Мережа складається з послідовності залишкових блоків, у яких вхідний сигнал додається до виходу згорткових операцій. Найпоширенішими

є модифікації ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152. Найбільш популярна модель – ResNet-50 – складається з 50 навчуваних шарів, серед яких чергуються згорткові шари, операції нормалізації, активації ReLU та pooling.

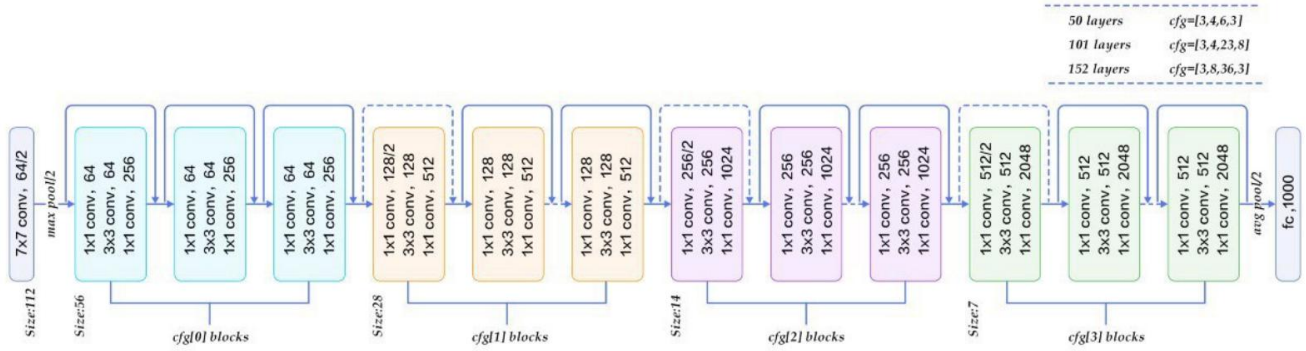


Рисунок 1.5 – Архітектура ResNet

На рисунку 1.5 зображено залишковий блок, який складається з двох згорткових шарів, між якими реалізовано пряме з’єднання з вхідного шару. Це дозволяє мережі зберігати інформацію та полегшує навчання.

Ключові особливості ResNet:

– використання прямого з’єднання між шарами для покращення потоку градієнта;

– можливість навчання дуже глибоких мереж (понад 100 шарів);

– підвищення точності без збільшення ризику перенавчання;

– універсальність – застосовується як у класифікації, так і в сегментації та детекції.

Значення архітектури ResNet. ResNet стала стандартом для побудови глибоких моделей у комп’ютерному зорі. Її блокова структура активно використовується в модифікованих архітектурах, таких як Inception-ResNet, ResNeXt і Detectron. Модель знайшла застосування в медицині, автономних системах та промисловій візуальній аналітиці.

Inception – це архітектура згорткової нейронної мережі, представлена компанією Google у 2014 році під назвою GoogLeNet. Основна концепція полягає в паралельному застосуванні згорток різного розміру до одного і того самого

вхідного зображення, що дозволяє виявляти ознаки на різних просторових масштабах.

Структура Inception. Модель побудована з блоків типу Inception, кожен з яких включає паралельні гілки з фільтрами розміром 1×1, 3×3, 5×5, а також операцію pooling. Результати всіх гілок об'єднуються по глибині (конкатенація). Такий підхід дозволяє досягти високої точності без суттєвого збільшення обчислювальної складності. У початковій версії GoogLeNet мережа містить 22 навчувані шари[17].

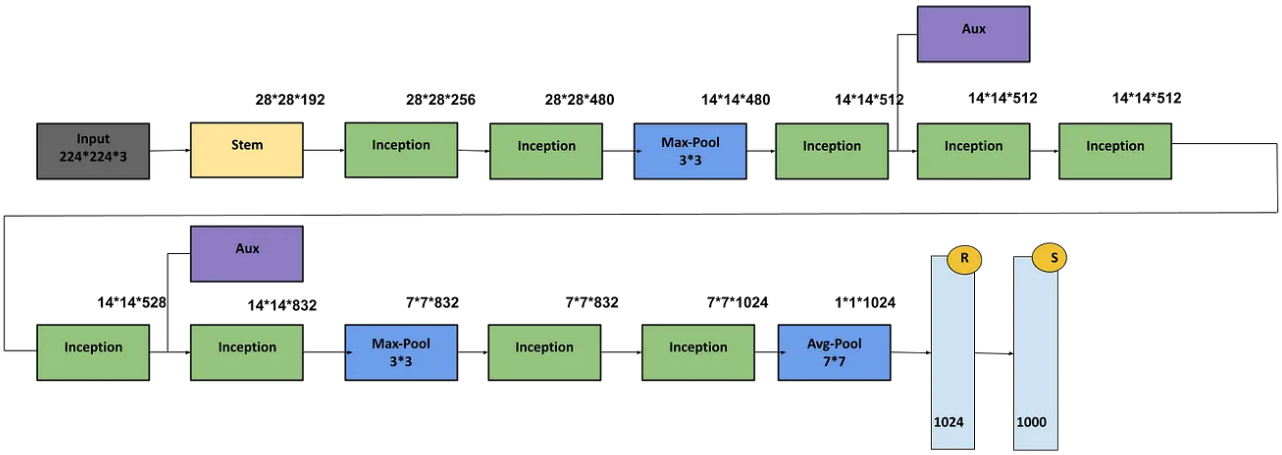


Рисунок 1.6 – Архітектура мережі Inception (GoogLeNet)

На рисунку 1.6 зображено повну архітектуру моделі Inception (GoogLeNet), що складається з послідовності модулів Inception, розміщених після первинного шару обробки (Stem). Структура включає шари згортки, pooling, допоміжні класифікатори (Aux) та фінальні шари класифікації. Модель приймає зображення розміром 224×224×3 та генерує вектор ознак розмірності 1024, який подається до класифікатора з 1000 вихідними класами[18].

Ключові особливості Inception:

– паралельна обробка згортками різного масштабу;

– використання 1×1 згорток для зменшення розмірності (виконують роль «бутилки»);

– компактність архітектури при високій точності;

– оптимальне співвідношення точності й ефективності;

Значення архітектури Inception. Inception заклала принципи модульності та багатогілкової обробки, які були успішно використані в подальших версіях (Inception-v2, v3, v4, Inception-ResNet). Архітектура знайшла широке застосування в комерційних і хмарних сервісах Google, а також у мобільних пристроях.

EfficientNet – серія згорткових нейронних мереж, представлена Google у 2019 році. Основна ідея моделі полягає у синхронному масштабуванні глибини, ширини та роздільної здатності вхідних зображень, що дозволяє будувати максимально ефективні мережі за кількістю параметрів та точністю.

Структура EfficientNet. Базова модель EfficientNet-B0 побудована з використанням MBConv-блоків, що поєднують роздільні згортки (depthwise separable convolutions) та механізм уваги squeeze-and-excitation. Старші версії (B1–B7) масштабуються автоматично згідно з принципом compound scaling.

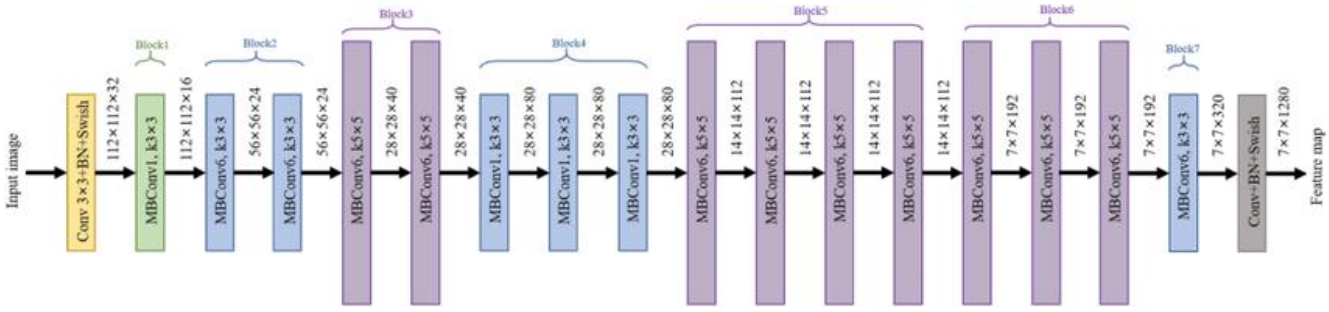


Рисунок 1.7 – Архітектура мережі EfficientNet-B0

На рисунку 1.7 представлено архітектуру EfficientNet-B0. Кожен прямокутний блок відповідає певному рівню обробки даних у моделі(рис.1.7) – від вхідного зображення до повнозв’язного класифікатора. Чітко видно повторювану структуру MBConv-блоків, що застосовуються з різними параметрами упродовж всієї глибини моделі.

Ключові особливості EfficientNet:

– *масштабування compound scaling.* Дозволяє автоматично балансувати збільшення глибини, ширини та роздільної здатності моделі;

– *MBConv-блоки*. Забезпечують ефективне обчислення завдяки глибинним згорткам і механізму уваги squeeze-and-excitation;

– *оптимальна кількість параметрів*. Модель B0 має лише ≈5,3 млн параметрів, але демонструє конкурентну точність;

– *гнучкість*. Архітектура масштабована до потужніших моделей B1–B7 із поступовим підвищенням точності.

Значення EfficientNet. Модель EfficientNet стала новим еталоном ефективності в задачах класифікації зображень. Вона поєднує високу точність і економне використання ресурсів, що забезпечило її широке використання в мобільних пристроях, хмарних сервісах, системах автоматичного розпізнавання об’єктів і медичній візуалізації. Її ідеї масштабування активно переймають у нових поколіннях архітектур нейронних мереж.

1.4 Диференціація CNN, GAN, Vision Transformers

У сучасному комп’ютерному зорі застосовуються різні підходи до побудови нейронних мереж для обробки зображень. Найпоширенішими з них є згорткові нейронні мережі (CNN), генеративні змагальні мережі (GAN) та трансформерні архітектури для зображень (Vision Transformers, ViT). Ці моделі суттєво відрізняються між собою за структурою, принципом обробки інформації, сферою застосування та вимогами до навчання.

Згорткові нейронні мережі (CNN) – клас моделей, спеціально розроблених для аналізу зображень. Вони засновані на локальних згорткових операціях, які дозволяють автоматично виділяти ознаки (features) на різних рівнях абстракції – від простих контурів до складних об’єктів.

Мережа CNN будується з згорткових шарів, що виконують витягування ознак, пулінгових шарів, що зменшують розмірність, та повнозв’язних шарів для класифікації. Завдяки своїй локальній структурі та використанню фільтрів з

обмеженим рецептивним полем, CNN є ефективними для обробки зображень стандартного розміру.

Приклади застосування: класифікація зображень, сегментація, детекція об’єктів, медична діагностика.

Генеративні змагальні мережі (GAN) – це архітектура, що складається з двох взаємопов’язаних моделей: генератора і дискримінатора, які змагаються між собою в процесі навчання. Генератор створює нові зображення, а дискримінатор намагається відрізнити їх від справжніх. Таким чином, модель навчається створювати реалістичні зображення, які неможливо відрізнити від реальних.

Навчання GAN відбувається у вигляді мінімаксної гри, що робить процес нестабільним і чутливим до налаштувань. Однак у разі вдалого навчання GAN демонструють вражаючу здатність до генерації високоякісного контенту.

Приклади застосування: генерація зображень, суперрезолюція, стилізація, відновлення пошкоджених фрагментів зображень.

Vision Transformers (ViT) – архітектура, яка переносить механізм самоуваги (self-attention), широко використовуваний у природній мовній обробці, на обробку зображень. Замість згорткових фільтрів модель ViT розбиває зображення на патчі, які потім обробляються як послідовність токенів[14][30].

На відміну від CNN, які обробляють локальні контексти, ViT здатні моделювати глобальні залежності між елементами зображення вже на ранніх етапах обробки. Це забезпечує високу точність, особливо при наявності великих обсягів даних для навчання.

Приклади застосування: класифікація зображень, трансферне навчання, мультимодальні моделі (текст + зображення).

Таблиця 1.1 – Порівняльна характеристика архітектур CNN, GAN та Vision Transformer

Ознака	CNN	GAN	Vision Transformer
Архітектура	Послідовність згорткових і pooling	Генератор + дискримінатор	Механізм самоуваги, токенізація патчів
Принцип роботи	Витягування локальних ознак	Змагальне навчання двох моделей	Моделювання глобальних залежностей
Основне призначення	Класифікація, сегментація	Генерація зображень, стилізація	Класифікація, інтеграція з NLP-моделями
Необхідний обсяг даних	Помірний	Високий	Дуже високий
Вимоги до ресурсів	Низькі–середні	Високі	Високі
Стабільність навчання	Висока	Нестабільне	Висока (але чутлива до гіперпараметрів)

1.5 Порівняння архітектур по точності, швидкості, обчислювальній складності

Під час вибору нейронної мережі для розв’язання задач комп’ютерного зору важливо враховувати не лише точність моделі, але й її обчислювальну ефективність та придатність до використання в різних умовах (наприклад, на сервері або в мобільному пристрої). У цьому розділі здійснено порівняння популярних архітектур згорткових нейронних мереж: AlexNet, VGG-16, ResNet-50, Inception-v3 та EfficientNet-B0.

Точність класифікації. Оцінювання точності проводиться за метрикою Top-1 accuracy на датасеті ImageNet, що відображає частку прикладів, для яких модель правильно передбачила найбільш ймовірний клас. Еволюція архітектур демонструє поступове зростання точності з появою нових моделей.(табл.1.2)

Таблиця 1.2 – Порівняння поширених архітектур за точністю, складністю та придатністю до розгортання

Архітектура	Top-1 точність (ImageNet)	Параметри (млн)	FLOPs (млрд)	Придатність до розгортання
AlexNet	~57 %	60	0.72	Проста, але застаріла
VGG-16	~71 %	138	15.5	Високоточна, неефективна
ResNet-50	~76 %	25.6	4.1	Баланс точності й ефективності
Inception-v3	~78 %	23.5	5.7	Висока ефективність
EfficientNet-B0	~77 %	5.3	0.39	Оптимальна для мобільних пристроїв

Аналіз результатів:

– VGG-16 забезпечує високу точність, але має надмірну кількість параметрів, що ускладнює її практичне використання;

– ResNet-50 та Inception-v3 досягають кращого балансу між точністю та обчислювальною складністю;

– EfficientNet-B0 демонструє найкращу ефективність при збереженні конкурентної точності, що робить її придатною для задач, де важливі низькі ресурси – наприклад, у мобільних застосунках.

2 МЕТОДИ НАВЧАННЯ ТА ОПТИМІЗАЦІЇ НЕЙРОННИХ МЕРЕЖ

2.1 Навчання з учителем, без учителя, transfer learning

Процес навчання нейронних мереж визначає здатність моделей до узагальнення та ефективного розв’язання прикладних задач. Залежно від доступності та характеру навчальних даних, виділяють кілька основних типів навчання: з учителем (supervised learning), без учителя (unsupervised learning), а також трансферне навчання (transfer learning). Кожен із підходів має свої переваги, обмеження та сфери застосування.

Навчання з учителем (Supervised Learning). Це найбільш поширений і формалізований тип навчання нейронних мереж. Модель навчається на основі пари “вхід – правильна відповідь”. Для кожного зображення x у тренувальному наборі дано відповідну мітку класу y, яка вказує на правильний результат[9][28].

Метою є побудова функції $f(x;\theta)$ (2.1), яка апроксимує залежність між вхідними даними та мітками:

$$f(x;\theta) \approx y \tag{2.1}$$

де θ – параметри нейронної мережі, які оновлюються під час навчання.

Цей тип навчання використовується в задачах:

– класифікації зображень (наприклад, визначення категорії об’єкта);

– сегментації (розмітка кожного пікселя);

– локалізації (визначення координат об’єкта);

– розпізнавання облич, медичної діагностики, контролю якості на виробництві.

Перевагою є висока точність за умови наявності великого обсягу якісно розмічених даних. Основним обмеженням є саме потреба в таких розмічених наборах.

