DOI: https://doi.org/10.26565/2304-6201-2026-69-04

УДК (UDC) 004.93

Коршенко Владислав Сергійович *Аспірант кафедри Кафедри кібербезпеки інформаційних систем, мереж і технологій, старший викладач кафедри математичного моделювання та аналізу даних, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: v.korshenko@karazin.ua
https://orcid.org/0000-0003-2197-072X

Узлов Дмитро Юрійович *Кандидат технічних наук, Директор ННІ КН та ШІ, Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків-22, Україна, 61022*
e-mail: dmytro.uzlov@karazin.ua*;*
https://orcid.org/0000-0003-3308-424X

Оцінка впливу наявності фотореалістичної текстури при генерації синтетичного датасету на точність моделей комп'ютерного зору

Актуальність. Сучасний розвиток комп'ютерного зору стикається з проблемою високої вартості та трудомісткості збору реальних анотованих даних. Використання синтетичних даних, згенерованих у графічних рушіях, є ефективною альтернативою, проте головною перешкодою залишається «розрив між доменами» (domain gap), що знижує точність моделей на реальних зображеннях.
Метою роботи є кількісна оцінка впливу фотореалістичної текстури цільового об'єкта на ефективність детектування моделями YOLO при переході від симуляції до реальності (Sim2Real).
Методологія дослідження базується на проведенні контрольованого експерименту в середовищі Unity, де було згенеровано два ідентичні синтетичні датасети, що відрізнялися лише типом текстури 3D-моделі: високодеталізованою фотореалістичною («Textured») та монохромною білою («White»). Навчання моделей проводилося на базі архітектури YOLOv11s із застосуванням стратегії переносу навчання (transfer learning) та двоетапного процесу тонкого налаштування. Валідація результатів здійснювалася на незалежному наборі виключно реальних фотографій.
Результати. Обидві моделі, що були навчені на двох датасетах («Textured» і «White»), досягли майже ідентичної точності на синтетичних валідаційних даних (mAP@0.5 ≈ 0.995). Однак на реальних фотографіях модель «Textured» продемонструвала в 11.6 разів вищий mAP@0.5, порівняно з результатом моделі «White». Показник повноти (recall) для текстурованої моделі виявився в 10.3 рази вищим, ніж у моделі, що покладалася лише на геометричну форму.
Висновки. Фотореалістична текстура є критично важливим чинником для успішного Sim2Real перенесення. Вона забезпечує формування в ранніх шарах нейронної мережі універсальних низькорівневих ознак, які є необхідними для розпізнавання об'єктів у реальному середовищі. Якісне текстурування 3D-асетів слід розглядати як стратегічний пріоритет, а не допоміжний етап візуалізації.

Ключові слова: *синтетичні дані, комп'ютерний зір, детектування об'єктів, розрив між доменами, робастність моделей, стійкість до зсуву домену.*

Як цитувати: Коршенко В. С., Узлов Д. Ю., "Оцінка впливу наявності фотореалістичної текстури при генерації синтетичного датасету на точність моделей комп'ютерного зору". *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2026. вип. 69. С.41-58. https://doi.org/10.26565/2304-6201-2026-69-04

How to quote: V Korshenko, D.Uzlov, "Assessment of the impact of photorealistic textures on the accuracy of computer vision models using synthetic datasets", *Bulletin of V. N. Karazin Kharkiv National University, series Mathematical Modelling. Information Technology. Automated Control Systems*, vol. 69, pp. 41–58, 2026. https://doi.org/10.26565/2304-6201-2026-69-04 [in Ukrainian]

Вступ

Сучасні досягнення в галузі комп'ютерного зору, особливо в задачах детектування об'єктів та семантичної сегментації, нерозривно пов'язані з використанням глибоких нейронних мереж, які для свого навчання потребують великих обсягів якісно анотованих даних [1].

Проте процес збору та ручної розмітки реальних зображень є надзвичайно трудомістким, фінансово витратним і часто пов'язаний з логістичними або етичними обмеженнями [2].

