

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: **«Нарощування даних в контексті задачі детекції об'єктів»**

Виконав: студент 4-го року
навчання
освітньої програми «Прикладна
математика»,
спеціальності 113 Прикладна
математика

Решетар Олександр Володимирович

Керівник: Швай Н.О.

кандидат фіз.-мат. наук, ст. викладач

Рецензент: Шаповал Н.В.

к.т.н., старший викладач каф.

штучного інтелекту,

ІПСА НТУУ «КПІ ім. Ігоря

Сікорського»

Кваліфікаційна робота захищена

з оцінкою _____

Секретар ЕК _____

(підпис)

«_____» _____ 20__ р.

Київ - 2023

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

ЗАТВЕРДЖУЮ

Зав.кафедри математики,
проф., доктор фіз.-мат. наук

_____ *Олійник Б.В.*
(підпис)

“ _____ ” _____ 2023

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

для кваліфікаційної роботи
студенту 4-го курсу, факультету інформатики
Решетарю Олександрову Володимировичу

Тема: «Нарощування даних в задачі детекції об'єктів»

Зміст кваліфікаційної роботи:

Анотація

1. Вступ

2. Задача детекції об'єктів

3. Від простих до складних методів аугментації

4. Генеративні змагальні мережі

5. Експерименти

Висновки

Список літератури

Дата видачі “ _____ ” _____ 2023 Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено «_____»_____2023р.

№ з/п	Перелік робіт	Термін виконання етапу	Підпис наукового керівника	Дата ознайомлення наукового керівника	Примітка
1.	Отримання теми кваліфікаційної роботи.	жовтень			
2.	Ознайомлення з темою кваліфікаційної роботи.	жовтень-листопад			
3.	Розробка плану та структури роботи.	листопад-грудень			
4.	Робота з науковою літературою, опис основних означень.	січень			
5.	Дослідження результатів отриманих в літературі.	лютий			
6.	Робота над текстовим оформленням результатів.	березень-квітень			
7.	Попередній аналіз кваліфікаційної роботи. Виправлення помилок.	травень			
8.	Попередній захист кваліфікаційної роботи.	травень			
9.	Захист кваліфікаційної роботи.	червень			

Науковий керівник _____
(ПІБ)

Виконавець кваліфікаційної роботи _____
(ПІБ)

Зміст

Анотація	3
1 Вступ	4
1.1 Актуальність.....	4
1.2 Мета та завдання дослідження	5
2 Задача детекції об'єктів	7
2.1 Постановка задачі	7
2.2 Проблеми з даними.....	8
3 Від простих до складних методів аугментації	12
3.1 Проста та сегментована вставка.....	12
3.2 Gaussian blur.....	13
3.3 Laplacian Pyramid	14
3.4 Poisson blending.....	18
4 Генеративні змагальні мережі	23
5 Експерименти	26
Висновки	28
Список літератури	29
Інтернет ресурси	31
Додаток А	32
Додаток Б	34
Додаток В	35

Анотація

Метою цієї кваліфікаційної роботи є дослідження та вдосконалення методів аугментації для нарощення візуальних даних.

У роботі було розглянуто прості та складні методи аугментації даних та отримано результати їхнього застосування. Була проведена загальна та порівняльна характеристика методів нарощення даних, а також був отриманий час необхідний для генерації тренувальної вибірки. Окрім цього, було запропоновано алгоритм вставки об'єктів у цільове зображення. Нарешті, були висунуті ідеї щодо подальшого дослідження та покращення аугментації даних.

1 Вступ

1.1 Актуальність

Продуктивність та точність нейронних мереж в багатьох випадках залежать від вибору датасету, а саме його розміру та наявності унікальних семплів. Однак, не для всіх задач можна отримати доступ до хороших даних, особливо тому, що їхня розмітка може дорого коштувати. Цю проблему можна обійти, використовуючи аугментацію даних - набір різних технік для збільшення кількості даних за рахунок існуючих. Це включає як компонування існуючих [1],[2],[3] так і генерацію нових семплів [4],[5],[6]. Ці методи є доволі ефективним засобом проти проблем, з якими стикаються більшість нейронних мереж, такі як перевивчення, одноманітність даних. Велика кількість даних сприяє покращенню моделі, вона стає стійкою до нетривіальних експериментів ну й найголовніше - показує точні й кращі результати. Для збирання та розмітки даних може знадобитись багато зусиль та часу: одним з прикладом є проблема у сфері медицини, а саме класифікація захворювань, наприклад ракових клітин в мозку, де обробка кожного зображення вимагає високої кваліфікації спеціаліста. Аугментація даних допомагає дослідникам значно зекономити ресурси, створюючи нові дані для тренування моделей.

Аугментацію даних вирізняють відповідно до поставленої задачі: для класифікації та сегментації зображень - найпростішим видом аугментації є так звані афінні перетворення. До них входять:

- змінення масштабу
- випадкові повороти
- горизонтальні/вертикальні гортання

Нагадаємо, афінні перетворення - це функції, які відображають простір сам в себе, при чому паралельні прямі переходять в паралельні, пересічні - в пересічні, мимобіжні - в мимобіжні. Формально, їх можна записати як:

$$f(x) = M \cdot x + v \quad (1)$$

де M - невироджена матриця, $v \in R^N$. Тобто, для здобуття потрібного ефекту необхідно і достатньо перемножити вхідні матриці (в нашому випадку це зображення) на відповідні перетворення. Наприклад, для перенесення зображення у певній вісі координат, можна застосувати дану матрицю:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Окрім них, вирізняють ще такі методи аугментації:

- вирізання
- змінення кольорової гамми
- додавання шуму

Більш складні методи, такі як GAN[5], передача нейронного стилю (Neural Style Transfer) [6] та інші змагальні нейронні мережі [7] разом з простими методами, дозволяють ще більше урізноманітнити навчальну вибірку.

У цій роботі було досліджено різноманітні техніки аугментації зображень, використовуючи такі алгоритми для комбінування зображень, як Poisson image blending, Gaussian blurring, CP-GAN [8,9]. Використання даних методів дозволило за короткий час збалансувати датасет PASCAL VOC, а

саме нагенерувати зображення двох класів найменш присутніх у тренувальній вибірці.

1.2 Мета та завдання дослідження

Мета кваліфікаційної роботи - дослідження комбінування зображень за допомогою простих та просунутих алгоритмів, що дозволяє збільшити розмір вхідних даних, таким чином допомагаючи моделі розпізнавати об'єкти в різних середовищах, уникнути перевивчення, а також збалансувати датасет.

В даному дослідженні були проведені такі експерименти для генерації зображень:

- Застосування методу copy-paste.
- Застосування методу copy-paste з використанням Gaussian blur, для того щоб не було чіткого контуру між вставленим об'єктом та цільовим зображенням.
- Застосування методу Poisson image blending, для розмиття контурів між об'єктом та зображенням.
- Застосування copy-paste, використовуючи сегментаційні маски.
- Генерація нових зображень, використовуючи GAN з технікою міхур.
- Генерація нових зображень, використовуючи GAN та Poisson image blending.

Набір даних Pascal VOC використовувався як вхідні дані завдяки зручному формату. На відміну від COCO, VOC зберігає інформацію про фотографії в окремих файлах. Моделі порівнювали за допомогою стандартної

оцінки середньої точності (mAP), зокрема mAP@0,5 та mAP@0,95. Бібліотеки `pymru cv2`, `Pillow` використовувалися для керування форматом і розміром зображень. Інші параметри моделі були записані в окремому конфігураційному файлі, знайденому в репозиторії. Деталі конфігурації набору даних описано у файлі `VOC.yaml`.

2 Задача детекції об'єктів

2.1 Постановка задачі

Детекція об'єктів є важливим завданням в області комп'ютерного зору. Дані моделі об'днують в собі задачу класифікації та локалізації. Іншими словами, маючи на вхід зображення модель не тільки визначає координати обмежувальних боксів для об'єктів, а й визначає належність до певного класу.