Навчання без учителя (Unsupervised Learning). У цьому режимі модель працює лише з нерозміченими даними, тобто відсутня інформація про правильні відповіді. Метою є виявлення прихованих закономірностей, кластерів, структури ознак тощо[6].

У контексті обробки зображень основними завданнями є:

– кластеризація подібних зображень (на основі кольору, форми, текстури);
– виявлення аномалій (наприклад, дефектів на зображенні);
– зниження розмірності (за допомогою автоенкодерів або PCA);
– попереднє навчання (pretraining) без міток.

Поширеними архітектурами є автоенкодери, кластеризуючі CNN, self-supervised моделі. Навчання без учителя особливо важливе там, де немає можливості розмітити великі датасети.

Transfer learning (Трансферне навчання). Transfer learning – це підхід, за якого модель, попередньо навчена на великому базовому наборі даних (наприклад, ImageNet), адаптується до нової задачі з обмеженим обсягом власних даних.

Суть полягає в тому, що перші шари моделі “переучувати” не потрібно, оскільки вони вже вивчили загальні ознаки зображень (контури, текстури), які корисні для більшості задач. Зазвичай змінюються лише кінцеві шари класифікатора або виконується тонке донавчання (fine-tuning) всієї мережі з меншою швидкістю навчання[13].

Сценарії використання:

– медичні зображення (де обмаль розмічених прикладів);
– спеціалізовані промислові об’єкти;
– задачі з нестандартними класами, але типовими візуальними структурами.

Transfer learning дозволяє істотно зменшити час та ресурси навчання, забезпечуючи при цьому високу якість результатів, навіть за наявності невеликих датасетів.

2.2 Функції втрат, оптимізатори (SGD, Adam, RMSprop)

Навчання нейронної мережі передбачає поступове вдосконалення її параметрів таким чином, щоб вихідні передбачення якомога точніше відповідали реальним міткам. Цей процес реалізується шляхом мінімізації функції втрат з використанням алгоритмів оптимізації. У цьому підрозділі розглянуто найбільш поширені функції втрат та оптимізатори, які застосовуються в задачах комп'ютерного зору.

Функції втрат (Loss functions) – це математичні вирази, які використовуються для кількісного оцінювання різниці між передбаченим значенням, отриманим від моделі, та істинним (очікуваним) значенням. Вони відіграють ключову роль у процесі навчання нейронних мереж, оскільки визначають, наскільки добре або погано модель виконує своє завдання. Значення функції втрат мінімізується в процесі оптимізації, що дозволяє поступово покращувати точність моделі. Вибір функції втрат залежить від типу задачі: для класифікації зазвичай використовуються кросентропія (categorical cross-entropy), для регресії – середньоквадратична помилка (mean squared error, MSE) або середня абсолютна похибка (MAE). Функція втрат визначає, наскільки передбачення моделі відрізняється від очікуваного результату. Вона є критерієм для оновлення ваг мережі через зворотне поширення помилки.

1. Крос-ентропія (Categorical Cross-Entropy) – це функція втрат, яка використовується переважно у задачах багатокласової класифікації. Вона вимірює різницю між двома ймовірнісними розподілами: фактичним (істинним) розподілом класів та передбаченим розподілом, що генерується нейронною мережею (2.2). Чим

ближче передбачений розподіл до істинного, тим менше значення функції втрат. Крос-ентропія стимулює модель призначати високі ймовірності правильному класу й низькі – неправильним[12].

Математично для одного зразка вона визначається як:

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(p_i) \tag{2.2}$$

де y_i – правильне значення (одиниця для правильного класу, нуль для інших), p_i – ймовірність, передбачена мережею, C – кількість класів.

2. Середньоквадратична помилка (Mean Squared Error, MSE) – це функція втрат, яка широко використовується у задачах регресії для оцінки якості передбачень моделі (2.3). Вона вимірює середнє значення квадратів різниць між фактичними (істинними) значеннями та передбаченими значеннями моделі. MSE характеризує загальну похибку передбачення: чим менше її значення, тим точніші прогнози моделі[24].

Математично визначається як:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.3}$$

де n – кількість прикладів у вибірці, y_i – істинне значення для i-го прикладу, $\hat{y}_i$ – передбачене значення для i-го прикладу.

Для класифікаційних задач використовується рідко, оскільки менш чутлива до розподілу ймовірностей.

3. Бінарна крос-ентропія (Binary Cross-Entropy) – це функція втрат, яка застосовується у задачах бінарної класифікації (2.4), де результат має лише два можливих класи (наприклад, «0» або «1»). Вона вимірює різницю між фактичною бінарною міткою та передбаченою ймовірністю, що генерується моделлю. Чим ближче передбачення до правильного класу, тим менше значення функції втрат[29].

Математично визначається як:

$$L = -[y \cdot \log(p) + (1 - y) \cdot \log(1 - P)] \quad (2.4)$$

де y – істинне значення (0 або 1), p – передбачена ймовірність належності до класу 1.

Binary Cross-Entropy карає великі відхилення передбачення від правильного класу, особливо коли модель впевнено помиляється. Вона є стандартом для задач класифікації «так/ні» у нейронних мережах.

Алгоритми оптимізації. Оптимізатори визначають, як змінюються ваги нейронної мережі під час навчання. Вони використовують градієнт функції втрат для коригування параметрів у напрямку зменшення помилки[20].

1. Стохастичний градієнтний спуск (SGD – Stochastic Gradient Descent) – ітеративний метод оптимізації градієнтного спуску за допомогою стохастичного наближення. Використовується для прискорення пошуку цільової функції шляхом використання обмеженого за розміром тренувального набору, який вибирається випадково при кожній ітерації[15].

Базовий метод стохастичного градієнтного спуску (2.5). Оновлює ваги за градієнтом на кожному прикладі або мініпакеті:

$$\theta = \theta - \eta \cdot \nabla \mathcal{L} \quad (2.5)$$

де θ – ваги, η – швидкість навчання, ∇L – градієнт функції втрат.

Незважаючи на простоту, має недоліки: може “застрягати” в локальних мінімумах, чутливий до вибору η.

2. RMSprop (Root Mean Square Propagation) – це адаптивний оптимізатор градієнтного спуску, який був запропонований Джеффом Гінтоном. Основна ідея RMSprop полягає в автоматичному коригуванні швидкості навчання для кожного параметра моделі залежно від величини його градієнтів. Це дозволяє уникнути

проблеми вибухаючих або зникаючих градієнтів, а також сприяє стабільнішому та швидшому навчанню нейронних мереж, особливо на нестаціонарних задачах, таких як обробка послідовностей або потокових даних[21].

На відміну від звичайного стохастичного градієнтного спуску (SGD), RMSprop використовує експоненційне згладжування квадратів градієнтів(2.6)(2.7).

$$v_t = \gamma v_{t-1} + (1-\gamma)(\nabla \mathcal{L})^2 \tag{2.6}$$

$$\theta = \theta - \frac{\eta}{\sqrt{v_t+\epsilon}} \cdot \nabla \mathcal{L} \tag{2.7}$$

де v_t – зважене середнє квадратів градієнтів на кроці t, γ – коефіцієнт експоненціального згладжування (рекомендоване значення – 0.9), ∇L – градієнт функції втрат L відносно параметрів моделі, η – швидкість навчання (learning rate), ε – невелике число для забезпечення чисельної стабільності (наприклад, 10^{-8}), θ – вектор параметрів моделі, що оновлюється.

Ці рівняння дозволяють моделі автоматично адаптувати крок оновлення параметрів залежно від масштабу градієнтів, що особливо корисно при роботі з глибокими або складними архітектурами. Підходить для нестаціонарних або “шумних” задач, особливо ефективний у RNN та GAN.

3. Adam (Adaptive Moment Estimation) – це один з найпопулярніших алгоритмів оптимізації у глибокому навчанні, який поєднує в собі ідеї методів моменту та RMSprop. Adam автоматично адаптує швидкість навчання для кожного параметра моделі, використовуючи як перший момент (середнє значення градієнтів), так і другий момент (середнє значення квадратів градієнтів).

Формули для оновлення параметрів у методі Adam наведено у виразах (2.8-2.10)

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla \mathcal{L} \tag{2.8}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla \mathcal{L})^2 \quad (2.9)$$

$$\theta = \theta - \eta \cdot \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (2.10)$$

де m_t – перша моментна оцінка (середнє градієнтів), v_t – друга моментна оцінка (середнє квадратів градієнтів), β_1, β_2 – коефіцієнти згладжування для моментів (типові значення: β_1 = 0,9, β_2 = 0,999), ∇L – градієнт функції втрат, η – швидкість навчання, ε – невелике число для запобігання діленню на нуль, θ – параметри моделі, що оновлюються[16].

Ці формули реалізують адаптивне оновлення параметрів шляхом балансування між поточними градієнтами та історичною інформацією про градієнти, що робить Adam стійким і ефективним оптимізатором для широкого кола завдань.

2.3 Аугментація даних та регуляризація

Навчання нейронних мереж, зокрема згорткових, супроводжується ризиком перенавчання, коли модель надмірно точно відтворює шаблони з тренувальних даних, але втрачає здатність до узагальнення. Для зменшення цього ефекту використовуються методи регуляризації та аугментація даних. Ці техніки дозволяють підвищити стійкість моделі та її ефективність на нових зображеннях.

Аугментація даних – це штучне збільшення обсягу навчального набору даних шляхом генерації варіацій вихідних прикладів, які зберігають вихідний клас[22]. У контексті комп’ютерного зору застосовуються такі перетворення:

– *геометричні трансформації*. Обертання, віддзеркалення, масштабування, обрізання (crop), зсув;

– *зміни освітлення та кольору*. Варіації яскравості, контрастності, насиченості, відтінків;

– *додавання шуму*. Імітація шуму камери або перекриття зображення випадковими перешкодами;

– *Cutout, Mixup, CutMix*. Сучасні методи, що змінюють не лише одне зображення, а створюють комбінації кількох зразків, підвищуючи стійкість до локальних особливостей.

Аугментація виконується в реальному часі (on-the-fly) під час навчання, що дозволяє значно збільшити різноманітність даних без потреби зберігати додаткові файли.

Регуляризаційні методи запобігають перенавчанню шляхом обмеження складності моделі або варіативності вагових коефіцієнтів.

1. Dropout – це метод регуляризації, що широко використовується в нейронних мережах для зниження ризику перенавчання (overfitting). Основна ідея полягає у випадковому "вимиканні" певної частини нейронів у шарі під час кожної ітерації навчання. З математичної точки зору, кожен нейрон має певну ймовірність p бути тимчасово деактивованим. Це означає, що під час проходження одного батчу частина нейронів не бере участі в обчисленнях – їх вихід примусово дорівнює нулю.

Такий підхід змушує модель не залежати від окремих вузлів, а шукати більш стійкі ознаки у вхідних даних. Dropout сприяє розвитку більш узагальненого представлення ознак і дозволяє уникнути надмірного пристосування моделі до навчальної вибірки. Завдяки цьому модель демонструє кращу здатність до генералізації на нових даних.

У фазі тестування всі нейрони активні, але вихідні значення масштабуються з урахуванням ймовірності збереження, тобто множаться на $1-p$, щоб компенсувати ефект вимкнених нейронів під час тренування.

Зазвичай Dropout застосовується після повнозв’язних шарів у згорткових нейронних мережах (CNN), оскільки саме ці шари містять найбільше параметрів і найбільш схильні до перенавчання.

2. L2-регуляризація (ваговий штраф або ridge-регуляризація) – це метод регуляризації, що додає до функції втрат додатковий член, який пропорційний

квадрату величини вагових коефіцієнтів моделі (2.11). Основна мета L2-регуляризації – зменшити складність моделі шляхом зменшення значень ваг, що дозволяє уникнути перенавчання (overfitting) і покращити узагальнювальну здатність нейронної мережі. Додає до функції втрат член, пропорційний квадрату ваг(2.11)

$$\mathcal{L}_{new} = \mathcal{L} + \lambda \sum_i \theta_i^2 \tag{2.11}$$

де λ – коефіцієнт регуляризації, θi – параметри моделі. Це стимулює ваги залишатися малими, запобігаючи “перенавчанню”.

3. Batch Normalization – це метод нормалізації проміжних результатів (активацій) у нейронній мережі, який застосовується під час навчання для покращення стабільності та прискорення збіжності моделі. Його основна мета – зменшення чутливості мережі до зміни розподілу вхідних даних на кожному шарі (ефект, відомий як "зсув внутрішнього розподілу").

Хоча головною ціллю є стабілізація та прискорення навчання, Batch Normalization також має виражений регуляризуючий ефект. Вона нормалізує активації в межах кожного мініпакету, що знижує чутливість до початкових значень ваг і коливань у розподілі навчальних даних. Завдяки цьому модель краще узагальнює і менш схильна до перенавчання.

Метод широко застосовується в сучасних архітектурах після згорткових або повнозв’язних шарів, особливо у складних глибоких мережах, де є потреба в стабільному навчальному процесі.

4. Early Stopping – це метод регуляризації, який використовується для запобігання перенавчанню нейронної мережі шляхом контролю за динамікою зміни валідаційної помилки під час навчання. Суть методу полягає у завершенні процесу тренування раніше запланованої кількості епох, якщо модель перестає покращуватись на валідаційній вибірці.

Механізм Early Stopping перериває навчання, якщо протягом кількох послідовних епох валідаційна помилка перестає знижуватись або починає зростати.

Це дозволяє уникнути ситуації, коли модель починає запам’ятовувати навчальні дані замість узагальнення. Таким чином, метод дає змогу зберегти ваги моделі у найкращому стані – тобто в тій точці, де вона досягла оптимального балансу між точністю та здатністю до узагальнення.

Early Stopping часто використовується у поєднанні з іншими техніками регуляризації та є особливо ефективним при роботі з великими наборами даних або складними моделями[23].

2.4 Пояснення ефекту перенавчання

Перенавчання (overfitting) – це поширене явище при навчанні нейронних мереж, коли модель демонструє високу точність на тренувальних даних, але суттєво погіршує результати на нових, раніше не бачених прикладах. Це свідчить про те, що модель не узагальнює закономірності, а запам’ятовує конкретні зразки навчального набору[26].

Причини перенавчання:

1. *Низька кількість або якість навчальних даних*. Коли обсяг тренувального набору недостатній для охоплення усіх варіацій даних, модель адаптується до специфіки окремих прикладів. Те саме стосується й несбалансованих або зашумлених наборів даних.

2. *Надмірна складність моделі*. Мережі з великою кількістю параметрів можуть апроксимувати практично будь-яку функцію, у тому числі й випадкові флуктуації у навчальному наборі. Це особливо актуально для моделей із великою кількістю повнозв’язних шарів або великою глибиною.

3. *Відсутність регуляризації.* Якщо не використовуються методи обмеження складності моделі (Dropout, L2, BatchNorm), вона схильна до перенавчання навіть на порівняно великих наборах.

4. *Надмірна кількість епох*. Занадто тривале навчання без моніторингу валідаційної помилки може призвести до того, що модель “підлаштовується” під шум навчальних даних.

Ознаки перенавчання:

– різкий розрив між точністю на тренувальних і валідаційних даних: модель досягає > 95% точності на train, але значно нижчу на validation;

– зростання валідаційної втрати після кількох епох: в той час як навчальна помилка продовжує зменшуватись;

– нестійка поведінка метрик на тестових даних або нових зображеннях.

Графічна інтерпретація

На типовому графіку навчання, що відображає зміну втрати (loss) та точності (accuracy) по епохах:

– Train loss → зменшується стабільно;

– Validation loss → зменшується, а потім починає зростати;

– це сигнал до early stopping або застосування регуляризації.

Контекст у комп’ютерному зорі. У задачах класифікації зображень перенавчання особливо характерне при роботі з невеликими датасетами (наприклад, медичні або промислові зображення) або коли модель надто складна (наприклад, VGG-19 чи ResNet-101 на 500 прикладах).

Застосування аугментації, трансферного навчання та регуляризації дозволяє істотно зменшити ризик перенавчання у таких випадках.

2.5 Підходи до підвищення точності та стабільності моделей

Після базового навчання нейронної мережі її продуктивність можна покращити за допомогою різних методів, спрямованих на підвищення точності, стабільність при узагальненні та зниження залежності від випадкових флуктуацій у даних. Сучасна практика глибинного навчання передбачає використання як

архітектурних, так і процедурних прийомів для підвищення ефективності моделей[8].