Ці виклики стають особливо гострими при роботі з рідкісними сценаріями – наприклад, аварійними ситуаціями для безпілотних транспортних засобів, або в умовах, де збір даних є небезпечним чи неможливим [11].

У відповідь на ці проблеми, все більшої популярності набуває використання **синтетичних даних**, згенерованих за допомогою комп'ютерної графіки [1].

Сучасні графічні рушії, такі як *Unity* та *Unreal Engine*, дозволяють створювати фотореалістичні сцени, які містять повністю контрольовані параметри освітлення, текстур і матеріалів, а також автоматично генерують розмітку (обмежувальні прямокутники, маски сегментації, карти глибини) [10].

Цей підхід знімає обмеження, пов'язані з браком даних, забезпечує стабільність якості анотацій та дозволяє масштабувати експерименти без втрати контрольованості [3].

Зростання цього напряму підтверджується академічними аналізами, які вказують на стрімке розширення ринку та застосувань синтетичних даних [18].

Однак, незважаючи на переваги, ключовою проблемою залишається так званий **розрив між доменами** (*domain gap* або *Sim2Real gap*), який проявляється у зниженні точності моделей при переході з симуляційних до реальних даних [12].

Цей розрив зумовлений статистичними та візуальними відмінностями між синтетичними та реальними зображеннями – від текстур і шуму сенсорів до складності освітлення та фону [13].

Для подолання цієї проблеми наукова спільнота розробила три основні стратегії [6]:

1. Доменна рандомізація (Domain Randomization, DR) – варіювання текстур, освітлення, ракурсів і матеріалів у широкому (навіть не реалістичному) діапазоні, що дозволяє моделі навчитися узагальнювати незалежно від конкретних умов [3], [4];
2. Доменна адаптація (Domain Adaptation, DA) – статистичне узгодження розподілів ознак між доменами з використанням нейронних мереж або змагальних методів [5], [6];
3. Підвищення фотореалізму (Photorealism) – фізично коректне відтворення текстур, матеріалів і освітлення для зменшення візуальної різниці між симуляційним і реальним світом [7], [8].

Саме третій підхід є фокусом цього дослідження.

Попередні роботи Hinterstoisser et al. [7] продемонстрували, що використання високоякісних, фотореалістичних синтетичних даних може забезпечити продуктивність, наближену до результатів навчання на реальних вибірках. Водночас внесок окремих аспектів реалізму – зокрема наявності або відсутності текстури об'єкта – залишається недостатньо вивченим.

Дослідження Jackson et al. [4] показують, що навіть у межах доменної рандомізації складність текстур підвищує точність моделей, що вказує на ключову роль текстурної інформації у формуванні переносимих ознак.

Теоретичну основу цього припущення заклали Yosinski et al. [8], які експериментально довели, що ранні шари згорткових нейронних мереж навчаються розпізнавати універсальні низькорівневі ознаки (градієнти, краї, кольорові переходи), тоді як пізніші – спеціалізовані, залежні від конкретної задачі.

Це означає, що саме якісна текстуризація синтетичних об'єктів може покращувати переносимість моделі при переході між доменами, оскільки збагачує її вхідні сигнали низькорівневими характеристиками, спільними для обох середовищ.

Метою даної роботи є експериментальна та кількісна оцінка впливу фотореалістичної текстури цільового об'єкта в синтетичних датасетах на ефективність детектування об'єктів моделлю архітектури YOLO[10] при переході від синтетичних до реальних даних (Sim2Real).

Для досягнення поставленої мети було розроблено контрольований експеримент, у якому варіювався виключно фактор текстуризації 3D-моделі, тоді як усі інші параметри сцени, процесу навчання та валідації залишалися незмінними.

Висунута гіпотеза полягає в тому, що саме фотореалістичні текстури є одним із ключових чинників зменшення розриву *Sim2Real*, підвищуючи точність та узагальнюваність моделей комп'ютерного зору.