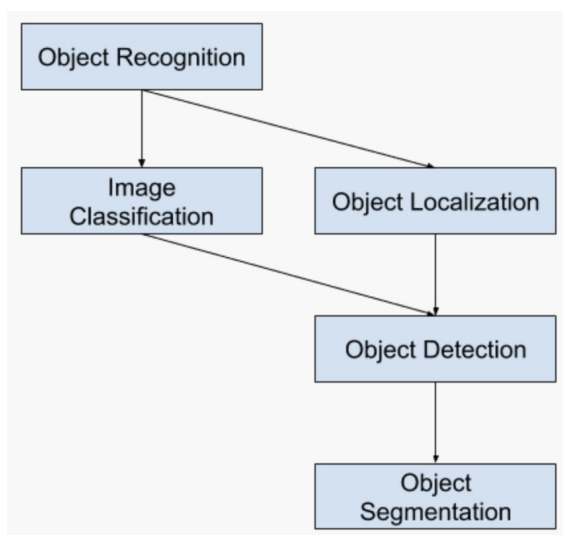


Рис. 1: Ієрархія задач комп'ютерного зору

Одні з перших моделей для розпізнавання об'єктів на фото були CNN, які були винайдені близько 1980 та використовувались переважно для упізнавання рукописних чисел. Якщо у більшості нейронних моделях застосовується множення матриць, то в CNN діє так звана операція конволюції, яка визначена в неперервному випадку як

$$f * g = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2)$$

Вона показує, наскільки форма однієї функції була змінена за рахунок іншої. Оскільки розглядається дискретний випадок, формула

перетворюється на:

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f(m)g(n - m) \quad (3)$$

Операція конволюції здійснюється за рахунок фільтрів, або кернелів, які проходять через зображення та вилучають найважливіші особливості. Конволюційні мережі складаються з декількох шарів, в кожному з яких присутні власні фільтри - таким чином, модель здатна впізнавати складні паттерни на зображеннях. Проте варто зазначити, що в ті часи можливості CNN були обмежені, оскільки глибинне навчання вимагає велику кількість даних для тренування моделей і відповідно, значні обчислювальні потужності. Сьогодні ситуація значно інша. Хоч початок CNN припадає на минуле століття, навіть сьогодні на її основі були побудовані більш швидкі та точні моделі. Модель Yolo(You Only Look Once) набрала швидкої популярності через неймовірну швидкість детекції об'єктів. Її суперник Alexnet показує дещо кращі показники mAP(mean average precision), проте значно програє у швидкості. На відміну від традиційних конволюційних мереж, Yolo не використовує відразу техніку проходження фільтру по зображенню, а розбиває його на клітини, тобто дискретизує, і вже до кожної клітини застосовує операцію конволюції.

Завдяки значному прогресу у сфері машинного навчання, моделі для розрізнення об'єктів стали невід'ємними складовими повсякденного життя. Застосування цих моделей має різноманітні приклади, починаючи від автопілотів у автомобілях для завчасної детекції перешкод, закінчуючи їхнім застосуванням у галузі медицини. Наприклад, система вирізняння обличчя використовується у соціальних мережах для зручного редагування фото, а розробки компанії Apple використовують модель штучного інтелекту для розпізнавання тексту/об'єктів на фото, зроблені мобільним пристроєм. Ці моделі також мають застосування у військовій справі, для своєчасного детекції ворожої техніки та прийняття відповідних рішень [10]. Ідеальним варіантом у даній сфері є повністю автономна система на базі штучного інтелекту, яка здатна

прорахувати можливі сценарії та завдати противнику нищівного удару [11].

Застосування моделей машинного навчання для розпізнавання об'єктів відкриває безмежні можливості у різних галузях. Використання цих моделей дозволяє підвищити ефективність та точність різних процесів, що дозволяє вирішувати багато сучасних проблем.

2.2 Проблеми з даними

Для тренування нейронної мережі необхідна достатня кількість даних: занадто мала кількість може призвести до перевивчення моделі на тренувальній вибірці, з іншого боку, дуже багато - спричинить дуже довге та нефективне навчання, оскільки модель братиме до уваги велику кількість особливостей. Зазвичай, більшість дослідників стикаються саме з нестачою даних. Збір даних та їхня розмітка може вартувати досить дорого особливо у тих випадках, коли вона вимагає кваліфікації відповідного спеціаліста. До прикладу, у галузі медицини, лейблінг зображень займає доволі багато ресурсів через необхідність кваліфікації особи, яка розмічає.

Окрім цього, навіть з достатньою кількістю даних в певному датасеті, сама їхня якість може бути суттєво спотворена. Тому логічним чином, після процесу очищення від неякісних семплів, варто застосувати методи аугментації даних, тобто техніку штучного збільшення навчального набору шляхом створення модифікованих копій набору даних з використанням існуючих даних. У випадку обробки зображень, існує широкий спектр методів аугментації, які можуть бути використані для покращення якості даних та розширення дата-сету. Ці методи можуть бути розділені на дві категорії: прості та складні. До простих методів аугментації належать лінійні трансформації, такі як обертання, стиснення та розтягування. Ці трансформації змінюють геометричні характеристики зображень, дозволяючи змінювати їх положення, масштаб та форму. Навіть прості

методи показали значний приріст [4] у оцінці моделі. У складніших методах аугментації використовуються більш високорівневі техніки, такі як комбінування зображень [2] та накладання одного зображення на інше [3]. Так, звичайний copy-paste [1] здатен генерувати доволі велику вибірку даних, якщо на початковому зображенні присутньо достатня кількість об'єктів.

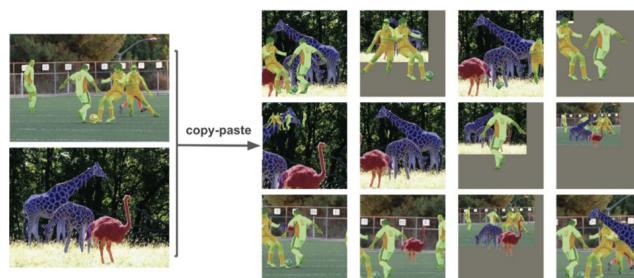


Рис. 2: Copy-paste algorithm

Метод мозаїки включає комбінування частин зображень з різних джерел для створення нових зображень з унікальними комбінаціями об'єктів. Він був вперше представлений у моделі YoloV4. На відміну від свого попередника CutMix, який міксував лише 2 зображення, даний підхід дозволив моделі розпізнавати об'єкти поза контекстом. Інший метод, такий як Mixup, дозволяє регуляризувати мережу і запобігти перенавчанню на тренувальних семплах. Використовуючи доволі простий підхід:

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda)x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j\end{aligned}\tag{4}$$

де x та y репрезентують відібрані семпли та one-hot закодовані лейбли, міхур легко імплементується в коді та дозволяє збільшити вибірку даних, не використовуючи багато обчислювальних можливостей.



Рис. 3: Mosaic algorithm

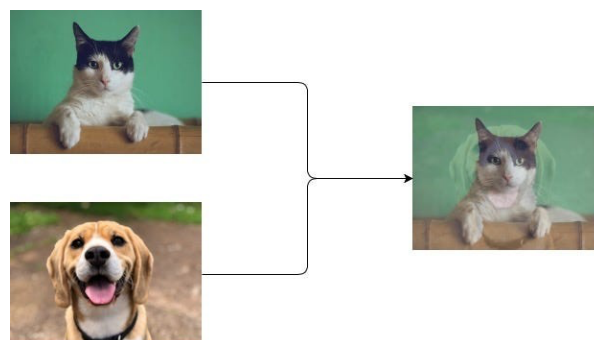


Рис. 4: Mixup algorithm

Крім вищезгаданих методів, генеративні змагальні мережі (GAN) також відіграють одну з важливих ролей у більш просунутих методах розширення даних. GAN мають здатність генерувати нові зображення на основі існуючих, навчаючись копіювати їх розподіл. Це дозволяє створювати реалістичні штучні зображення, які можна використовувати, щоб розширити набір даних. До недавнього часу розширення даних в основному використовувалося для навчання звичайних глибоких мереж, тоді як у [13] дослідники намагалися застосувати методи розширення для навчання самих GAN. Вони виявили цікаве явище: якщо певні операції, такі як виріз або колір тремтіння застосовуються до реальних зображень, дискримінатор може помилково сприйняти їх як справжні, що призведе до створення нових зразків із спотвореними зображеннями в мережі. Як наслідок, більшість моделей GAN використовують випадкове кадрування та/або перевертання для збільшення. Крім того, аугментація лише реальних фотографій не тільки не покращує точність навчених GAN, але й може погіршити її. Тому автори дійшли висновку, що корисно застосовувати методи доповнення як до реальних, так і до створених зображень, гарантуючи, що дискримінатор не зможе відрізнити реальні зображення від підроблених виключно на основі методів аугментації.