1. Fine-tuning (тонке донавчання) – це одна з ключових технік перенесення навчання (transfer learning), яка дозволяє ефективно адаптувати великі попередньо навчені моделі до нових, більш специфічних завдань. Цей метод особливо корисний у випадках, коли цільовий набір даних є обмеженим за обсягом, або містить нові класи, відсутні у початковому наборі.

Це підхід, при якому модель, попередньо натренована на великому наборі (наприклад, ImageNet), адаптується до нової задачі шляхом часткового донавчання. Як правило:

– перші шари залишаються замороженими (не змінюються);

– останні шари (класифікатор) перенавчаються;

– за потреби виконується повне донавчання з низькою швидкістю навчання.

Це дозволяє значно покращити точність навіть на малих датасетах і скоротити час тренування.

2. Ensemble-моделі – це підхід у машинному навчанні та глибокому навчанні, який передбачає об’єднання кількох окремих моделей для покращення загальної точності, стійкості та здатності до узагальнення. Замість використання однієї моделі, ансамблеві методи поєднують результати декількох моделей (однорідних або різнорідних), щоб зменшити похибки, пов’язані з варіативністю, упередженістю або перенавчанням[25].

Ансамблювання (ensemble learning) полягає в об’єднанні передбачень кількох незалежних моделей. Основні варіанти:

– середнє (average) або більшість (voting) для класифікації;

– усереднення ймовірностей;

– stacking – побудова метамоделі, яка комбінує передбачення базових моделей.

Ансамблі забезпечують більш стабільний результат, оскільки зменшують вплив випадкових помилок окремих моделей.

3. Learning rate scheduling – це стратегія динамічної зміни швидкості навчання (learning rate) протягом тренування нейронної мережі. Її мета – покращити збіжність, стабільність та якість навчання за рахунок адаптації темпу оновлення ваг на різних етапах тренувального процесу.

Динамічна зміна швидкості навчання під час тренування дозволяє досягати кращої збіжності та уникати локальних мінімумів. Поширені варіанти:

– Step Decay – зменшення learning rate через кожні кілька епох;

– Exponential Decay – поступове експоненціальне зменшення;

– Cosine Annealing – згладжене зменшення з можливими повторними підйомами;

– ReduceLROnPlateau – автоматичне зниження при стабілізації валідаційної помилки.

4. Knowledge Distillation – це метод стиснення нейронних мереж, за якого менш ресурсомістка модель (student) навчається на основі поведінки попередньо натренованої, більш складної моделі (teacher). Основна ідея полягає не лише в повторенні остаточних міток (ground truth), а й у спробі відтворення “м’яких” ймовірностей – вихідних розподілів teacher-моделі, які містять додаткову інформацію про структуру даних і відносини між класами.

Це метод, за якого велика (teacher) модель навчає меншу (student) модель. Student-модель намагається повторити не тільки остаточні мітки, а й проміжні “м’які” ймовірності, що дозволяє зберегти точність при меншій кількості параметрів.

Цей підхід особливо корисний для розгортання моделей у середовищах з обмеженими ресурсами.

5. Self-training (самонавчання) та pseudo-labeling (псевдо-розмітка) – це підходи до напівавтоматичного навчання, що дозволяють ефективно використовувати нерозмічені дані у випадках обмеженого доступу до якісної розмітки. Ці методи базуються на поетапному розширенні навчальної вибірки за рахунок самостійно згенерованих міток.

Алгоритм дії полягає у наступному:

– на першому етапі модель навчається на невеликій кількості розмічених прикладів;

– потім вона застосовується до нерозмічених даних для автоматичного прогнозування міток (pseudo-labels);

– ці нові мітки додаються до навчального набору, і модель проходить повторне тренування на розширених даних.

Такий підхід дає змогу покращити загальну якість моделі без необхідності в дорогій ручній розмітці. Self-training та pseudo-labeling особливо ефективні в прикладних задачах, де отримання розмічених даних є складним або дорогим – наприклад, у медицині, промисловому контролі якості, екологічному моніторингу тощо.

Ці методи можуть комбінуватись між собою: наприклад, можна поєднати fine-tuning, навчання з динамічним learning rate та self-training, що дає значне підвищення точності без потреби у збільшенні даних.

3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ

3.1 Класифікація, сегментація, генерація зображень

У цьому підрозділі буде розглянуто три ключові завдання комп’ютерного зору, які широко вирішуються за допомогою нейронних мереж: класифікація, сегментація та генерація зображень. Кожне з цих завдань має важливе практичне значення – від автоматичного сортування зображень до створення нових візуальних елементів у дизайні, медицині, промисловості та цифровому мистецтві.

На основі експериментального дослідження буде проаналізовано, як сучасні архітектури нейромереж застосовуються у вирішенні зазначених завдань. Для кожного з підрозділів подано приклади, результати роботи моделей, а також використані сервіси або фреймворки. Такий підхід дозволяє не лише теоретично охарактеризувати можливості нейронних мереж, а й наочно показати їхню ефективність на практиці.

Класифікація зображень – одна з фундаментальних задач комп’ютерного зору. Метою є автоматичне присвоєння вхідному зображенню мітки класу з обмеженого набору (наприклад, «кішка», «пес», «м’яч»). Така задача формалізується як побудова функції (3.1):

$$f: X \to Y \tag{3.1}$$

де X – простір зображень, а Y – множина міток класів. Модель аналізує вхідне зображення, виділяє з нього ознаки, і на виході формує вектор ймовірностей, після чого застосовується функція softmax для визначення найбільш ймовірного класу.

Для реалізації класифікатора найчастіше використовуються згорткові нейронні мережі (CNN), які автоматично витягують ієрархії ознак – від простих (краї, текстура) до складних (форми, об'єкти)[7].

Практичне застосування: створення моделі класифікації через Teachable Machine. Щоб продемонструвати процес побудови класифікаційної моделі, було використано безкоштовний онлайн-інструмент Teachable Machine від Google. Цей сервіс дозволяє створити CNN-класифікатор з графічним інтерфейсом без написання коду. Для побудови моделі використовується архітектура MobileNet у режимі transfer learning, тобто перенавчання на основі заздалегідь натренованої моделі.

Крок 1. Підготовка датасету. Було створено три класи:

– клас 1: Кіт;

– клас 2: Пес;

– клас 3: М’яч.

До кожного класу завантажено по 5 прикладів зображень, взятих із відкритих джерел. Такі мініатюрні набори даних дозволяють перевірити, чи здатна модель запам’ятати ознаки класу й узагальнити їх на нові приклади (рис.3.1).

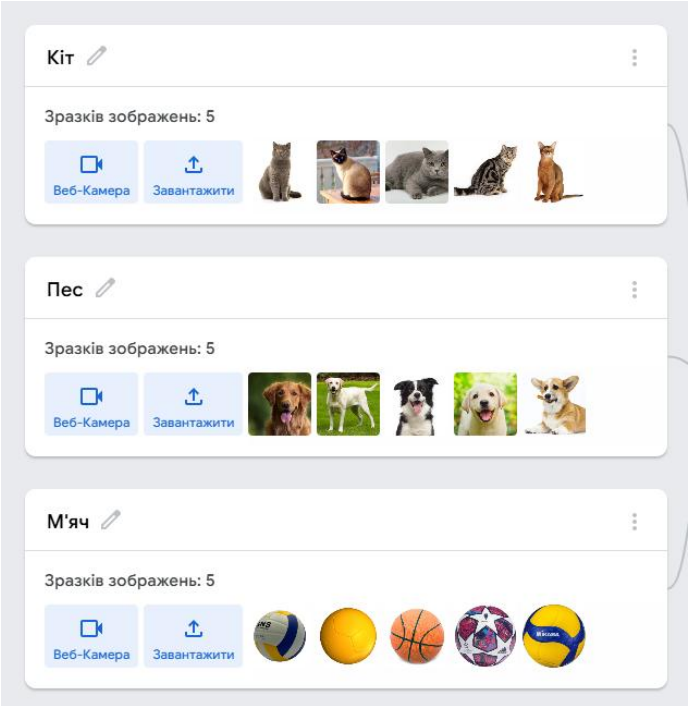


Рисунок 3.1 – Завантаження зображень до класів

У кожен із трьох створених класів – «Кіт», «Пес» та «М’яч» – було завантажено по 5 зображень, що ілюструють відповідний об’єкт. Ці зображення були підібрані так, щоб охопити різні ракурси, стилі зйомки та кольорові варіації, що дозволяє моделі навчитися узагальнювати ознаки класу. Таким чином, отримано базовий навчальний набір, достатній для початкового тестування принципів роботи згорткової нейронної мережі в задачі класифікації.

Крок 2. Навчання моделі. Після додавання зображень було запущено процес навчання – натисканням кнопки “Train Model”. Модель тренується прямо у браузері. Використовується попередньо натренований MobileNet як витягувач ознак, а на його основі навчається фінальний класифікатор.

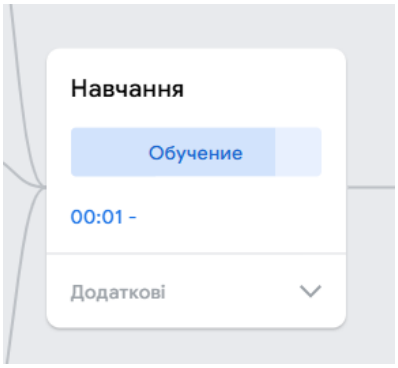


Рисунок 3.2 – Процес навчання моделі у Teachable Machine

На скріншоті (рис. 3.2) видно, що процес навчання моделі триває лише кілька секунд, оскільки використовується попередньо натренована архітектура (MobileNet), до якої адаптується лише фінальний класифікатор. У ході навчання модель підлаштовує свої вагові коефіцієнти відповідно до особливостей невеликого набору зображень, що дозволяє ефективно розпізнавати візуальні патерни навіть за обмеженої кількості прикладів. Такий підхід ілюструє переваги transfer learning у задачах швидкої адаптації моделі.

Крок 3. Тестування моделі. Після навчання було проведено тестування на новому зображенні, не використаному під час тренування. Завантажене фото собаки було класифіковано як “пес” із ймовірністю 99%, що свідчить про здатність моделі до узагальнення.

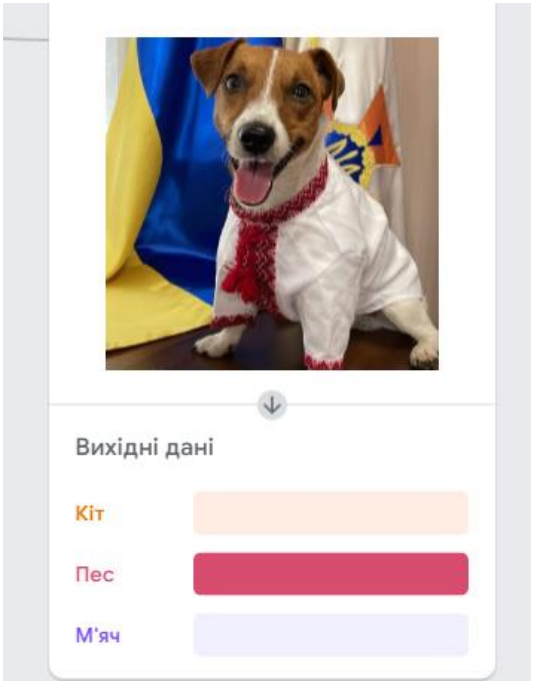


Рисунок 3.3 – Результат класифікації нового зображення

Після завершення навчання було проведено тестування моделі на нових зображеннях, які не входили до тренувального набору. Спочатку було завантажено зображення собаки, яке модель раніше не бачила, і результат передбачення показав високий рівень точності – зображення було класифіковано як “пес” із ймовірністю 99% (рис. 3.3). Це свідчить про те, що модель не просто запам’ятала навчальні приклади, а навчилася узагальнювати характерні візуальні ознаки класу.

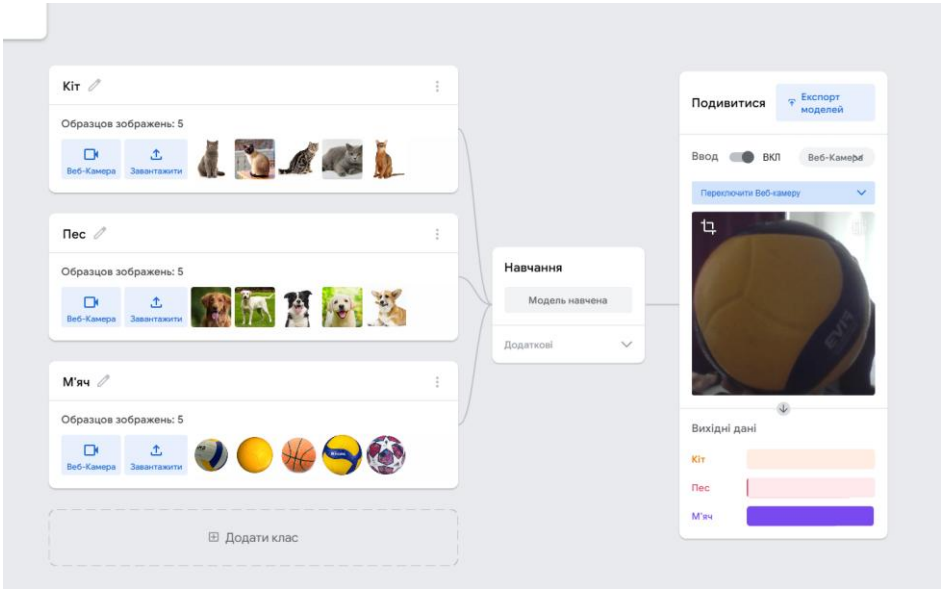


Рисунок 3.4 – Результат класифікації нового зображення

Для перевірки ще одного класу – “м’яч” – була використана функція тестування в режимі реального часу через вебкамеру (рис. 3.4). У кадр було поміщено реальний фізичний об’єкт – м’яч. Модель розпізнала його правильно з високою точністю в 98%, продемонструвавши, що навіть при мінімальному навчальному наборі вона здатна ідентифікувати ключові ознаки об’єктів у змінених умовах – у цьому випадку з відеопотоку, а не з фото.

Сегментація зображень – це задача комп’ютерного зору, яка полягає у визначенні, до якого класу належить кожен піксель зображення. На відміну від класифікації, де результатом є лише загальна мітка для всього зображення, у сегментації створюється піксельна маска, що точно позначає межі об'єктів.

Існує два основних типи сегментації:

– *семантична сегментація (semantic segmentation)*. Пікселі класифікуються за класами без відокремлення окремих об’єктів;

– *сегментація об'єктів (instance segmentation)*. Кожен об’єкт виділяється окремо навіть у межах одного класу.

Сегментація використовується в численних задачах: від медичної діагностики (виділення пухлин на МРТ) до автономного водіння (розпізнавання пішоходів, доріг, знаків).

Практичне застосування: SegmentAnything Model (SAM). Для демонстрації сегментації на практиці було використано відкритий онлайн-інструмент Segment Anything Model (SAM), розроблений Meta AI. Ця модель здатна виконувати високоточну сегментацію без попереднього навчання на конкретному наборі зображень. Достатньо лише натиснути на об’єкт на зображенні, і модель миттєво побудує відповідну маску.

Крок 1. Завантаження зображення. Було завантажено кольорове зображення, що містить кілька об’єктів (наприклад, ноутбук, чашка, людина). Обране зображення має просту композицію для наочної демонстрації роботи моделі.

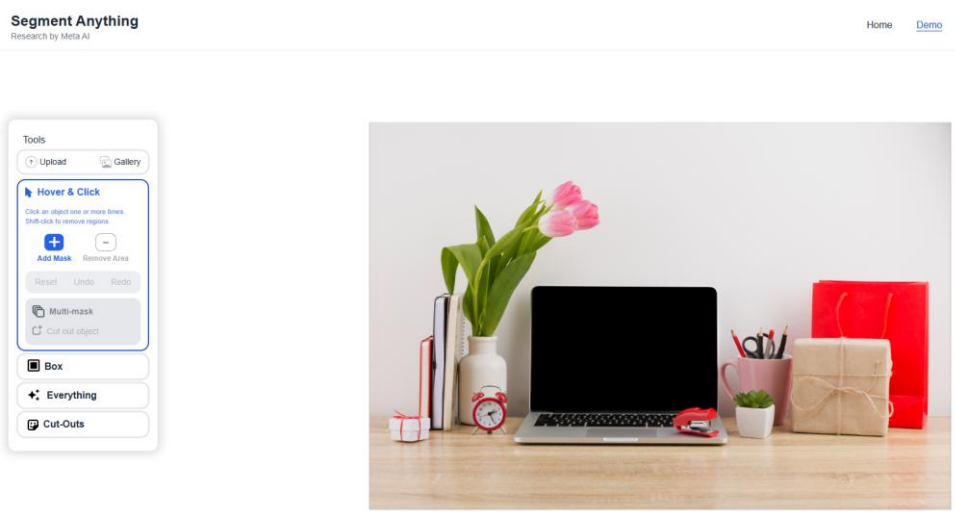


Рисунок 3.5 – Завантажене зображення для сегментації в SAM

Для демонстрації сегментації було використано зображення робочого столу з різними об'єктами: ноутбуком, канцелярією, квітами, чашкою, подарунковою упаковкою. Всі об'єкти мають чіткі контури та контрастний фон, що дозволяє наочно показати, як модель Segment Anything виділяє кожен з них окремо.

Крок 2. Виділення об’єкта. У SAM для виділення об’єкта достатньо натиснути на нього лівою кнопкою миші. Наприклад, було обрано об’єкт “ноутбук”, після чого модель автоматично побудувала точну піксельну маску, яка відокремлює ноутбук від фону.



Рисунок 3.6 – Результат виділення об’єкта (ноутбука) у SAM

Модель Segment Anything побудувала чітку та акуратну піксельну маску, яка точно охоплює лише контури ноутбука, не захоплюючи прилеглі об’єкти або фон. Виділення було виконане автоматично – достатньо було одного кліку миші по об'єкту. Сегментація продемонструвала високу точність навіть без попереднього навчання або ручної розмітки, що підкреслює універсальність і потужність підходу zero-shot segmentation, реалізованого в SAM. Такий рівень точності робить модель придатною для швидкого аналізу зображень у дизайнерських, освітніх чи дослідницьких цілях.

Крок 3. Сегментація всього зображення. Для демонстрації можливостей автоматичного розпізнавання кількох об’єктів одночасно було проведено повну сегментацію всього зображення. Модель Segment Anything самостійно виявила та побудувала окремі маски для кожного візуально відокремленого елемента: ноутбука, квітів у вазі, чашки з канцелярією, коробок, годинника, скріпочника тощо. Кожен об’єкт отримав свою власну маску, яка точно відповідає його контурам, навіть за складного переплетення форм і близького розташування.

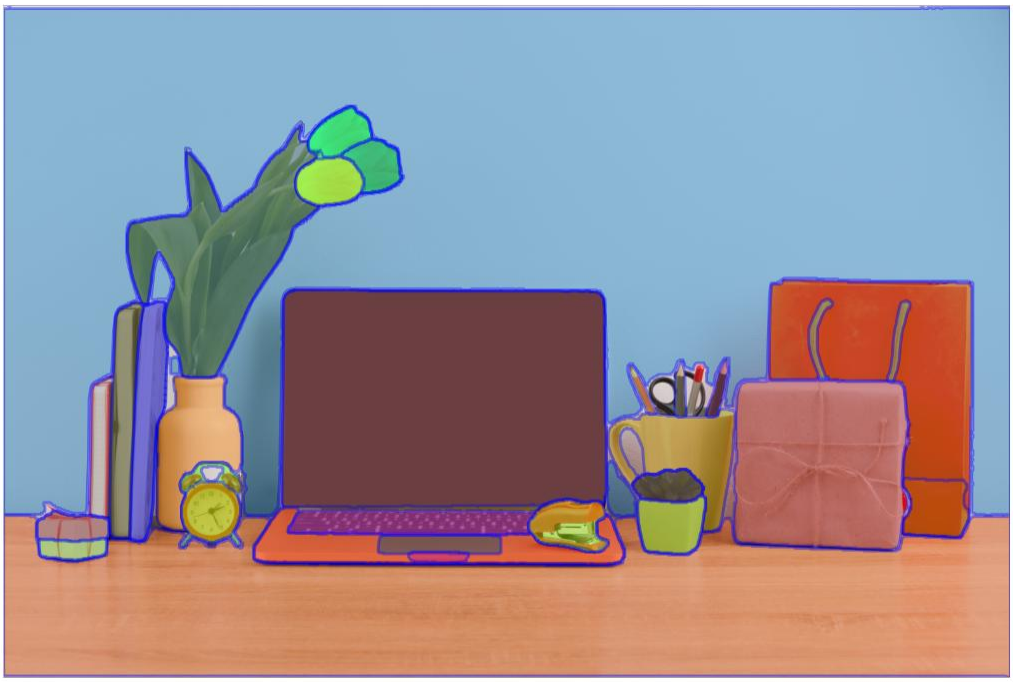


Рисунок 3.7 – Повна сегментація зображення за допомогою SAM

На ілюстрації видно, що модель побудувала точні піксельні маски для всіх об’єктів на зображенні: від великих (ноутбук, сумка, ваза) до дрібних (годинник,

квітка, скріпочник). Така детальність досягається без попереднього навчання на цьому конкретному наборі даних.

Альтернатива: автоматична сегментація Remove.bg. Для автоматизованої сегментації людини або основного об’єкта можна також використати сервіс Remove.bg, який миттєво видаляє фон. Хоча він не надає доступу до масок безпосередньо, результат – ізольований об’єкт на прозорому фоні – є формою сегментації.

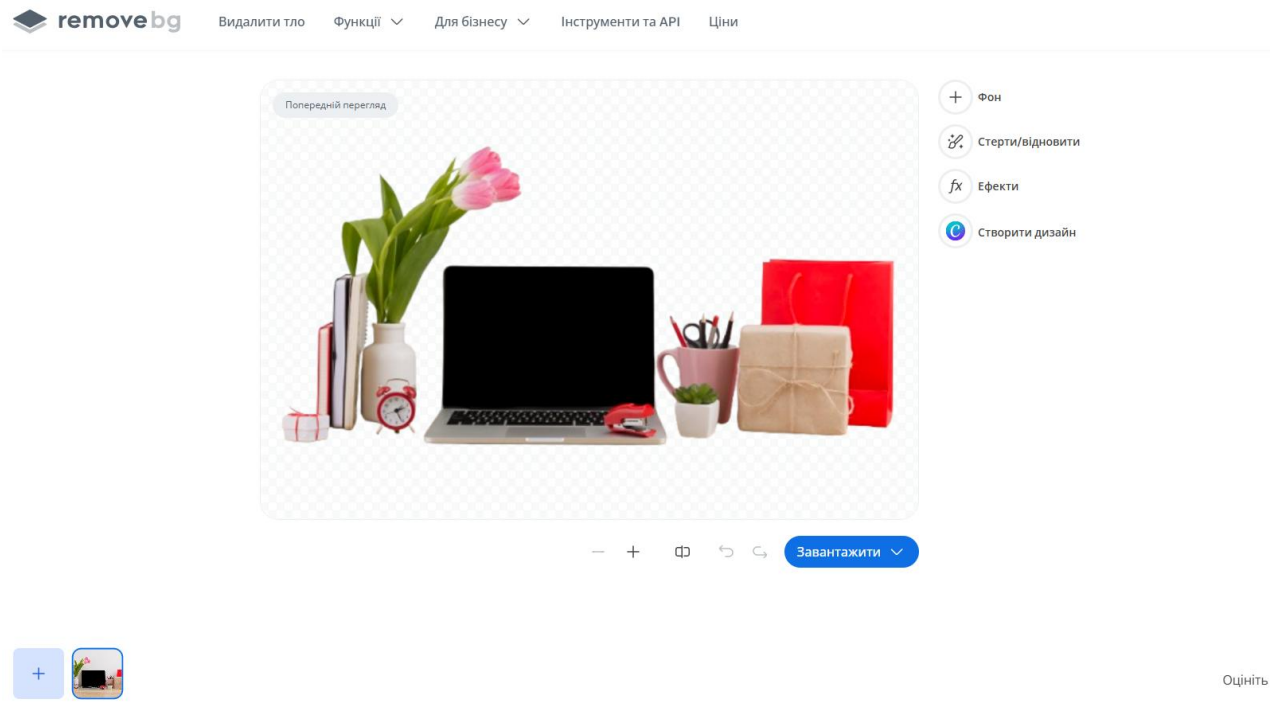


Рисунок 3.8 – Автоматичне видалення фону через Remove.bg

Remove.bg автоматично виділяє основний об’єкт (людину) та створює зображення з прозорим фоном.

Модель SAM побудована на основі трансформерної архітектури, яка поєднує візуальний енкодер, prompt-обробник і декодер масок. Вона є прикладом zero-shot segmentation, де сегментація виконується без попередньої адаптації до конкретного набору об'єктів. Це дозволяє ефективно застосовувати модель навіть у загальних задачах дизайну, редагування або попередньої обробки даних для тренування інших мереж.

Генерація зображень за допомогою нейронних мереж – це процес створення нових графічних зображень на основі заданих умов або описів. Цей тип задач

активно використовується у сферах дизайну, геймдизайну, ілюстрації, маркетингу та досліджень. Завдяки сучасним генеративним моделям користувач може створити зображення не лише за наявним шаблоном, а й з нуля – за допомогою текстового опису.

Генерація через локальний інтерфейс Stable Diffusion WebUI. Для демонстрації практичного застосування було використано встановлений на ПК інтерфейс Stable Diffusion WebUI (AUTOMATIC1111). Цей інструмент надає повний контроль над генерацією зображень: вибір нейромережевої моделі, параметри семплінгу, роздільність, кількість кроків, CFG Scale, а також підтримує prompt і негативний prompt, що дозволяє тонко керувати результатами.

Крок 1. Вибір моделі. Першим етапом є вибір нейромережевої моделі (checkpoint), яка буде використана для генерації. У цьому випадку було обрано модель photon_v1.safetensors, яка забезпечує високу деталізацію й універсальність у створенні персонажів.

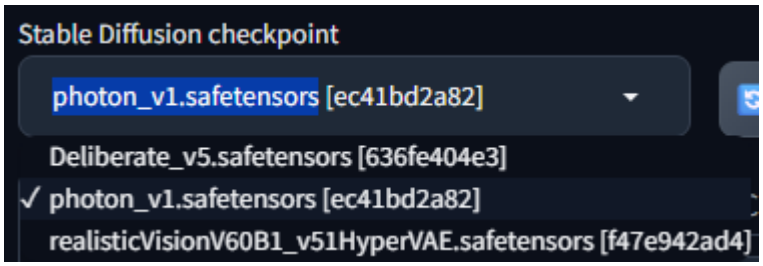


Рисунок 3.9 – Вибір моделі photon_v1 у локальному Stable Diffusion WebUI

Інтерфейс дозволяє завантажувати будь-які сумісні моделі у форматі .safetensors. Обрано модель photon_v1, що добре підходить для портретної генерації з елементами реалізму та стилізації.

Крок 2. Введення prompt та негативного prompt. У відповідних полях інтерфейсу були введені основні параметри генерації . Текстовий запит (prompt), який містить опис ключових візуальних елементів, які повинні з’явитись на зображенні (3.10). Також було введено негативний prompt, який вказує, чого потрібно уникати у зображенні (3.11).

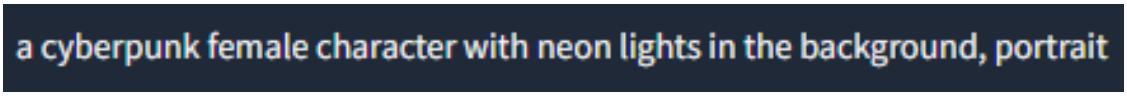


Рисунок 3.10 – Заповнення поля prompt

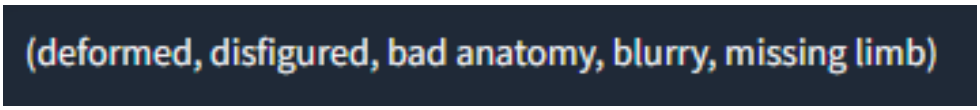


Рисунок 3.11 – Заповнення поля негативний prompt

Крок 3. Налаштування параметрів генерації. Перед запуском генерації було встановлені такі параметри (табл.3.1):

Таблиця 3.1 – Параметри генерації перед запуском

Параметр	Значення
Метод відбору зразків	DPM++ 2M
Кроки	20
Розмір зображення	720×720
CFG Scale	8
Hires. fix	увімкнено для кращої деталізації
Batch count / size	1 (по одному зображенню за раз)

Ці налаштування впливають на якість, чіткість, стиль і тривалість генерації. Змінюючи їх, можна досягати різних результатів.

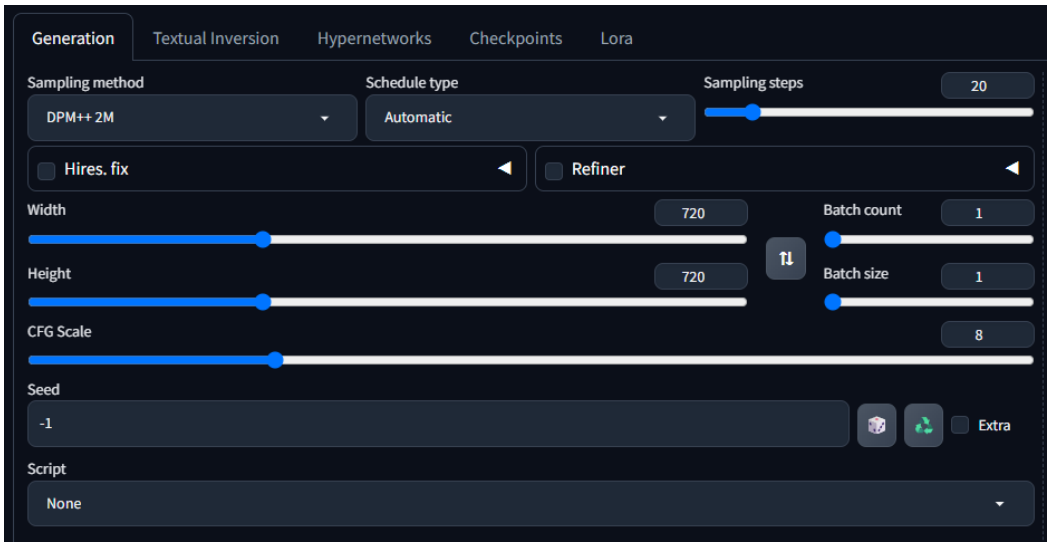


Рисунок 3.12 – Налаштування параметрів генерації перед запуском

У цьому вікні (рис. 3.12) задаються числові параметри, які регулюють поведінку моделі під час створення зображення.

Крок 4. Результат генерації. Після натискання кнопки Generate модель створила нове зображення (рис. 3.13) відповідно до текстового опису. Було згенеровано портрет персонажа в стилі кіберпанк, із яскравими неоновими вогнями та художнім фокусом на обличчі.