Загалом, комбінація простих і складніших методів аугментації дозволяє покращити розпізнавання об'єктів на зображеннях, збільшити різноманітність даних та знизити ризик перенавчання моделі на обмеженому датасеті.

3 Від простих до складних методів аугментації

3.1 Проста та сегментована вставка

До цього моменту було згадано прості методи аугментації зображень, які можуть значно збільшити розмір датасету та запобігти перевивченню моделі. Наприклад, *copy-paste* - використовує доволі простий алгоритм аугментації: проста вставка об'єктів з одного зображення в інше. Такий підхід може утворювати чисельну кількість нових тренувальних даних, оскільки вибір об'єктів так само як і вибір місця вставки є великим. На відміну від міксапа та мозаїки, даний підхід копіює не увесь обмежувальний бокс, а лише пікселі конкретного об'єкта. Деякі дослідники намагались визначити хороше місце для вставки, проте як показали в [1], випадкова вставка демонструє чудові результати. Метод, який був використаний для цього, працює наступним чином: вибирається два випадкових зображення з датасету, на які накладається випадкова зміна масштабу та горизонтальне відображення. Перше є технікою, яка використовується для випадкового зміщення масштабу об'єктів або зображень з метою аугментації даних або отримання різноманітності в навчальних наборах. Це може включати випадкове збільшення або зменшення розміру об'єктів, їх розтягування або стискання, залежно від встановлених параметрів або випадкового вибору. Друге ж використовує для створення варіацій зображень горизонтальні відображення зображень вздовж вертикальної осі. У результаті об'єкти або елементи зображення, які зазвичай знаходяться зліва, з'являються справа, а ті, що зазвичай знаходяться справа, з'являються зліва. Ця техніка може бути корисною для нейронних мереж, що вивчають зображення, оскільки вона дозволяє розширити навчальний набір та покращити загальну здатність моделі до упізнавання об'єктів у різних геометричних конфігураціях. Однак даний підхід має певний недолік: об'єкт, вставлений в цільове зображення, має видимий контур, що робить зображення нереалістичним. Як виникає ця

проблема? У випадку відсутності макси, її роль займає обмежувальний бокс. Якщо маска нашого об'єкту вже задана, можна використати даний підхід

$$I(x, y) = M(x, y)S(x, y) + (1 - M(x, y))T(x, y) = \begin{cases} S(x, y), & M(x, y) = 1 \\ T(x, y), & M(x, y) = 0 \end{cases} \quad (5)$$

де $I(x, y)$ - скомпоноване зображення, $S(x, y)$ - об'єкт який потрібно вставити, $M(x, y)$ - маска об'єкту, $T(x, y)$ - цільове зображення. Приклад даного підходу продемонстровано на рис. 5



Рис. 5: Сегментована вставка

Згідно з нашими спостереженнями, контур вставленого об'єкту досить чітко виділяється на цільовому зображенні. Але є спосіб виправити цю ситуацію: можна застосувати функцію лінійного змішування, яка створює плавний перехід між вставленим об'єктом та цільовим зображенням на його контурі.

3.2 Gaussian blur

Гауссовий блюр є ще однією технікою аугментації, яку використовують в першу чергу перед переходом до більш складних. Як вже було зазначено, деяка вибірка з датасету може бути сильно деградована в якості і такі випадки доволі часто трапляються в житті:

- Камера нерухома, але об'єкти, які вона виявляє, часто рухаються; напр. камера спостерігає за дорожнім рухом
- Камера рухається, але об'єкти, які вона виявляє, нерухомі; напр. користувач має мобільний додаток для сканування обкладинки книги
- Камера та об'єкти, які вона виявляє, рухаються; напр. дрон летить над вітряним полем посівів

Як не дивно, розмиття зображення використовує спеціальний фільтр, який при проході через зображення конволює відповідні регіони. Гауссовий блюр можна застосувати до вищезгаданого методу. Шляхом використання масок та градієнтом переходу можна зменшити різкий перехід між вставленим об'єктом та цільовим зображенням. Раніше було розглянуто функцію, що створює нове зображення. Щоб її покращити, можна використовувати певну функцію f на масці, як описано далі:

$$I(x, y) = f(M)S(x, y) + (1 - f(M))T(x, y) \quad (6)$$

Отримавши градієнт маски, з'являється можливість створити плавний перехід між кольоровими гаммами вставки та цільового зображення. Гаусівський блюр іноді використовують для дата-аугментації на одному зображенні, додаючи певний шум на фото. У цьому дослідженні було обрано комбінування зображень за допомогою цієї техніки, що можна інтерпретувати як покращений *copy-paste*. Нижче наведено приклад такої вставки.



Рис. 6: Гауссовий блюр

Хоча автори в [1] вказали на незмінність результату після використання техніки злиття, в нашому ж випадку це дало незначний приріст (див. розділ "Експерименти").

3.3 Laplacian Pyramid

Метод Laplacian Pyramid можна використовувати для подальшого змиття контурів та злиття кольорів двох об'єктів. Нижче наведено приклад зображення, яке було покращено за допомогою цього методу.

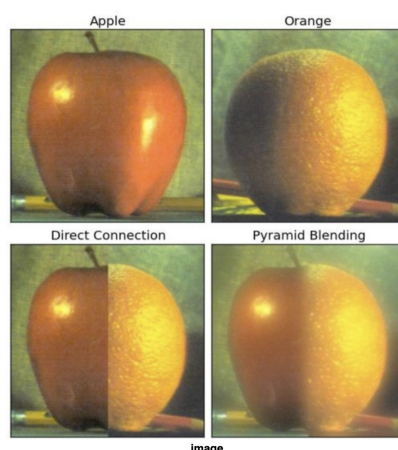


Рис. 7: Використання піраміди Лапласа

Метод Laplacian Pyramid забезпечує ефективне розмиття контурів та злиття кольорів двох об'єктів шляхом використання пірамід Гауса. Починаючи з Гаусових пірамід, побудовані Лапласів піраміди забезпечують послідовне зменшення розміру зображень шляхом застосування операцій down-sampling та згладжування. Цей процес дозволяє зберігати інформацію при зменшенні розмірів зображення. Згладжування використовує операцію конволюції для зменшення різких контрастів та кутів. Чим вища частота, тим більше зменшене зображення наближається до оригіналу, але одночасно менше інформації зберігається.

Метод Laplacian Pyramid досягає балансу між частотою та якістю нової інформації шляхом використання рекурсивної операції вирахування різниці між оригінальним зображенням та його розмірами в піраміді. Це сприяє розмиттю контурів та злиттю кольорів об'єктів, що використовуються.

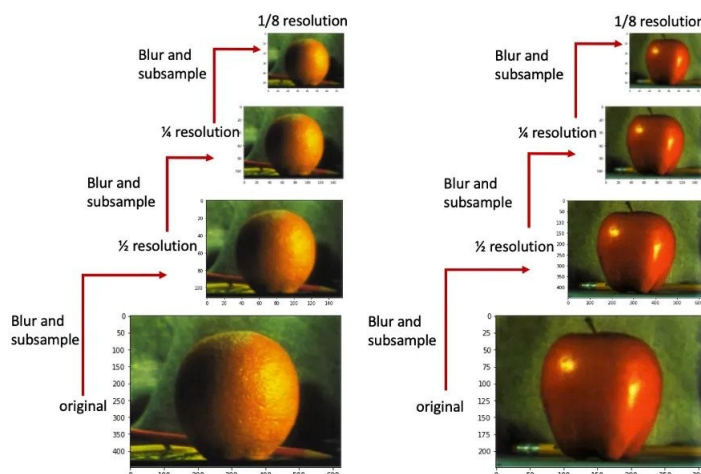


Рис. 8: Використання гаусового фільтру та down-sampling

Нарешті, для побудови піраміди Лапласа потрібно від початкового зображення $\mathcal{G}_c$ відняти наступне зображення, яке було up-sampled методом інтерполяції. Зазвичай для up-sample використовують метод вставки нульових пікселів в кожний ряд/стовпець та потім застосовують гаусовий фільтр, щоб зменшити сильні контрасти.