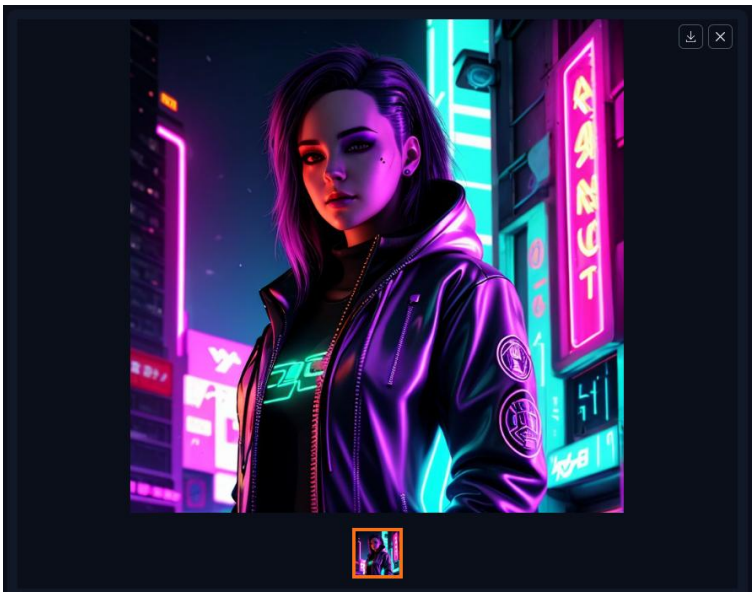


Рисунок 3.13 – Згенероване зображення за допомогою photon_v1.safetensors

На зображенні присутні всі ключові елементи prompt'у, включаючи стилістичне оформлення, світло, кольори й композицію.

Локальний запуск Stable Diffusion через WebUI дозволяє повністю контролювати процес генерації зображень – від вибору моделі до тонкої настройки результату. Це відкриває широкі можливості для індивідуального візуального дизайну, навіть без глибоких знань програмування чи глибинного навчання. Такий інструмент може ефективно використовуватись для генерації концепт-арту, ілюстрацій, банерів, обкладинок та іншого графічного контенту.

3.2 Експериментальне дослідження: порівняння архітектур у практиці

Практичне застосування нейронних мереж у сфері обробки зображень сьогодні не обмежується використанням одного універсального підходу. Навпаки, різні архітектури нейромереж – такі як Diffusion Models, Generative Adversarial Networks (GAN), Variational Autoencoders (VAE), Vision Transformers (ViT) – показують найкращі результати у вузькоспеціалізованих завданнях. У цьому підрозділі буде розглянуто три інструменти, які використовуються в практиці, кожен з яких базується на окремому класі архітектур. Порівнюючи їх, можна зробити висновки щодо доцільності використання кожного підходу у певних сценаріях.

Stable Diffusion – тестування моделі в режимі img2img на прикладі композитного зображення. У межах дослідження було проведено практичний експеримент із використанням генеративної моделі Stable Diffusion, яка поєднує в собі дифузійний механізм (denoising diffusion probabilistic model), трансформерну архітектуру (через CLIP) та варіаційний автоенкодер (VAE). Особливу увагу було зосереджено на режимі img2img, який дає змогу трансформувати вже наявне зображення згідно з заданим текстовим описом[19].

Мета експерименту: перевірити, наскільки ефективно модель Stable Diffusion здатна об’єднувати різнорідні вхідні об’єкти – людину, м’яч та штучно підставлений фон – в цілісну сцену, і водночас оцінити якість обробки рук, фону, світла та пропорцій. Це імітує реальний сценарій застосування в дизайні чи рекламній продукції, де часто виникає потреба в стилізації складених вручну композицій.

Вхідні матеріали. Для підготовки вхідного зображення було вручну поєднано три елементи:

– фотографія чоловіка з нейтральним виразом обличчя та відкритими руками (основа сцени)(рис.3.14);

– зображення футбольного м’яча(рис. 3.15);

– фон – футбольне поле під нічним освітленням(рис. 3.16).



Рисунок 3.14 – Фотографія чоловіка (Взято з Freepik)



Рисунок 3.15 – Зображення футбольного м’яча (Взято з Freepik)



Рисунок 3.16 – Фон футбольне поле (Взято з Freepik)



Рисунок 3.17 – Вхідне зображення (змонтовано вручну)

Зображення (рис. 3.17) було змонтоване за допомогою графічного редактора і збережене у високій якості для подальшої обробки в WebUI Stable Diffusion.

Параметри генерації показані в таблиці 3.2:

Таблиця 3.2 – Параметри обробки через функцію img2img перед запуском

Параметр	Значення
Метод відбору зразків	DPM++ 2M
Кроки	20
Розмір зображення	1280×720
CFG Scale	4
Сила деноїдизації	0.4
Seed: -1	-1 (випадковий)

Prompt, який описує зображення (рис. 3.18):

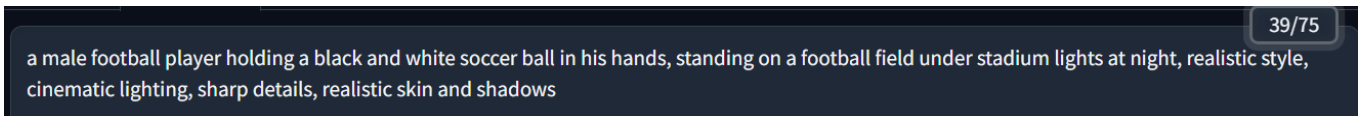


Рисунок 3.18 – Промт для img2img

Negative prompt який вказує, що не має бути показано на вихідному зображенні (рис. 3.19):

cartoon, painting, illustration, (worst quality, low quality, normal quality:2)

Рисунок 3.19 – Негативний промт для img2img



Рисунок 3.20 – Фінальний результат генерації

Результат обробки показано на рисунку 3.20. На ньому видно що программа Stable Diffusion здійснила стилізацію зображення, зберігши загальну композицію, але суттєво покращивши:

– реалістичність освітлення – відтінки та тіні стали більш природними;

– контури об'єктів – м’яч і руки мають чіткі обриси, м’яч виглядає інтегрованим у простір;

– риси обличчя – модель адаптувала їх під заданий образ (спортсмена), не змінюючи загальної пози;

– фон – футбольне поле і світло стали глибшими, менш "штучними".

Результат виглядає цілісно і може бути використаний у рекламному або презентаційному дизайні.

Виявлені особливості та обмеження:

– при низькому denoising strength (0.4) модель залишає композицію незмінною, але покращує її візуальні характеристики. Це оптимально для стилізації, а не повної заміни сцени.

– важливим виявився якісний негативний prompt – без нього модель могла додати артефакти або мультиплікаційні риси.

– незважаючи на змістовність запиту, модель не завжди ідеально розуміє просторову логіку – положення рук і м’яча іноді потребують додаткової корекції в графічному редакторі.

Висновки експерименту. Stable Diffusion у режимі img2img показала високу ефективність у задачі стилістичного об’єднання різнорідних об’єктів в одне цілісне зображення. Її можна успішно використовувати для:

– адаптації сцени під художній стиль;

– вирівнювання кольорів, світла та текстур;

– підготовки комерційних чи візуальних матеріалів без повної перекомпозиції сцени.

Такий експеримент підтверджує можливість застосування дифузійних моделей у реальних проєктах, зокрема у сфері візуального дизайну, маркетингу або навіть журналістики.

Застосування генеративної моделі Artbreeder для створення портретних зображень. Одним із прикладів практичного використання нейронних мереж у візуальному дизайні є генеративна платформа Artbreeder, побудована на основі архітектур GAN (Generative Adversarial Networks) та VAE (Variational Autoencoders). Цей інструмент дозволяє користувачам створювати нові зображення шляхом маніпуляцій у латентному просторі – тобто, змінюючи значення векторних ознак, що відповідають за риси обличчя, вік, емоції, колір волосся, стиль тощо.

Сервіс Artbreeder працює на основі заздалегідь натренованих моделей і надає інтерфейс типу "слайдери" (рис. 3.21), що дозволяє у зручній формі змінювати вихідне зображення. Користувач може редагувати готові портрети, схрещувати кілька зображень між собою або створювати нові, змінюючи десятки візуальних характеристик.

З метою демонстрації можливостей платформи було виконано створення персонажа жіночої статі для подальшого використання в умовному дизайнерському проєкті. Було обрано задачу формування зображення дівчини

азійської зовнішності з емоцією здивування, яка є релевантною для ілюстрацій, обкладинок або візуального контенту соціальних мереж.

Налаштування слайдерів у середовищі Artbreeder Splicer охоплювали такі параметри (табл. 3.3):

Таблиця 3.3 – Параметри для генерації фото у середовищі Artbreeder Splicer

Параметр	Значення
Стать	жіноча (gender = 1)
Вік	18 років
Тип волосся	темне (black hair = +11)
Емоційні параметри	здивування (surprised = +8)
Етнічна ознака	азійська зовнішність (asian = +7)
Видалення вторинних ознак	facial hair = -7

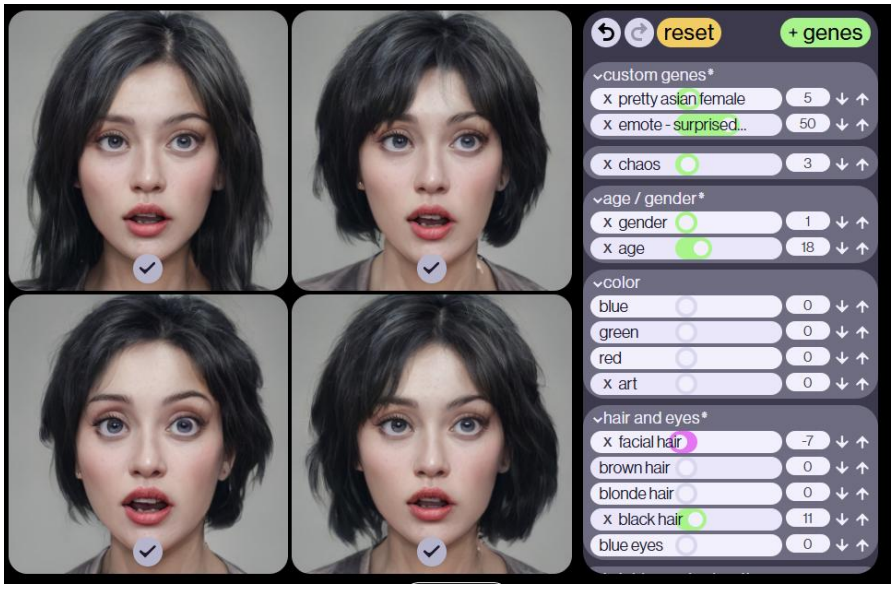


Рисунок 3.21 – Інтерфейс редагування ознак зображення в Artbreeder (Splicer)

Результатом генерації стало портретне зображення молодої дівчини зі щиро здивованим виразом обличчя (рис. 3.22). Зображення відзначається високим рівнем фотореалізму, деталізацією шкіри та природністю переданої емоції. Застосована генеративна модель забезпечила достатній контроль над ключовими рисами, зберігаючи при цьому узгодженість зображення та цілісність стилю.



Рисунок 3.22 – Згенероване портретне зображення в Artbreeder (архітектура GAN/VAE)

Практичне застосування такого підходу є доцільним у випадках, коли потрібно швидко отримати унікальні портрети для дизайн-завдань: маркетингових кампаній, візуального сторітелінгу, гейміфікаційних платформ або цифрових аватарів. Перевагою є простота інтерфейсу та можливість миттєвого візуального редагування без необхідності глибоких технічних знань.

Серед обмежень Artbreeder варто зазначити відсутність роботи з тілом, одягом чи композиційним фоном – платформа фокусується виключно на портретній зоні. Це частково обмежує її застосування для повноцінного сценічного дизайну, проте не знижує цінності для задач, пов’язаних із персоналізацією облич.

Таким чином, Artbreeder демонструє ефективність архітектур GAN/VAE у вузькоспеціалізованих дизайнерських задачах, пов’язаних зі створенням стилізованих образів та візуальних концептів.

Використання ClipDrop для локалізації та редагування зображень. У сучасних задачах дизайну та візуальної обробки зображень особливе місце посідають інструменти, що дозволяють швидко та точно редагувати контент на основі штучного інтелекту. Одним із таких є ClipDrop – онлайн-платформа, яка поєднує методи глибинного навчання та трансформерні архітектури для

автоматичного виділення об’єктів, очищення фону, заміни середовища або додавання нових елементів.

ClipDrop працює у браузері та не вимагає спеціальних технічних навичок. Усі функції реалізовані через інтуїтивно зрозумілий інтерфейс. Сервіс дозволяє завантажити будь-яке зображення, після чого на основі нейромережевого аналізу сцени система автоматично сегментує об’єкти, відокремлює фон або пропонує варіанти редизайну сцени. Основу технології становлять моделі на базі Vision Transformer (ViT) та векторного текстово-візуального узгодження (CLIP), які забезпечують узагальнене розуміння контенту.

Експериментальне дослідження: Заміна об'єкта. Для експерименту було використано зображення, на якому зображена червона чашка, розміщена на дерев'яній поверхні з фоном червоного кольору (рис. 3.23). Мета полягала у заміні чашки на білий чайник, що дозволяє оцінити здатність моделі не просто доповнювати простір, а семантично адаптувати новий об’єкт до наявної сцени.

На першому етапі виділяємо чашку за допомогою інструменту вільного виділення. Далі у зазначену зону вводимо текстовий опис нового об’єкта (рис. 3.24).



Рисунок 3.23 – Вхідне зображення з чашкою на дерев'яній поверхні

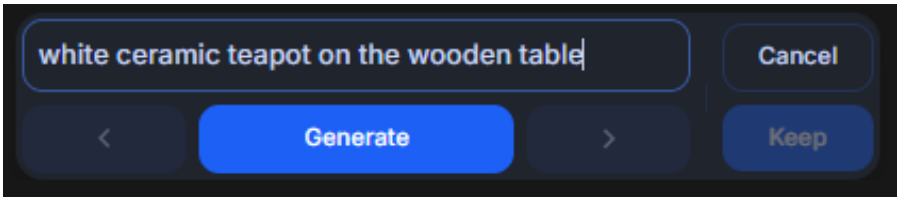


Рисунок 3.24 – Промт нового об’єкта (білий чайник)

Після цього система виконує генерацію нового зображення із заміщенням об'єкта відповідно до візуального контексту (рис. 3.25).



Рисунок 3.25 – Результат заміни чашки на білий чайник за допомогою ClipDrop

Як видно з результату, модель коректно врахувала:

– положення об’єкта на поверхні (збережено перспективу та напрям тіні);

– освітлення сцени (чайник освітлений так само, як попередня чашка);

– поєднання кольорів (новий об’єкт не порушує кольорову гармонію сцени).

Це свідчить про те, що трансформерна модель не лише оперує текстурою, але й виконує семантичне узгодження об’єктів і фону.

Аналіз результатів. Практична ефективність ClipDrop у задачах генеративного редагування полягає у можливості інтерактивно модифікувати зображення, не використовуючи спеціалізовані графічні редактори. Модель здатна:

– виконувати заміщення будь-якого об’єкта, не залишаючи слідів редагування;

– узгоджувати геометрію, освітлення та кольори з іншими частинами сцени;

– адаптувати різні типи контенту (предмети, ландшафти, меблі тощо) відповідно до призначення користувача.

Висновок експерименту. Функція Generative Fill у ClipDrop є прикладом ефективного застосування трансформерних моделей для семантичного

редагування візуального контенту. Вона дозволяє адаптивно змінювати об’єкти на зображеннях, зберігаючи цілісність сцени, що особливо актуально у сфері дизайну, електронної комерції та створення візуального контенту. Проведений експеримент із заміною чашки на білий чайник підтверджує здатність моделі до контекстно обґрунтованої генерації та показує її високу практичну цінність.

3.3 Аналіз отриманих результатів

У рамках розділу 3 були реалізовані експерименти з використанням трьох архітектур нейронних мереж: дифузійної моделі (Stable Diffusion), генеративної змагальної мережі (Artbreeder) та трансформерної моделі (Segment Anything + ClipDrop). Метою стало не лише оцінити технічні можливості кожної з них, а й зрозуміти, як ці підходи працюють у реальних практичних завданнях, які виникають у сфері комп’ютерного зору.

Для зручності порівняння всі три експерименти були виконані на базі однієї і тієї ж композиції: зображення чоловіка, який тримає футбольний м’яч на фоні футбольного поля (рис. 3.17). Такий підхід дозволив створити єдину референтну точку, на основі якої можна оцінити поведінку кожної архітектури в однакових умовах.

Під час аналізу увага зосереджувалась на таких елементах:

– наскільки добре модель може об’єднати різні об’єкти в єдину сцену;

– як точно вона передає риси обличчя, анатомію рук, положення предметів;

– чи правильно спрацьовує інтеграція фону, і чи узгоджуються елементи між собою;

– наскільки гнучке і кероване управління генерацією;

– чи придатний отриманий результат до реального використання: дизайну, реклами, візуального оформлення.

Таблиця 3.4 – Порівняння результатів експерименту

Завдання / аспект	Stable Diffusion (Diffusion)	Artbreeder (GAN/VAE)	SAM + ClipDrop (Transformer)
Об’єднання об’єктів в одну сцену	Успішно об’єднав людину, м’яч і фон у цілісну сцену, зберіг композицію та атмосферу	Немає підтримки сцени – тільки генерація обличчя, неможливо з'єднати об’єкти	Не виконує генерацію – лише дозволяє окремо редагувати або перемістити об’єкти
Обробка обличчя	Загальні риси збережено, але спостерігались дрібні помилки у деталях (наприклад, симетрія очей або шкіра)	Якісно змінює вираз, вік, стать, колір – ідеально для контролю облич	Не редагує обличчя, не надає інструментів для зміни рис
М’яч у руках	М’яч виглядає інтегрованим, з правильним освітленням, але може деформуватись при сильному denoising	Не підтримується взагалі – жодного предмета в руках створити не можна	Можна вирізати або вставити м’яч, але не стилізувати чи освітити
Руки, пальці, положення	Часто спостерігаються анатомічні помилки (зайві пальці, злиття), особливо в області кистей	Не підтримується – генерація обмежена до рівня шиї	Можна вручну замінити, але не виправити чи деталізувати
Фон і освітлення	Вдале відтворення фону: нічна сцена, освітлення, глибина поля – усе виглядає узгоджено	Не працює з фоном – лише однотонний або розмитий	Фон можна змінити вручну, але стилізація не підтримується
Контроль користувача	Повна гнучкість через prompt, denoising, seed, масштабування	Прості слайдери для контролю параметрів (вираз, вік, стать тощо)	Мінімальний контроль: лише вручну виділення, вирізання, заміна
Придатність до дизайну / комерції	Результат можна без додаткової обробки використати в постері або прев’ю	Можна використати для аватарів або персонажів, але не в сцена-рівнях	Чудово для редизайну окремих об’єктів або вирізання для колажу

Порівняльний аналіз архітектур:

1. *Diffusion (Stable Diffusion).* Цей підхід виявився найбільш збалансованим: він не лише з'єднав елементи у логічну композицію, але й забезпечив відповідний стиль і настрій сцени. Проте він має слабке місце – деталізація рук, облич, мілких об'єктів, які часто мають спотворення. Модель дає чудові результати там, де важлива цілісність сцени, освітлення, глибина простору;

2. *GAN / VAE (Artbreeder).* Artbreeder на базі GAN/VAE показав найвищу якість в області морфінгу обличчя. Інструмент дозволяє легко змінювати вік, стать, емоції, світло. Але повноцінні сцени або взаємодія з іншими об'єктами (руки, предмети, фон) – недоступні. Це хороший варіант для генерації аватарів або кастомізованих персонажів, але не сцен;

3. *Transformer (SAM + ClipDrop).* Цей інструмент не створює зображення, але є потужним для локального редагування. Він ідеальний для завдань, де потрібно вирізати, перемістити або замінити окремі частини сцени. Наприклад, вручну замінити фон, вставити інший предмет, домалювати щось. Його не можна вважати генератором зображень, але як інструмент постобробки він дуже корисний. Під час будь-якого з цих етапів здійснюється комунікація із замовником та різноманітними фахівцями для усунення проблемних питань чи своєчасного внесення змін. Дотримання цих етапів допоможе зробити мережу більш якісною та зрозумілою в обслуговуванні. Оскільки наявність озвучених схем, таблиць та документів дозволяє якісніше та швидше виправляти неполадки й підтримувати мережу в робочому стані. Тому, що системному адміністратору не потрібно буде самому складати ці схеми й шукати потрібні пристрої та підключення навмання.

Найбільш універсальним рішенням для дизайну сцен виявився Stable Diffusion. У той час як Artbreeder забезпечує найкращу якість у сфері портретів, а SAM + ClipDrop – ефективний як редактор контенту, справжній потенціал відкривається саме при поєднанні цих інструментів у єдиному робочому процесі. Такий підхід дозволяє компенсувати обмеження кожного з них та отримати повноцінний результат, який відповідає професійним вимогам.

3.4 Поєднання нейромережевих моделей для реалізації дизайнерського завдання

У цьому підрозділі демонструється, як можна поєднати різні архітектури нейронних мереж для розв’язання реального практичного завдання у сфері дизайну – створення графічної обкладинки для відео на платформу YouTube. Такий кейс є прикладом інтегрованого використання інструментів на основі diffusion-моделей, трансформерів та генеративно-змагальних мереж, і показує переваги гібридного підходу в умовах творчої роботи з зображеннями.

Постановка задачі. У якості технічного завдання клієнт надіслав наступний запит: “Обкладинка має бути присвячена проєкту Paws – мобільному застосунку, аналогічному до Notcoin. Необхідно використати образ людини, яка асоціюється з багатством (наприклад, Ілон Маск), додати позаду купюри в доларах, логотип проєкту у вигляді монети. Фон – інтер’єр кімнати. Текст: PAWS – Халявні 100$ на акаунт.”

Згідно з цим завданням, було реалізовано повний цикл створення зображення, який включає етапи підготовки зображень, генерації фону, локального редагування, компоновки елементів та стилізації сцени.

1. Пошук і підготовка зображень. На першому етапі було зібрано наступні вхідні графічні елементи (рис.3.26):

– фотографія Ілона Маска;
– зображення доларових купюр;
– логотип проєкту Paws у вигляді монети.

Для подальшої компоновки було необхідно прибрати фон із кожного зображення. Для цього використано інструмент ClipDrop, зокрема функцію автоматичного видалення заднього фону (рис. 3.27).



1_dollar.png



960px-Elon_Musk_Royal_Society_cr.png



Dollars.png



paws_paws_telegram.png

Рисунок 3.26 – Вхідні зображення



1_dollar.png



960px-Elon_Musk_Royal_Society_cr.png



Dollars.png



paws_paws_telegram.png

Рисунок 3.27 – Вихідні зображення без заднього фону

2. Генерація фону в Stable Diffusion. Наступним кроком стала генерація фотогенічного інтер'єру кімнати, яка буде використовуватися як фон. Ця задача виконувалась у Stable Diffusion WebUI (AUTOMATIC1111) за допомогою методу txt2img, що базується на дифузійному моделюванні.

Використаний prompt для генерації фону (рис. 3.28):

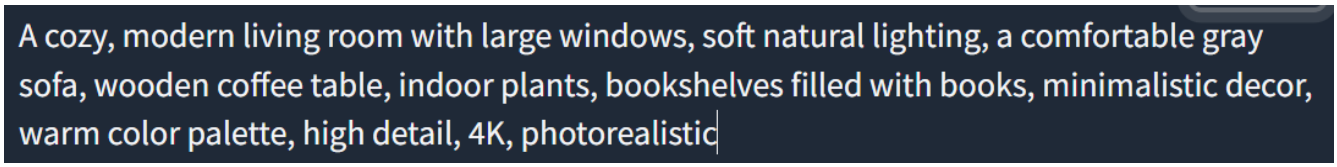


Рисунок 3.28 – Промт для генерації фону через txt2img

Negative prompt для генерації фону (рис. 3.29):

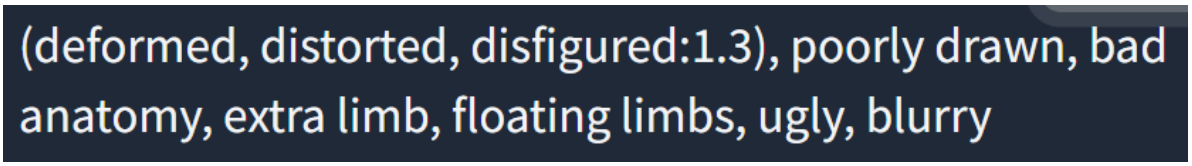


Рисунок 3.29 – Негативний промт для генерації фону через txt2img

Налаштування генерації показані в таблиці 3.5:

Таблиця 3.5 – Параметри генерації перед запуском

Параметр	Значення
Метод відбору зразків	DPM++ 2M
Кроки	20
Розмір зображення	1280×720
CFG Scale	8

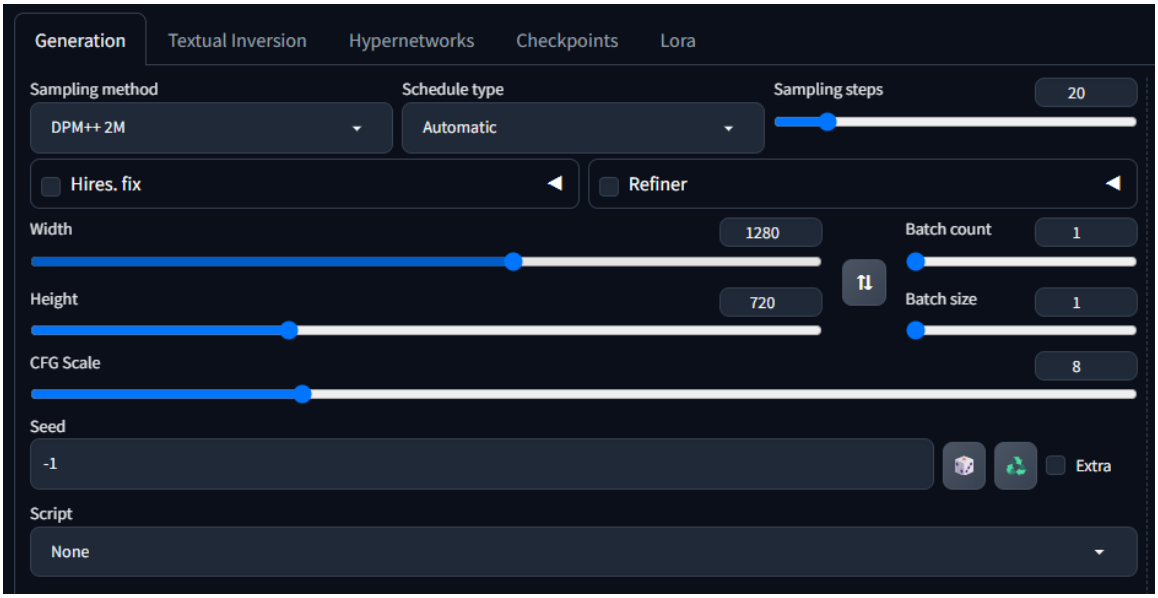


Рисунок 3.30 – Налаштування генерації в Stable Diffusion



Рисунок 3.31 – Згенерований фон кімнати

3. Локальна обробка зображення через ClipDrop. Оскільки в згенерованому фоні були небажані елементи – такі як рослини, що привертали увагу, – було виконано локальне редагування фону за допомогою функції Cleanup в сервісі ClipDrop.



Рисунок 3.32 – Виділення об’єктів для видалення



Рисунок 3.33 – Результат очищення сцени

Таким чином, було створено чистий, готовий до вставки композиційних елементів фон, вільний від зайвих деталей.

4. Компоновка зображення в графічному редакторі. У графічному редакторі Adobe Photoshop всі підготовлені елементи були зібрані в єдину композицію (рис. 3.30) – обличчя, гроші, логотип та фон. Цей етап дозволив побачити попередній макет обкладинки.



Рисунок 3.34 – Композиція без обробки

5. Стилізація сцени через Stable Diffusion (img2img). Щоб фіналізувати візуальний стиль, композиція була пропущена через режим img2img у Stable Diffusion для стилізації та покращення деталізації сцени.

Prompt для фінальної обробки (рис. 3.35):

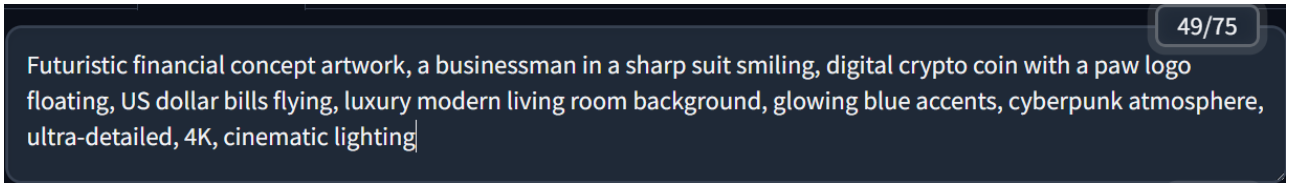


Рисунок 3.35 – Промт для фінальної обробки через img2img

Negative prompt для фінальної обробки (рис.3.36):

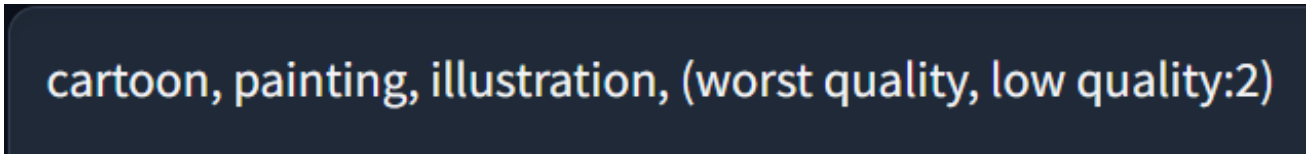


Рисунок 3.36 – Негативний промт для фінальної обробки через img2img

Таблиця 3.6 – Параметри фінальної обробки за допомогою функції img2img

Параметр	Значення
Метод відбору зразків	DPM++ 2M
Тип розкладу	Karras
Кроки	20
Розмір зображення	1920×1080
CFG Scale	4
Сила деноїдизації	0.4
Seed: -1	-1 (випадковий)

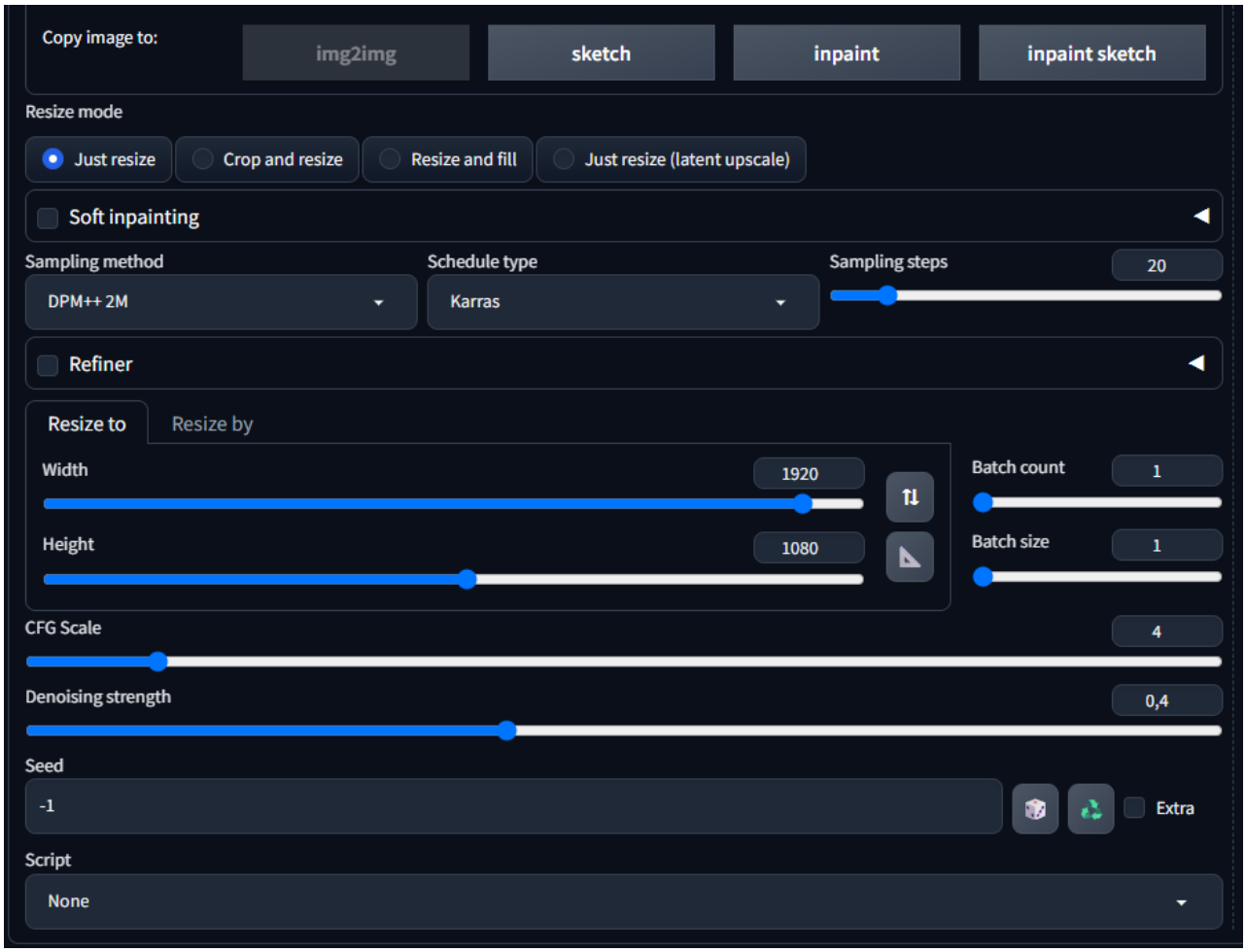


Рисунок 3.37 – Налаштування генерації



Рисунок 3.38 – Зображення після обробки нейромережею

У результаті було отримано зображення високої якості з гармонійним поєднанням кольору, світла та композиції (рис.3.38). Однак, було помічено зміну рис обличчя та появу дрібних артефактів, що потребували корекції.