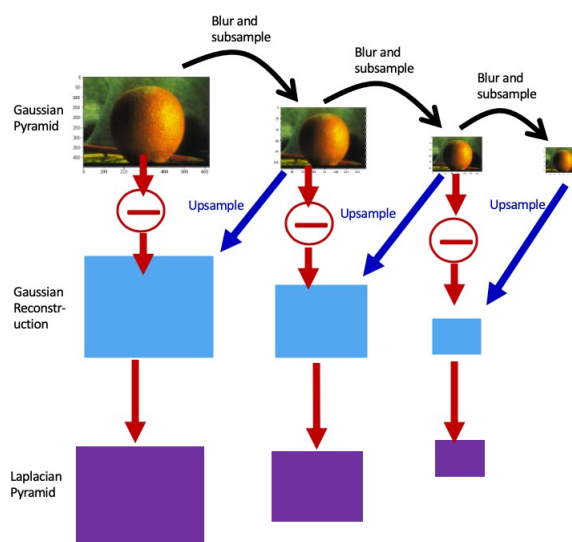


Рис. 9: Піраміди Лапласа

Після upscale розмір наступного зображення буде таким самим як теперішнє, що дозволяє їх віднімати. У результаті одержано нове зображення, яке зберігатиме лише особливі атрибути об'єктів, наприклад кути.

$$\text{Laplacian Image}_k = \text{Original Image}_k - \text{Upsampled Gaussian}_{k-1}$$



Рис. 10: Зображення лапласа

Цей алгоритм застосовується до обох зображень. Було застосовано Гаусове згладжування на масках об'єктів, що дозволяє нам зробити плавний перехід на границях між ними. Також було використано рівняння $\text{Laplacian}_k = \text{Laplacian}_a \times \text{Mask}_k + (1 - \text{Mask}_k) \times \text{Laplacian}_b$, щоб відтворити складене зображення на кожному кроці. В кінці отримано зображення тієї ж самої розмірності, що й початкові. Повний алгоритм зображений на схемі

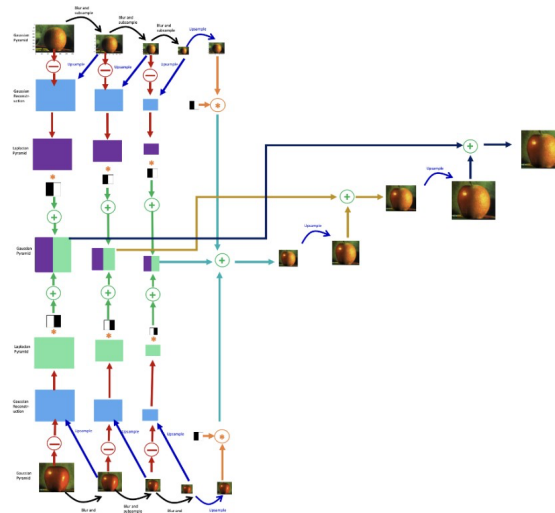


Рис. 11: Згладжування за допомогою пірамід Лапласа

Тобто:

1. Будуємо піраміди Лапласа для двох вхідних зображень
2. Будуємо піраміди Гаусса для масок

3. Формуємо на кожному кроці зображення Лапласа

Незважаючи на ефективність минулого методу, на деяких прикладах результат не такий який б хотілось. Основними обмеженнями є:

1. Неможливість сильного контрасту(наприклад сірий на білому - сірий на чорному)
2. Зображення повинні добре розміщуватись
3. Кольорове тло «проливається»



Рис. 12: Проливання кольорового фону

Даний метод використали в [13] для аугментації зображення в гістопатології для дослідження клітин під мікроскопом. При збільшенні масштабу в 200 та 400 разів, цей підхід показує кращі результати ніж без аугментації. В цьому дослідженні зосереджується на увага на більш широкому спектру зображень, включаючи різні види класів.

3.4 Poisson blending

Хоча попередні методи частково розв'язували проблему видимих контурів вставлених об'єктів, все ж таки прагнеться до більшого вдосконалення. Но-

вий метод повинен максимально злити вставлений об'єкт з цільовим зображенням. В 2003 році був запропонований метод Poisson blending, який ґрунтується на використанні пірамід Лапласа для градієнтів зображень, що покращує колірну гаму вставленого об'єкта. Цей підхід дозволяє змішувати деталі на локальному рівні навколо межі виділення для дрібних деталей, тоді як нижчі частоти змішуються на більшій відстані навколо цих меж. Для роботи з градієнтом зображення цей підхід використовує рівняння часткових похідних Пуассона з граничними умовами Діріхле, яке визначає лапласіан невідомої функції в області вставки. Маючи методи для створення оператора Лапласа у випадку зображень, рівняння Пуассона може бути розв'язано аналітичним чином. Це також можна інтерпретувати іншим чином: шукається функція, градієнт якої, в термінах L_2 норми, близький до деякого векторного поля - в даному випадку поле керування. Існує безліч способів задати поле керування, їхній вибір залежить від об'єкта в початковому зображенні.

Пуассонівське рівняння й раніше застосовувалось в задачі комп'ютерного зору. Так, в [14] дослідники використали його для редагування зображення, використовуючи розріджений набір крайових елементів. Для приглушення об'єкту, відповідні лейбли було видалено; для додавання об'єкту, відповідні края та їхні кольорові гамми були інкорпоровані. Після цього, нове зображення було інтерпольовано за рахунок доданих країв. Попри це, даний підхід є не зовсім точним, оскільки деякі частини зображення втрачаються протягом конвертування в/з контурної області. Перш ніж перейти до самого методу, введемо деякі позначення: нехай $\exists T$ - замкнена підмножина $\mathbb{R}^2$, в нашому випадку цільове зображення, I - замкнена підмножина T з межею ∂I . Введемо позначення скалярної функції, визначеної в цільовому зображенні f^* , при чому $f^* \in T \setminus \text{Int}(I)$. Позначимо за f та v невідому функцію та векторне поле, визначені на I . Найпростішим методом інтерполяції є проблема мінімізації:

$$\begin{aligned} \min_{f(x,y)} \iint_{\Omega} |\nabla f(x,y)|^2 dx dy, \\ \text{s.t. } f|_{\partial I} = f^*|_{\partial I} \text{ on } \partial I \end{aligned} \quad (7)$$

При цьому, дана задача оптимізації повинна задовільняти рівнянню Ойлера-Лагранжа:

$$\begin{aligned}\nabla^2 f &= 0, \forall x \text{ in } \Omega, \\ f|_{\partial I} &= f^*|_{\partial I} \text{ on } \partial\Omega\end{aligned}\tag{8}$$

Даний підхід продукує небажані артефакти, а також може розмивати текстури. Натомість, можна покращити метод шляхом додавання поля керування v :

$$\begin{aligned}\min_{f(x,y)} \iint_{\Omega} |\nabla f(x,y) - v|^2 dx dy, \\ s.t. f|_{\partial I} &= f^*|_{\partial I} \text{ on } \partial\Omega\end{aligned}\tag{9}$$

де розв'язком буде розв'язок задачі Пуассона:

$$\begin{aligned}\Delta f &= \nabla \cdot v, \\ s.t. f|_{\partial I} &= f^*|_{\partial I} \text{ on } \partial\Omega\end{aligned}\tag{10}$$

Це фундамент для застосування рівняння Пуассона для вставки, інші методи додають додаткові обмеження для покращення точності. Оскільки зображення RGB, то відповідно для кожного каналу вирішується своє Пуассонівське рівняння. Одна із прикладних задач, яку вирішує Пуассонівське рівняння - це так зване seamless cloning, вставка без видимих контурів. В даному випадку можна взяти за v градієнтне поле початкового зображення ($v = \nabla g$). Знову ж таки, маючи початкове зображення S та цільове T , формується складене зображення I за допомогою рівняння Пуассона. Задача оптимізації схожа на рівняння теплопровідності, де спостерігається зміна тепла всередині площини при заданих граничних умовах.