6. Фінальна обробка в Photoshop. Фінальний етап включав в себе:

– повернення зображення до Photoshop;
– ручне видалення артефактів;
– вставлення оригінального обличчя Маска;
– додавання текстового шару (PAWS – Халявні 100$ на акаунт).



Рисунок 3.39 – Готова обкладинка для YouTube

Демонстрований приклад підтверджує ефективність поєднання нейронних мереж різної архітектури. За рахунок гнучкого розподілу завдань між інструментами вдалося:

– значно скоротити час на створення візуального контенту (30 хвилин замість 2 годин);

– підвищити якість обробки фону, об'єктів та стилізації сцени;

– забезпечити креативну гнучкість завдяки можливості швидкої генерації варіантів.

Отриманий результат підтверджує практичну цінність гібридного підходу, де diffusion-моделі, GAN і трансформери застосовуються спільно – кожен для своєї підзадачі. Такий підхід має перспективи широкого впровадження в маркетингу, дизайні та медіа-виробництві.

ВИСНОВКИ

Нейронні мережі глибинного навчання сьогодні є ключовим інструментом у сфері комп'ютерного зору. Їх активне впровадження у практичні задачі обробки зображень демонструє ефективність новітніх архітектур, зокрема дифузійних моделей, генеративно-змагальних мереж та трансформерів.

У рамках роботи було проведено систематизований огляд основних архітектур: CNN, GAN, Diffusion та Transformer, а також розглянуто методи навчання, оптимізації та техніки, які забезпечують покращення точності та стабільності моделей. Окрему увагу приділено сучасним інструментам генерації та обробки зображень, зокрема Stable Diffusion, Artbreeder, Segment Anything та ClipDrop.

Проведені експерименти дозволили на практиці оцінити можливості кожного з підходів у вирішенні задач класифікації, генерації, сегментації та локального редагування зображень. Для аналізу було використано єдиний вхідний приклад – композитне зображення з людиною, м'ячем і фоном футбольного поля. Це дозволило зіставити результати між архітектурами в однакових умовах.

За підсумками дослідження встановлено, що жодна архітектура не є універсальною. Кожна з них показала високі результати у своїй спеціалізованій задачі. Diffusion-моделі (Stable Diffusion) дозволяють досягати реалістичних і стилізованих зображень, GAN/VAE (Artbreeder) надають гнучкий контроль при генерації облич, а трансформери (SAM + ClipDrop) забезпечують ефективні інструменти для локальної обробки сцен.

У ході практичної частини було доведено, що найбільш ефективний результат досягається при поєднанні різних архітектур. Такий підхід дозволяє компенсувати слабкі сторони кожного методу й реалізовувати складні задачі з високою точністю, стабільністю та творчим контролем. Наприклад, для створення обкладинки або рекламного банера можна поетапно застосовувати інструменти,

що базуються на різних моделях: генерація сцени через дифузію, корекція обличчя через GAN, редагування фону через трансформерну сегментацію.

Результати цієї роботи підтверджують доцільність використання гібридного підходу в сучасному дизайні, візуальній комунікації, маркетингу та медіа. Інвестування в розвиток подібних архітектур і технологій є перспективним напрямом, який має значний потенціал для подальших прикладних рішень у сфері штучного інтелекту та обробки зображень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глибоке (глибинне) навчання. Нейронні мережі. Уроки для початківців. W3Schools українською. W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів. URL: https://w3schoolsua.github.io/ai/ai_neural_networks.html#gsc.tab=0 (дата звернення: 16.04.2025).
2. Дізнайтеся, як глибинне навчання змінює технології, сприяє інноваціям. Agiliway - Custom Software Development Company. URL: https://www.agiliway.com/uk-ua/hlybynne-navchannya/ (дата звернення: 15.04.2025).
3. Згорткові нейронні мережі (частина 2) – IT Master - електроніка та програмування. Головна – IT Master - електроніка та програмування. URL: https://itmaster.biz.ua/programming/vision/cnns2.html (дата звернення: 11.04.2025).
4. Овдієнко С. Д., Бученко І. А. Генерація зображень за допомогою нейронних мереж: сучасні підходи та можливості застосування. Застосування програмного забезпечення в ІКТ : VI Всеукр. науково-техн. конф., м. Київ, 24 квіт. 2025 р. Київ, 2025. С. 348–350. URL: https://duikt.edu.ua/uploads/p_2779_56719466.pdf.
5. Овдієнко С. Д., Бученко І. А. Підходи до побудови нейронних мереж для обробки зображень. Сучасні інтелектуальні інформаційні технології в науці та освіті : V Всеукр. науково-практ. конф., м. Київ, 15 трав. 2025 р. Київ, 2025. Подано до друку.
6. A novel deep unsupervised learning-based framework for optimization of truss structures / H. T. Mai та ін. Engineering with Computers. 2022. URL: https://doi.org/10.1007/s00366-022-01636-3 (дата звернення: 14.05.2025).
7. Aghdam H. H., Heravi E. J. Guide to convolutional neural networks: a practical application to traffic-sign detection and classification. Springer, 2017. 305 с.

8. Batenburg K. J., Sijbers J. Adaptive thresholding of tomograms by projection distance minimization. Pattern recognition. 2009. Т. 42, № 10. С. 2297–2305. URL: https://doi.org/10.1016/j.patcog.2008.11.027 (дата звернення: 11.04.2025).

9. Belcic I., Stryker C. What is supervised learning? | IBM. IBM - United States. URL: https://www.ibm.com/think/topics/supervised-learning (дата звернення: 13.05.2025).

10. Brital A. GoogLeNet CNN architecture explained (inception V1). Medium. URL: https://medium.com/@AnasBrital98/googlenet-cnn-architecture-explained-inception-v1-225ae02513fd (дата звернення: 26.04.2025).

11. Dhillon A., Verma G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in artificial intelligence. 2019. Т. 9, № 2. С. 85–112. URL: https://doi.org/10.1007/s13748-019-00203-0 (дата звернення: 12.04.2025).

12. GeeksforGeeks. Categorical Cross-Entropy in Multi-Class Classification - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/categorical-cross-entropy-in-multi-class-classification/ (дата звернення: 11.04.2025).

13. George Karimpanal T., Bouffanais R. Self-organizing maps for storage and transfer of knowledge in reinforcement learning. Adaptive behavior. 2018. Т. 27, № 2. С. 111–126. URL: https://doi.org/10.1177/1059712318818568 (дата звернення: 19.05.2025).

14. Gershgorn D. The data that transformed AI research–and possibly the world. Quartz. URL: https://web.archive.org/web/20170727173117/https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/ (дата звернення: 12.04.2025).

15. Gradient-based learning applied to document recognition / Y. Lecun та ін. Proceedings of the IEEE. 1998. Т. 86, № 11. С. 2278–2324. URL: https://doi.org/10.1109/5.726791 (дата звернення: 19.05.2025).

16. Home | Department of Mathematics. URL: https://math.berkeley.edu/~sethian/2006/Applications/Medical_Imaging/artery.html (дата звернення: 09.04.2025).

17. Improvement of the model of object recognition in aero photographs using deep convolutional neural networks / V. Slyusar та ін. Eastern-European journal of enterprise technologies. 2021. Т. 5, № 2 (113). С. 6–21. URL: https://doi.org/10.15587/1729-4061.2021.243094 (дата звернення: 26.04.2025).

18. Inception-v4, inception-resnet and the impact of residual connections on learning / C. Szegedy та ін. Proceedings of the AAAI conference on artificial intelligence. 2017. Т. 31, № 1. URL: https://doi.org/10.1609/aaai.v31i1.11231 (дата звернення: 27.04.2025).

19. Introducing stable diffusion 3.5 – stability AI. Stability AI. URL: https://stability.ai/news/introducing-stable-diffusion-3-5 (дата звернення: 14.04.2025).

20. Kashyap P. Understanding rmsprop: a simple guide to one of deep learning's powerful optimizers. Medium. URL: https://medium.com/@piyushkashyap045/understanding-rmsprop-a-simple-guide-to-one-of-deep-learnings-powerful-optimizers-403baeed9922 (дата звернення: 09.04.2025).

21. Keras documentation: RMSprop. Keras: Deep Learning for humans. URL: https://keras.io/2/api/optimizers/rmsprop/ (date of access: 09.04.2025).

22. Khoma Y., Bench A. Comparative analysis of the specialized software and hardware for deep learning algorithms. Computer systems and network. 2019. Т. 1, № 1. С. 97–102. URL: https://doi.org/10.23939/csn2019.01.097 (дата звернення: 15.05.2025).

23. Kiwiel K. C. Convergence and efficiency of subgradient methods for quasiconvex minimization. Mathematical programming. 2001. Т. 90, № 1. С. 1–25. URL: https://doi.org/10.1007/pl00011414 (дата звернення: 18.04.2025).

24. Mean squared error (MSE). Probability, Statistics & Random Processes | Free Textbook | Course. URL:

https://www.probabilitycourse.com/chapter9/9_1_5_mean_squared_error_MSE.php (дата звернення: 05.04.2025).

25. Opitz D., Maclin R. Popular ensemble methods: an empirical study. Journal of artificial intelligence research. 1999. Т. 11. С. 169–198. URL: https://doi.org/10.1613/jair.614 (дата звернення: 01.06.2025).

26. Overfitting. 403 Forbidden. URL: https://web.ma.utexas.edu/users/mks/statmistakes/ovefitting.html (дата звернення: 29.04.2025).

27. Sahu S. K., Mokhade A., Bokde N. D. An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: recent progress and challenges. Applied sciences. 2023. Т. 13, № 3. С. 1956. URL: https://doi.org/10.3390/app13031956 (дата звернення: 15.05.2025).

28. Supervised determined source separation with multichannel variational autoencoder / H. Kameoka та ін. Neural computation. 2019. Т. 31, № 9. С. 1891–1914. URL: https://doi.org/10.1162/neco_a_01217 (дата звернення: 18.04.2025).

29. Understanding binary cross-entropy / log loss: a visual explanation | Towards Data Science. Towards Data Science. URL: https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a/ (дата звернення: 18.04.2025).

30. Vision transformers or convolutional neural networks? Both! | towards data science. Towards Data Science. URL: https://towardsdatascience.com/vision-transformers-or-convolutional-neural-networks-both-de1a2c3c62e4/ (дата звернення: 29.04.2025).

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ



Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

на тему: «Підходи до побудови нейронних мереж для зображень»

Виконав студент групи КІД-41
Овдієнко Сергій Дмитрович

Керівник кваліфікаційної роботи:
Бученко Ігор Анатолійович, старший викладач кафедри КІ

Київ – 2025

Мета роботи – Дослідити підходи до побудови, навчання та практичного застосування нейронних мереж для обробки зображень, проаналізувати сучасні архітектури, а також реалізувати експериментальні приклади з використанням популярних AI-сервісів

Об`єкт дослідження – Нейронні мережі глибинного навчання, що використовуються для обробки, аналізу та генерації зображень

Предмет дослідження – Архітектурні підходи (CNN, GAN, Transformer, Diffusion), методи навчання та інструменти застосування нейромереж у задачах комп'ютерного зору

Актуальність обраної теми

У наш час нейронні мережі стали основою більшості сучасних систем штучного інтелекту, особливо в галузі комп'ютерного зору. Обробка зображень за допомогою глибоких моделей використовується у медицині, безпеці, промисловості, цифровому дизайні та багатьох інших сферах. Ефективність таких систем значною мірою залежить від вибору архітектури нейронної мережі, способу її навчання та адаптації до конкретних задач. При цьому часто виникає необхідність досягти високої точності за обмежених ресурсів, наприклад, при розгортанні моделей на мобільних або вбудованих пристроях. У таких умовах актуальним є не лише аналіз різних архітектур, а й вивчення можливостей їх поєднання, оптимізації та практичного застосування для реальних задач, таких як класифікація, сегментація чи генерація зображень.

Глибинне навчання

Глибинне навчання (deep learning) — це підхід у сфері штучного інтелекту, що базується на використанні багаторівневих нейронних мереж, які автоматично витягують ознаки з даних.

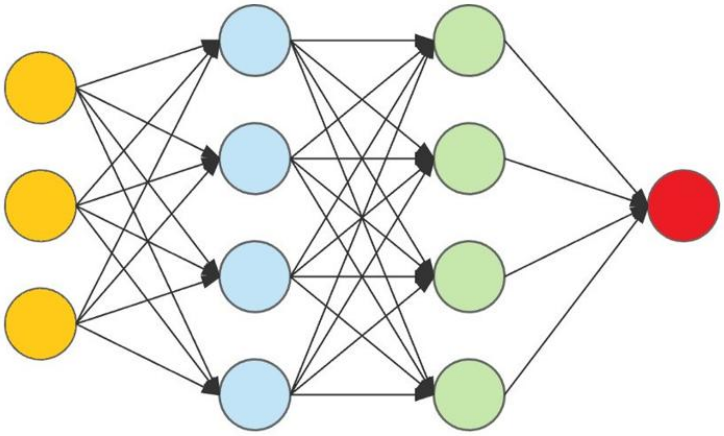


Рис. 1.1 – Схематична структура згорткової нейронної мережі (CNN)

Класичні архітектури

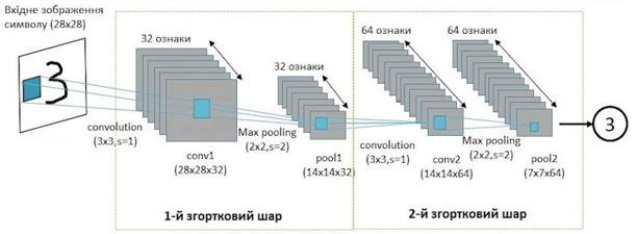


Рис. 1.2 – Архітектура LeNet-5

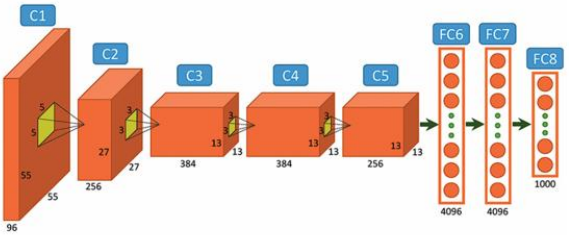


Рис. 1.3 – Архітектура AlexNet

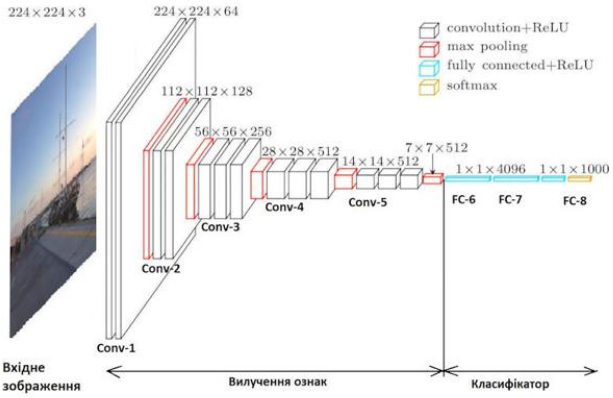


Рис. 1.4 – Архітектура VGG-16

Сучасні архітектури

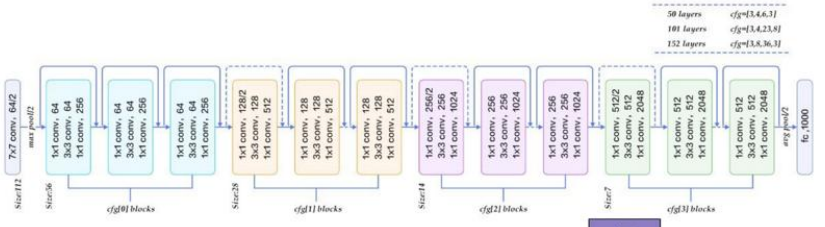


Рис. 1.5 – Архітектура ResNet

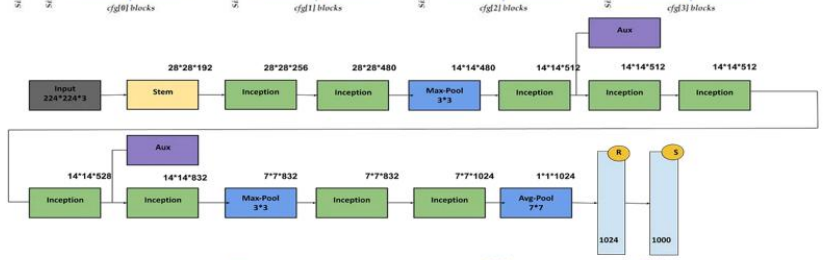


Рис. 1.6 – Архітектура мережі Inception (GoogLeNet)

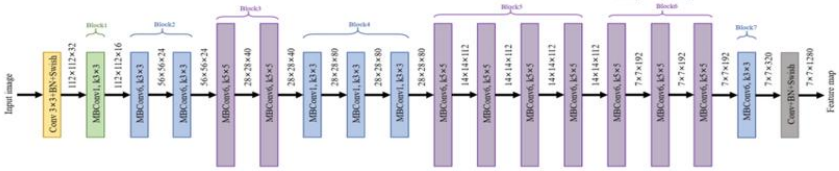


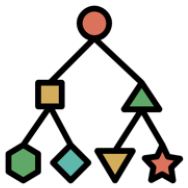
Рис. 1.7 – Архітектура мережі EfficientNet-B0

Порівняння поширених архітектур

Архітектура	Top-1 точність (ImageNet)	Параметри (млн)	FLOPs (млрд)	Придатність до розгортання
AlexNet	~57 %	60	0.72	Проста, але застаріла
VGG-16	~71 %	138	15.5	Високоточна, неефективна
ResNet-50	~76 %	25.6	4.1	Баланс точності й ефективності
Inception-v3	~78 %	23.5	5.7	Висока ефективність
EfficientNet-B0	~77 %	5.3	0.39	Оптимальна для мобільних пристроїв

Методи навчання нейронних мереж

- навчання з учителем (Supervised Learning);
- навчання без учителя (Unsupervised Learning);
- трансферне навчання (Transfer learning);







Функції втрат та оптимізація

Функції втрат:

1. Крос-ентропія (Categorical Cross-Entropy)(2.1) $\mathcal{L} = -\sum_{i=1}^{C} y_i \log(p_i)$
2. Середньоквадратична помилка (Mean Squared Error, MSE)(2.3) $\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
3. Бінарна крос-ентропія (Binary Cross-Entropy)(2.4) $L = -[y \cdot \log(p) + (1-y) \cdot \log(1-P)]$

Алгоритми оптимізації:

1. Стохастичний градієнтний спуск (SGD – Stochastic Gradient Descent)(2.5) $\theta = \theta - \eta \cdot \nabla\mathcal{L}$
2. RMSprop (Root Mean Square Propagation)(2.6-2.7) $v_t = \gamma v_{t-1} + (1-\gamma)(\nabla\mathcal{L})^2 \quad \theta = \theta - \frac{\eta}{\sqrt{v_t+\epsilon}} \cdot \nabla\mathcal{L}$
3. Adam (Adaptive Moment Estimation)(2.8-2.10) $m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla\mathcal{L} \quad v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla\mathcal{L})^2$

$$\theta = \theta - \eta \cdot \frac{m_t}{\sqrt{v_t}+\epsilon}$$

Класифікація через Teachable Machine

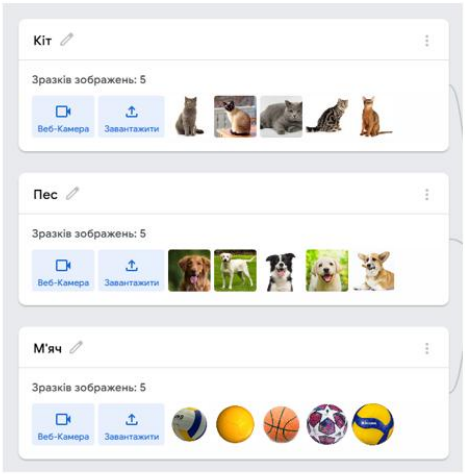


Рис. 3.1 – Завантаження зображень до класів

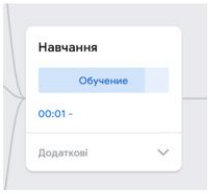


Рис. 3.2 – Процес навчання моделі у Teachable Machine

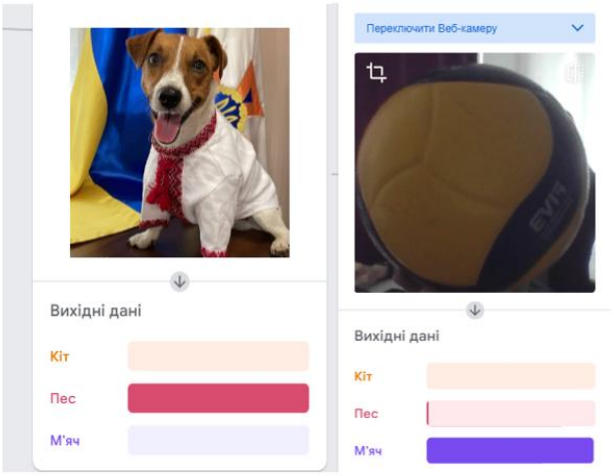


Рис. 3.3 – Результат класифікації нового зображення

Сегментація через SegmentAnything Model



Рис. 3.5 – Завантажене зображення для сегментації в SAM

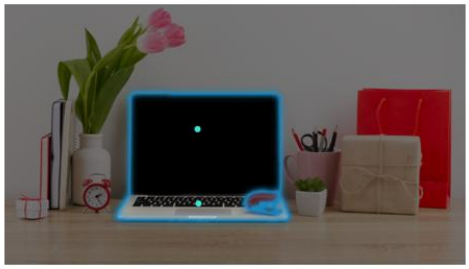


Рис. 3.6 – Результат виділення об’єкта (ноутбука) у SAM

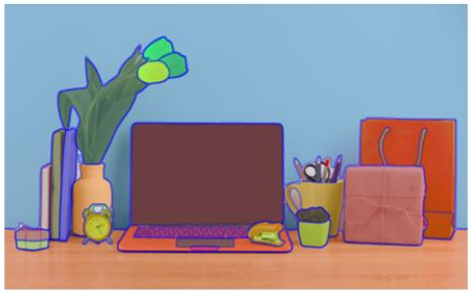


Рис. 3.7 – Повна сегментація зображення за допомогою SAM

Генерація зображення через Stable Diffusion

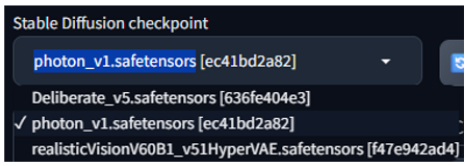


Рис. 3.9 – Вибір моделі photon_v1 у локальному Stable Diffusion WebUI

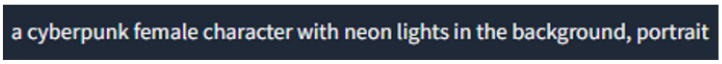


Рис. 3.10 – Заповнення поля prompt

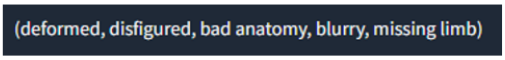


Рис. 3.11 – Заповнення поля негативний prompt

Табл. 3.1 – Параметри генерації перед запуском

Параметр	Значення
Метод відбору зразків	DPM++ 2M
Кроки	20
Розмір зображення	720×720
CFG Scale	8
Hires. fix	увімкнено для кращої деталізації
Batch count / size	1 (по одному зображенню за раз)

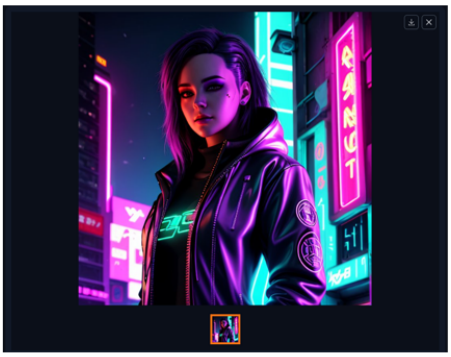


Рис. 3.11 – Налаштування параметрів генерації перед запуском

Порівняння архітектур на практиці

Stable Diffusion (Diffusion)



Рис. 3.16 – Вхідне зображення (змонтовано вручну)



Рис. 3.19 – Фінальний результат генерації

Artbreeder (GAN/VAE)

Табл. 3.3 – Параметри для генерації фото у середовищі Artbreeder Splicer

Параметр	Значення
Стать	жіноча (gender = 1)
Вік	18 років
Тип волосся	темне (black hair = +11)
Емоційні параметри	здивування (surprised = +8)
Етнічна ознака	азійська зовнішність (asian = +7)
Видалення вторинних ознак	facial hair = -7

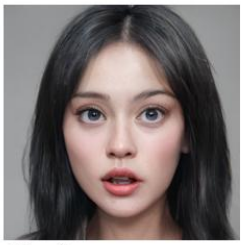


Рис. 3.19 – Згенероване портретне зображення в Artbreeder (архітектура GAN/VAE)

ClipDrop (Transformer)



Рис. 3.20 – Вхідне зображення з чашкою на дерев'яній поверхні

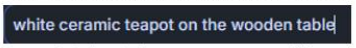


Рис. 3.21 – Промт нового об’єкта (білий чайник)



Рис. 3.22 – Результат заміни чашки на білий чайник

Аналіз результатів

Референтною точкою є зображення чоловіка, який тримає футбольний м’яч на фоні футбольного поля.

Завдання / аспект	Stable Diffusion (Diffusion)	Artbreeder (GAN/VAE)	SAM + ClipDrop (Transformer)
Об’єднання об’єктів в одну сцену	Успішно об’єднав людину, м’яч і фон у цілісну сцену, зберіг композицію та атмосферу	Немає підтримки сцени — тільки генерація обличчя, неможливо з'єднати об’єкти	Не виконує генерацію — лише дозволяє окремо редагувати або перемістити об’єкти
Обробка обличчя	Загальні риси збережено, але спостерігались дрібні помилки у деталях (наприклад, симетрія очей або шкіра)	Якісно змінює вираз, вік, стать, колір — ідеально для контролю облич	Не редагує обличчя, не надає інструментів для зміни рис
Обробка м’яча у руках	М’яч виглядає інтегрованим, з правильним освітленням, але може деформуватись при сильному denoising	Не підтримується взагалі — жодного предмета в руках створити не можна	Можна вирізати або вставити м’яч, але не стилізувати чи освітити
Руки, пальці, положення	Часто спостерігаються анатомічні помилки (зайві пальці, злиття), особливо в області кистей	Не підтримується — генерація обмежена до рівня шиї	Можна вручну замінити, але не виправити чи деталізувати
Фон і освітлення	Вдале відтворення фону: нічна сцена, освітлення, глибина поля — усе виглядає узгоджено	Не працює з фоном — лише однотонний або розмитий	Фон можна змінити вручну, але стилізація не підтримується
Контроль користувача	Повна гнучкість через prompt, denoising, seed, масштабування	Прості слайдери для контролю параметрів (вираз, вік, стать тощо)	Мінімальний контроль: лише вручну виділення, вирізання, заміна
Придатність до дизайну / комерції	Результат можна без додаткової обробки використати в постері або прев’ю	Можна використати для аватарів або персонажів, але не в сцена-рівнях	Чудово для редизайну окремих об’єктів або вирізання для колажу

Реалізація дизайнерського завдання



Рис. 3.23 – Вихідні зображення без заднього фону



Рис. 3.27 – Згенерований фон кімнати



Рис. 3.29 – Результат очищення сцени



Рис. 3.30 – Композиція без обробки



Рис. 3.34 – Зображення після обробки нейромережею



Рис. 3.35 – Готова обкладинка для YouTube

ВИСНОВКИ

Нейронні мережі глибинного навчання сьогодні є ключовим інструментом у сфері комп'ютерного зору. Їх активне впровадження у практичні задачі обробки зображень демонструє ефективність новітніх архітектур, зокрема дифузійних моделей, генеративно-змагальних мереж та трансформерів.

У рамках роботи було проведено систематизований огляд основних архітектур: CNN, GAN, Diffusion та Transformer, а також розглянуто методи навчання, оптимізації та техніки, які забезпечують покращення точності та стабільності моделей. Окрему увагу приділено сучасним інструментам генерації та обробки зображень, зокрема Stable Diffusion, Artbreeder, Segment Anything та ClipDrop.

Проведені експерименти дозволили на практиці оцінити можливості кожного з підходів у вирішенні задач класифікації, генерації, сегментації та локального редагування зображень. Для аналізу було використано єдиний вхідний приклад — композитне зображення з людиною, м'ячем і фоном футбольного поля. Це дозволило зіставити результати між архітектурами в однакових умовах.

ПРОДОВЖЕННЯ ВИСНОВКІВ

За підсумками дослідження встановлено, що жодна архітектура не є універсальною. Кожна з них показала високі результати у своїй спеціалізованій задачі. Diffusion-моделі (Stable Diffusion) дозволяють досягати реалістичних і стилізованих зображень, GAN/VAE (Artbreeder) надають гнучкий контроль при генерації облич, а трансформери (SAM + ClipDrop) забезпечують ефективні інструменти для локальної обробки сцен.

У ході практичної частини було доведено, що найбільш ефективний результат досягається при поєднанні різних архітектур. Такий підхід дозволяє компенсувати слабкі сторони кожного методу й реалізовувати складні задачі з високою точністю, стабільністю та творчим контролем. Наприклад, для створення обкладинки або рекламного банера можна поетапно застосовувати інструменти, що базуються на різних моделях: генерація сцени через дифузію, корекція обличчя через GAN, редагування фону через трансформерну сегментацію.

Результати цієї роботи підтверджують доцільність використання гібридного підходу в сучасному дизайні, візуальній комунікації, маркетингу та медіа. Інвестування в розвиток подібних архітектур і технологій є перспективним напрямом, який має значний потенціал для подальших прикладних рішень у сфері штучного інтелекту та обробки зображень.

РЕЗУЛЬТАТИ ВПРОВАДЖЕННЯ

1. Овдієнко С. Д., Бученко І. А. Генерація зображень за допомогою нейронних мереж: сучасні підходи та можливості застосування. Застосування програмного забезпечення в ІКТ : VI Всеукр. науково-техн. конф., м. Київ, 24 квіт. 2025 р. Київ, 2025. С. 348–350. URL: https://duikt.edu.ua/uploads/p_2779_56719466.pdf.

2.Овдієнко С. Д., Бученко І. А. Підходи до побудови нейронних мереж для обробки зображень. Сучасні інтелектуальні інформаційні технології в науці та освіті : V Всеукр. науково-практ. конф., м. Київ, 15 трав. 2025 р. Київ, 2025. Подано до друку