$$\begin{aligned}\min_{I(x,y) \in \Omega} \iint_{\Omega} \|\nabla I(x,y) - \nabla S(x,y)\|^2 dx dy, \\ s.t. I(x,y) &= T(x,y) \text{ on } \partial\Omega\end{aligned}\tag{11}$$

Тобто це зводиться до мінімізації різниці градієнтів утвореного зображення та цільового всередині площини вставки, відповідно на границі градієнт повинен співпадати з цільовим зображенням. Роз'язком даної задачі:

$$\begin{aligned} \nabla^2 I(x, y) &= \nabla^2 S(x, y) \in \Omega, \\ s.t. I(x, y) &= T(x, y) \in \partial\Omega \end{aligned} \quad (12)$$

де ∇^2 - оператор Лапласа. Воно вирішується елементарним чином, знаючи, що оператор Лапласа для зображень виглядає так:

0	1	0
1	-4	1
0	1	0

Рис. 13: Додатний оператор Лапласа

Зазвичай фільтр розміром 3x3 або 5x5 будується для виявлення контурів об'єктів у вертикальному та горизонтальному напрямках. Інший фільтр може розпізнавати контури під різним кутом. Алгоритм детекції контурів також відомий як "zero-crossing" оскільки оператор Лапласа, який застосовується, є другою похідною і дозволяє виявляти різку зміну контрасту, тобто границі об'єктів.

Є два випадки розташування пікселів: на межі та всередині площини.

Розглянемо перший випадок, тоді рівняння (9) можна розписати як:

$$\begin{aligned} &-4I(x, y) + I(x + 1, y) + I(x, y + 1) + I(x - 1, y) + I(x, y - 1) \\ &= -4S(x, y) + S(x + 1, y) + S(x, y + 1) + S(x - 1, y) + S(x, y - 1) \end{aligned} \quad (13)$$

Нагадую, в даній рівності, в якості невідомого виступає I . Виходить звичайна лінійна система з 5-ма невідомими. Дану систему можна роз'язати доволі швидко за допомогою комп'ютерних обчислень. Перейдемо до 2 випадку:

припустимо піксель(червоний) знаходиться на краю



Рис. 14: Різні положення пікселя під час проходження фільтру: піксель знаходиться на межі вставки(рисунок зліва); піксель знаходиться всередині(малюнок справа).

2 доданки є відомими, оскільки на краях $I(x,y)=T(x,y)$:

$$\begin{aligned} & -4I(x,y) + I(x+1,y) + I(x,y+1) + T(x-1,y) + T(x,y-1) \\ & = -4S(x,y) + S(x+1,y) + S(x,y+1) + S(x-1,y) + S(x,y-1) \end{aligned} \quad (14)$$

Прирівнюючи обидва рівняння та переносячи відомі в праву сторону, отримаємо звичайну лінійну систему рівнянь:

$$\begin{bmatrix} \dots & -1 & \dots \\ -1 & 4 & -1 \dots \\ 0 & -1 & 0 \dots \\ \dots & \dots & \dots \end{bmatrix} \times \begin{bmatrix} I(\cdot, \cdot) \\ I(\cdot, \cdot) \\ \dots \end{bmatrix} = [S, T] \quad (15)$$

Приклади даного методу зображені на рис.15

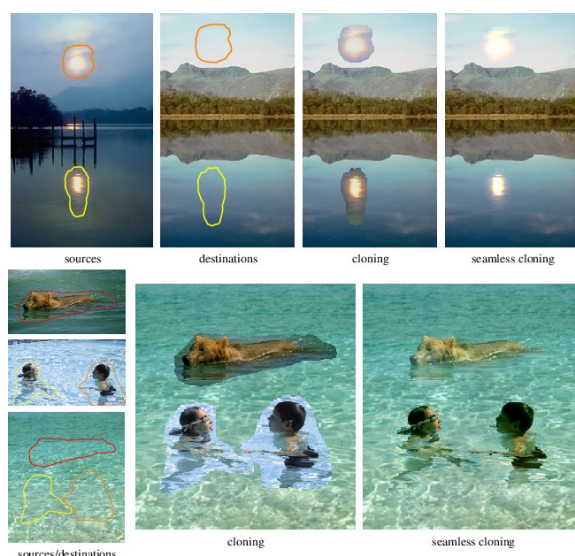


Рис. 15: Приклади пуассонівської вставки

Як бачимо з рисунків, контури вставлених об'єктів(-а) зливаються з цільовим зображенням у місці вставки, при цьому самі(сам) об'єкти втрачають дещо в яскравості. Попри це, новоутворені зображення виходять дуже реалістичні, що позитивно впливає на якість аугментованих даних.

Але далеко не кожна така вставка буде успішною - в залежності від задачі, місце в цільовому зображенні повинно бути вибрано достатньо коректно, щоб градієнти приблизно були однаковими. У випадках, коли об'єкт не займає всю площину вставки, маючи так звані "дірки звичайне Пуассонівське рівняння може занадто сильно усереднити кольорову гамму, що призводить до небажаного результату.



Рис. 16: Вставка об'єкту з дірками

Таке також трапляється у ситуаціях, коли є необхідність змішування текстур цільового зображення та вставленого об'єкту. Тоді в кожній точці вставки залишається сильніша варіація з цільового або вставленого зображення, використовуючи поле керування:

$$\forall x \in \Omega, v(x) = \begin{cases} \nabla f^*(x), & \text{if } |\nabla f^*(x)| > |\nabla g(x)| \\ \nabla g(x), & \text{otherwise.} \end{cases} \quad (16)$$

Рівняння Пуассона має широке застосування у вирівнюванні країв, локальній зміні яскравості та зміні кольору зображень. Однією з основних переваг цього рівняння є його ефективність та здатність до швидких обчислень, що дозволяє створювати високореалістичні зображення. Завдяки рівнянню Пуассона досягаються високі рівні якості і деталізації в обробці зображень, дозволяючи отримувати зображення з виразними краями, різними рівнями яскравості та точно відтвореними кольорами. Це робить рівняння Пуассона потужним інструментом для редагування та покращення зображень у візуальній обробці та комп'ютерній графіці.

4 Генеративні змагальні мережі

Останнім часом генеративні змагальні мережі стали дуже популярними у сфері нейронних мереж. Ці мережі мають досить просту архітектуру, але вражаюче високі результати генерації нового контенту. Відомим прикладом застосування GAN є модель GPT-3/4, розроблена компанією OpenAI, яка може не тільки відповідати на запитання користувача, але й генерувати програмний код або зображення на основі текстового опису (DALL-E).

Також, генеративні змагальні мережі знайшли застосування в сфері дата аугментації. Перед тим, як розглянути застосування GAN, корисно згадати їхню архітектуру.

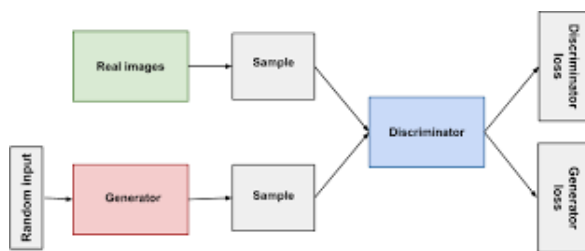


Рис. 17: GAN

Головною ідеєю GAN-ів є використання двох мереж - генератора та дискримінатора, які співпрацюють, щоб створити нові дані, які можуть бути використані для навчання моделі. Генератор створює нові дані на основі вхідного набору даних, тоді як дискримінатор оцінює, наскільки добре нові дані відповідають реальним даним. Генератор намагається обманути дискримінатор, створюючи більш реалістичні дані, тоді як дискримінатор намагається розрізнити між реальними та штучно згенерованими даними.

У цьому дослідженні [6] було використано генеративну змагальну мережу (GAN) для створення синтетичних числових даних. Автори провели навчання моделі виключно на цих синтетичних даних. На незбалансованому наборі даних GAN показав кращі результати, ніж вихідний набір даних, але не досяг таких же результатів, як підходи SMOTE або ADASYN для надмірної вибір-

ки меншості даних. На збалансованому наборі даних GAN показав ще кращі результати, ніж на оригінальні. Дослідники в роботі [15] використали GAN для отримання нових даних про пухлини головного мозку і змогли збільшити показник mAP. Їх GAN може генерувати МР-зображення розміром 256x256 з метастазами в мозок випадкової форми в обмежувальних прямокутниках у потрібному положенні та розмірі. Умовні GANи є покращеним варіантом, оскільки вони можуть генерувати нові дані з урахуванням певних умов, таких як генерація соняшників замість випадкових квітів. Розширення GAN може призвести до помірного, але значного покращення ефективності сегментації в багатьох випадках, залежно від обсягу реальних даних, доступних для навчання. Найбільші покращення зазвичай спостерігаються, коли реальні дані обмежені, але може відбутися зниження покращення на дуже низькому рівні доступних даних через неадекватне навчання GAN. Дослідження показують, що синтетичні дані не приводять до гіршої продуктивності у випадку даних СТ, навіть якщо доступні великі обсяги реальних даних. Проте, у випадку МР спостерігається зменшення коефіцієнта подібності Dice (DSC) після певної переломної точки, яка пов'язана з кількістю доступних даних. Після цієї точки, збільшення GAN може навпаки завдати шкоди замість допомогти.

Перевагою традиційного методу доповнення даних порівняно з GAN є можливість екстраполяції. Хоча GAN можуть ефективно заповнювати прогалини в розподілі дискретних навчальних даних і збільшувати джерела дисперсії, які складно збільшити за допомогою інших методів, вони не можуть розширити розподіл за межі навчальних даних. В експерименті, описаному в [16], було побудовано мережу для генерації масок об'єктів на фотографіях. Було використано цей підхід для дата аугментації. Ідея полягає в тому, що генератор створює маску та копіює різні частинки зображення в цільове, а дискримінатор визначає, чи є складене зображення реальним. Якщо генератор створює маску не для об'єкту або погані шматки фону, він буде оштрафований дискримінатором. Нове зображення утворюється відомим чином (1). Дослідники в [9] зіткнулись з проблемою, що змішувальний GAN(blending GAN) генерує чіткі маски для об'єктів лише для малих зображень, в той час як для великих утворюються багато шуму. Тому, вони запропонували рі-

шення застосувати пуассонівське рівняння до алгоритму GANa, так званий GP-GAN, для згладжування сильного контрасту між зображеннями. Однак, в датасеті PASCAL VOC не для кожного зображення присутня маска відповідного об'єкта, тому даний варіант був відкинутий, оскільки створювати маски для всіх зображень доволі довгозатратна задача. Натомість, було прийнято рішення застосувати підхід [16], в якому нейронна мережа вчиться генерувати маски для об'єктів реалістично, тобто складене зображення було схоже на реальне (це вирішує дискримінатор). Однак, було виявлено проблему з розмірами зображень: автори репозиторію тренували модель на датасеті CIFAR-10, де розміри фотографій складають 64x64, тоді як розміри VOC зображень більші. Тренування на зображеннях, які присутні в датасеті, не вдалося провести через обмежену кількість пам'яті графічного процесора для виконання обчислень. Тому довелося зменшувати вхідні фото до розміру 64x64, що привело до погіршення якості деяких фото. Приклади згенерованих зображень GANами зображено на рис. 18



Рис. 18: CP-GAN

Можливості GAN (генеративно-змагальних мереж) в аугментації даних вражають, оскільки вони ефективно вивчають внутрішній розподіл вхідних даних і генерують дуже реалістичні вибірки. Однак, є кілька обмежень:

1. Оцінка: Відсутня внутрішня метрика для оцінки якості згенерованих вибірок. Хоча були запропоновані експериментальні метрики, потрібні подальші дослідження в цьому напрямку.

2. Нестабільність тренування: Тренування GAN зазвичай є нестабільним і вимагає значних обчислювальних ресурсів. Часто потрібно знайти деякий досконалий баланс між мережами генератора та дискримінатора, що може бути викликом.

Незважаючи на ці обмеження, GAN залишаються перспективним підходом для аугментації даних, пропонуючи потенціал для покращення різноманітності та реалістичності згенерованих вибірок. Подальші дослідження та розвиток технік тренування GAN очікуються з метою подолання цих викликів та розкриття повного потенціалу GAN у різних застосуваннях.

5 Експерименти

У загальному, було проведено п'ять експериментів з різною складністю. Пропонується додаткові варіанти для подальшої роботи, які будуть описані в кінці. Оскільки проблема стосується аугментації даних, було обрано два класи з найменшою кількістю фото (bus та diningtable) та збільшили їх кількість близько на 400 фото для кожного класу. Час, потрібний на генерацію зображень наведений в табл.2(Див. додаток В). Вставка об'єктів у цільове зображення відбувалась випадковим чином (Алгоритм наведений у додатку Б). Цей підхід був застосований для всіх експериментів. Середовище, яке було застосоване для аугментації даних є Colab від компанії Google. Тренування відбулось на платформі Kaggle використовуючи графічний процесор P100. Результати дослідів наведені в табл.1

Метод	Клас	Епох	map.5	map.95
Baseline	-	50	0.693	0.442
Naive Copy-paste	dining table, bus	50	0.695	0.45
Segmented copy-paste paste	dining table, bus	50	0.699	0.465
Gaussian blur	dining table, bus	50	0.711	0.582
Poisson image editing	dining table, bus	50	0.702	0.441
Copy-GAN	dining table, bus	50	0.706	0.449
Copy-GAN + Gaussian blur	dining table, bus	50	0.698	0.444

Табл. 1: Результати експериментів

Після аналізу таблиці можна зробити висновок, що метод, який використовує згладження Гаусса, демонструє найбільшу ефективність. Метод вставки Пуассона і GAN показали приблизно схожі результати, хоча перший є більш високорівневим методом порівняно з останнім. Це може бути зумовлено наявністю проблемних регіонів у цільовому зображенні, що сприяє появі артефактів у вставленому зображенні. Також варто відзначити, що сегментована вставка показала кращі результати порівняно з простою вставкою, саме через кращі маски для об'єктів на зображенні.

Для поліпшення результатів можна удосконалити алгоритм точного визначення місця вставки, враховуючи схожість градієнтів вирізаного зображення та регіону цікавості. Це допоможе уникнути створення нереалістичних зображень, які можуть негативно вплинути на якість моделей. Також пропонується розглянути застосування цих методів на всіх класах датасету.

Процес навчання моделей можна ознайомитись у додатку А.

Висновки

У цьому дослідженні були розглянуті різні методи збільшення об'єму даних для задачі детекції об'єктів. Аугментація зображень має велике значення в навчанні нейронних мереж, оскільки вона дозволяє ефективно збільшити кількість доступних тренувальних даних. У вступі були викладені прості методи аугментації, зокрема афінні трансформації. В наступних розділах акцент був зроблений на складніших підходах, таких як Пуассонівська вставка, Піраміди Лапласа та інші. Крім того, був застосований підхід з використанням генеративних змагальних мереж - генерація масок для об'єктів на початковому зображенні.

За результатами експериментів можна зробити висновок, що використання генеративних змагальних мереж є найбільш ефективним серед інших методів. Методи, які вставляють об'єкти у цільове зображення, можуть бути поліпшені шляхом використання кращого алгоритму вставки, який враховує схожість градієнтів вирізаних об'єктів та регіонів у цільовому зображенні.

Результати цього дослідження мають велике значення для вибору оптимальної стратегії збільшення розміру датасету. Методи, які були розглянуті в цій роботі, дозволяють уникнути проблем, які виникають при застосуванні наївних методів аугментації зображень, і забезпечують кращі контури об'єктів. Продовження дослідження в напрямку вдосконалення цих методів може значно підвищити точність детекції об'єктів та покращити результати моделей машинного навчання, що використовуються в цій області.

Список літератури

1. *Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, et al.* Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. arXiv:2012.07177, 2020
2. *Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao.* YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004.10934, 2020
3. *Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, David Lopez-Paz.* mixup: Beyond Empirical Risk Minimization, arXiv:1710.09412. 2018
4. *Suorong Yang, Weikang Xiao, Mengcheng Zhang, et al.* Image Data Augmentation for Deep Learning: A Survey. arXiv:2204.08610, 2022
5. *Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, et. al.* Generative Adversarial Networks, arXiv:1406.2661, 2014
6. *Fabio Henrique Kiyotaki dos Santos Tanaka, Claus Aranha.* Data Augmentation Using GANs, arXiv:1904.09135, 2019.
7. *Leon A. Gatys, Alexander S. Ecker, Matthias Bethge* A Neural Algorithm of Artistic Style, arXiv:1508.06576, 2015
8. *Lingzhi Zhang, Tarmily Wen, Jianbo Shi.* Deep Image Blending, arXiv:1910.11495, 2019.
9. *Huikai Wu, Shuai Zheng, Junge Zhang, Kaiqi Huang.* GP-GAN: Towards Realistic High-Resolution Image Blending, arXiv:1703.07195, 2019.
10. Military object detection in defense using multi-level capsule networks [Електронний ресурс] // Soft Computing. – 2023. – Режим доступу до ресурсу: <https://link.springer.com/article/10.1007/s00500-021-05912-0>.
11. Artificial Intelligence and Machine Learning for Autonomous Military Vehicles [Електронний ресурс] // Engineer Research and Development Center. – 2020. – Режим доступу до ресурсу: <https://apps.dtic.mil/sti/pdfs/AD1106872.pdf>.

12. *Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, Han Zhang.* Image Augmentations for GAN Training, arXiv:2006.02595, 2020
13. *Steve Tsham Mpinda Ataky, Jonathan de Matos, et. al.* Data Augmentation for Histopathological Images Based on Gaussian-Laplacian Pyramid Blending, arXiv: 2002.00072, 2020
14. *James H. Elder, Richard M. Goldberg.* Image Editing in the Contour Domain. IEEE Trans. Pattern Anal. Mach. Intell. 23(3): 291-296 (2001)
15. *Changhee Han, Kohei Murao, Shin'ichi Satoh, Hideki Nakayama.* Learning More with Less: GAN-based Medical Image Augmentation, arXiv:1904.00838, 2019
16. *Relja Arandjelovic, Andrew Zisserman.* Object Discovery with a Copy-Pasting GAN, arXiv:1905.11369, 2019
17. *Debidatta Dwibedi, Ishan Misra, Martial Hebert.* Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection, arXiv:1708.01642, 2017
18. *Wei Jiang, Na Ying.* Improve Object Detection by Data Enhancement based on Generative Adversarial Nets, arXiv:1903.01716, 2019
19. *Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, Olga Russakovsky.* A Study of Face Obfuscation in ImageNet, arXiv:2103.06191, 2022
20. *Pedro Tabacof, Eduardo Valle* Exploring the Space of Adversarial Images, arXiv:1510.05328, 2016

Интернет ресурсы

1. Image blending using laplacian pyramids
2. Image blending techniques
3. Affine transformation
4. Poisson blending
5. Poisson image editing
6. Fast-Poisson Image editing
7. Adversarial attacks
8. Object detection
9. Deep learning in medical imaging
10. Convolution

Додаток А

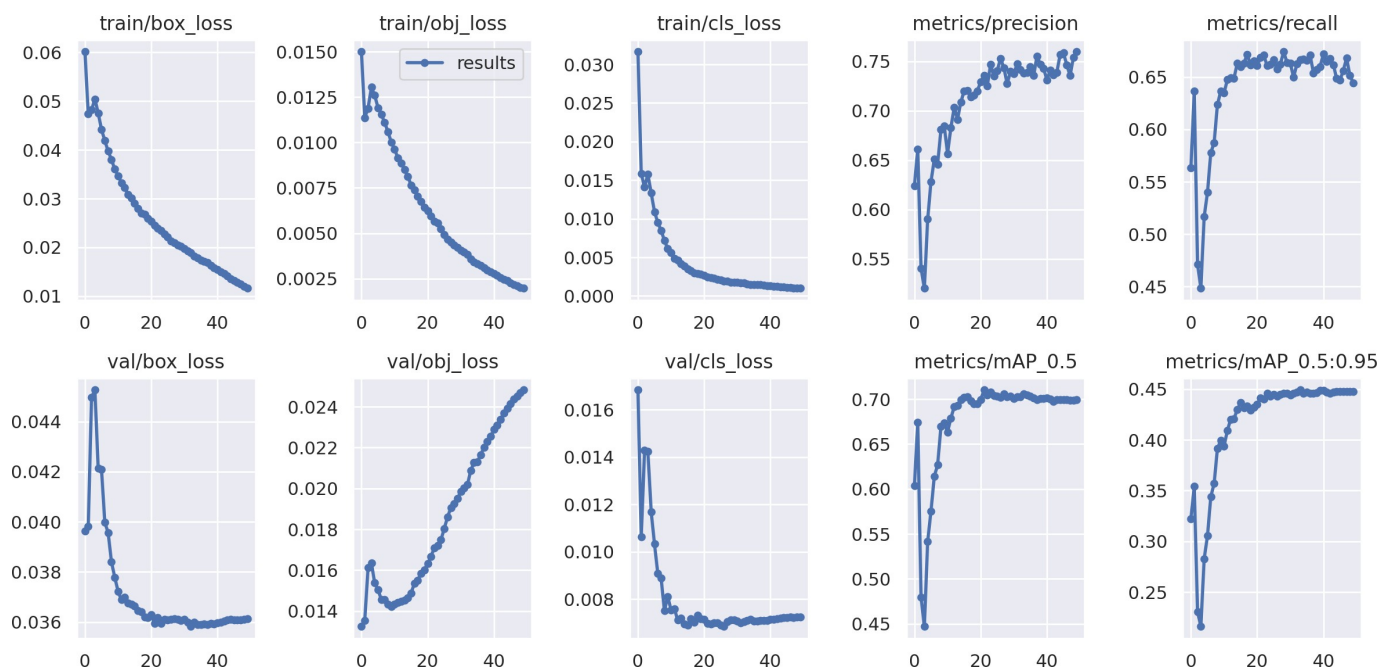


Рис. 19: CP-GAN

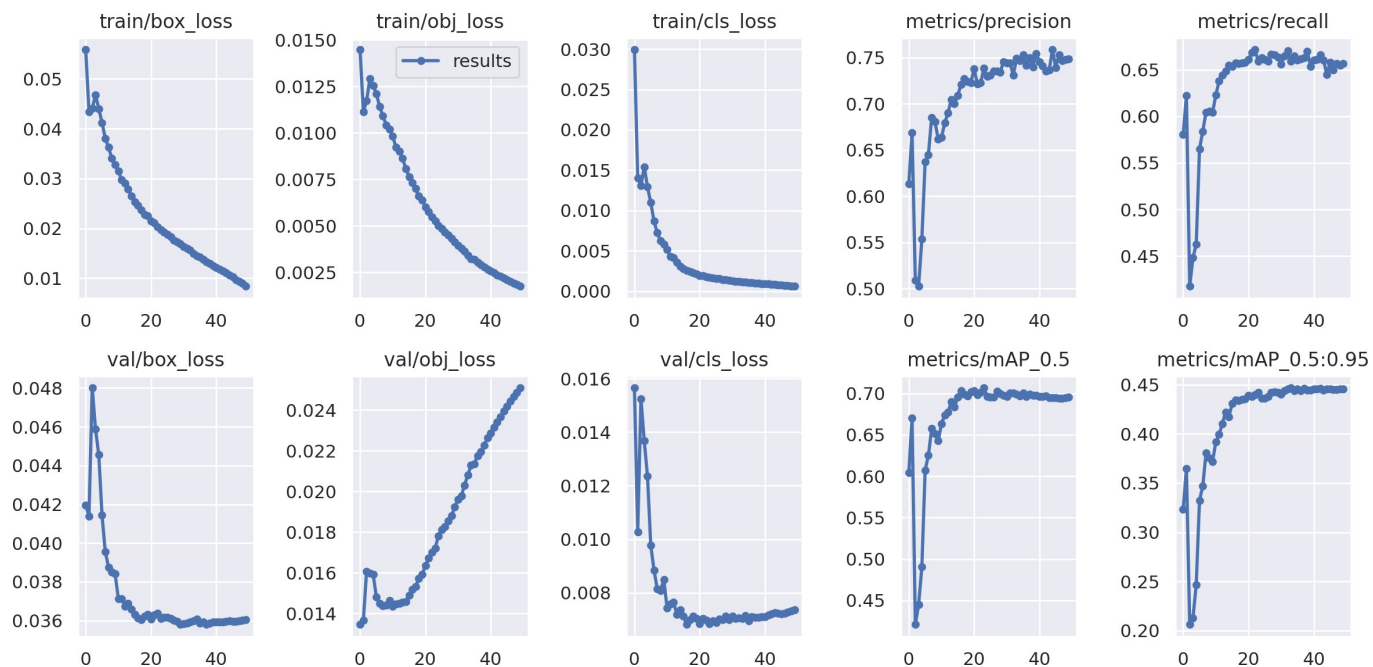


Рис. 20: Segmented Copy-Paste

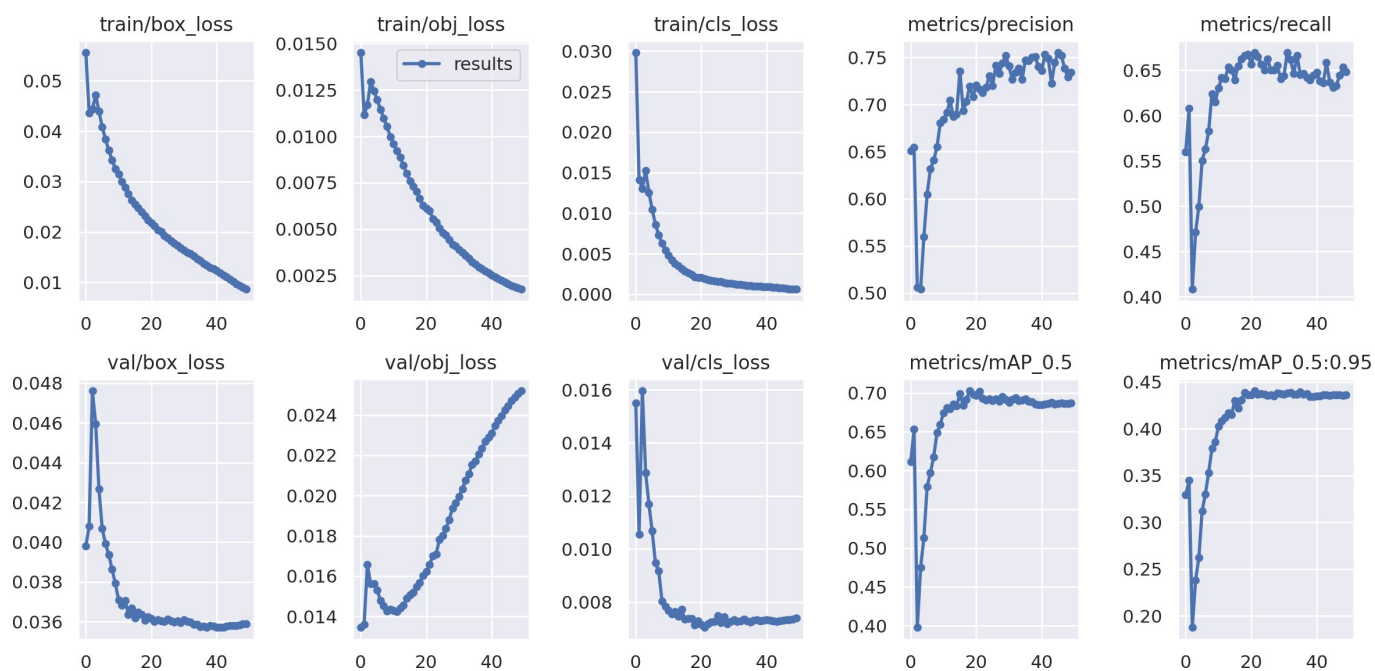


Рис. 21: Poisson

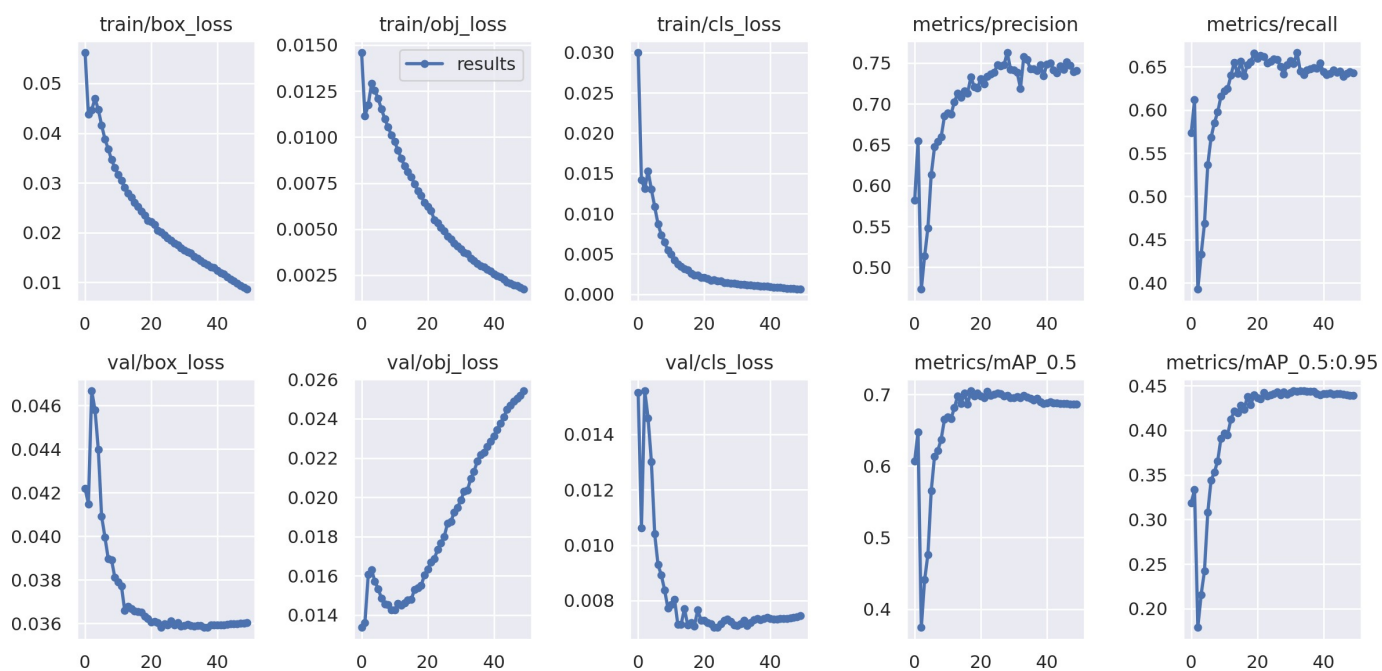


Рис. 22: GAN + Gaussian Blur

Додаток Б

Algorithm 1 Алгоритм вставки

Require: $(x_0, y_0) \geq 0, (x_1, y_1) \leq img.width, img.height$

$scale \leftarrow 100$

$overlap \leftarrow 0.05$

$points \leftarrow []$

while $points_{size} = 0$ **do**

if $scale < 50$ **then**

$overlap \leftarrow overlap + 0.05$

$scale \leftarrow 100$

end if

$mask_x \leftarrow bbox[2] - bbox[0]$

$mask_y \leftarrow bbox[3] - bbox[1]$

for $i = 0, i < width$ **do**

for $j = 0, j < height$ **do**

$x_0, y_0 \leftarrow i, j$

$x_1, y_1 \leftarrow \min(i + mask_x, width), \min(j + mask_y, height)$

for $targetBbox \in targetBoxes$ **do**

$x_{0_{target}}, y_{0_{target}} \leftarrow targetBbox[0], targetBbox[1]$

$x_{1_{target}}, y_{1_{target}} \leftarrow targetBbox[2], targetBbox[3]$

 check 4 corners of the source object

end for

if $overlap \leq \text{area of intersections with all target boxes}$ **then**

$points.append((i, j))$

end if

end for

end for

end while

Додаток В

Метод	Час
Проста вставка	23,3s
Сегментована вставка	20,025s
Гауссовий блюр	26,5s
Пуассонівська вставка	51,6s
Copy-GAN	2,56s
Copy-GAN + Gaussian blur	12,2s

Табл. 2: Час на генерацію 1 зображення.