2. Матеріали та методи

Цей розділ детально описує методологію, використану для перевірки нашої дослідницької гіпотези. Експериментальний дизайн було розроблено таким чином, щоб забезпечити контрольовані умови для ізоляції та кількісної оцінки впливу фотореалістичної текстури на ефективність моделі комп'ютерного зору. Процес дослідження було послідовно розділено на три основні етапи: (1) генерація двох варіантів синтетичного датасету, що відрізняються лише однією цільовою характеристикою; (2) навчання двох ідентичних моделей детектування об'єктів на цих датасетах за однакових умов; (3) валідація та порівняльний аналіз продуктивності моделей на незалежних тестових наборах, що містять реальні зображення.

Основою нашого експерименту було створення висококонтрольованих синтетичних наборів даних. Цей підхід дозволив нам ізолювати вплив текстури об'єкта як єдиної змінної, усунувши сторонні фактори, які могли б вплинути на результати навчання моделі. Для генерації зображень було обрано ігровий рушій **Unity**, оскільки він надає широкий набір інструментів для створення фотореалістичних сцен, симуляції освітлення та автоматизації процесу збору даних з ідеально точною розміткою.

Для створення віртуального середовища ми використали готову 3D-сцену лісу (Рис. 1), використану на підставі ліцензії Unity Asset Store та її умов використання. Ця сцена створювала оптимальний візуальний контекст з природним оточенням, що містить різноманітні фонові об'єкти (дерева, кущі, траву) та неоднорідний ландшафт, що дозволило згенерувати складні для аналізу зображення.

Рис. 1: Сцена лісу, що була використана як оточення для розміщення моделі цільового об’єкту
Figure 1: Forest scene used as the environment for placing the target object model

В якості цільового об'єкта для детектування було обрано 3D-модель качки (Рис. 2). Цей вибір був зумовлений трьома основними причинами. По-перше, з точки зору геометрії, об'єкт має помірну складність: він поєднує плавні контури та вигнуті поверхні, але не має надмірно дрібних деталей, таких як хутро чи окремі пера, що могло б ускладнити аналіз впливу саме базової текстури. По-друге, різноманітність природних забарвлень качок робить їх чудовим прикладом для вивчення важливості текстурних ознак. По-третє, наявність високодеталізованих 3D-моделей у вільному доступі спростила підготовку до експерименту. Модель було розміщено у центрі сцени, щоб забезпечити достатній простір для позиціонування камери з різних ракурсів.

Рис. 2: Сцена з розміщеним у центрі цільовим об’єктом
Figure 2: Scene with a target object placed in the center

Освітлення в сцені було реалізовано за допомогою одного глобального джерела світла типу Directional Light, що імітує сонячне світло. Для досягнення м'яких та реалістичних тіней було встановлено параметр Shadow Type у значення Soft Shadows, а інтенсивність світла Intensity – 2.27. Налаштування оточення, такі як матеріал неба (Skybox) та загальне розсіяне світло (Ambient Color), також були сконфігуровані для створення природної денної атмосфери. Від цих параметрів залежали ключові візуальні аспекти сцени: яскравість, контрастність, колір тіней та відблиски на поверхнях.

Збір зображень виконувався за допомогою віртуальної камери (Рис. 3), позиція якої контролювалася спеціально розробленим скриптом. Цей скрипт дозволив автоматизувати процес зйомки, систематично переміщуючи камеру навколо цільового об'єкта. Камера оберталася на фіксованій відстані (Distance = 35 одиниць) з визначеним кроком по горизонталі (Horizontal Step = 20 градусів) та вертикалі (Vertical Step = 30 градусів). Такий підхід гарантував, що об'єкт буде знято з великої кількості різноманітних, але відтворюваних ракурсів.

Рис. 3: Віртуальна камера, розміщена на сцені, та її поле зору
Figure 3: Virtual camera placed on the scene and its field of view

Головною умовою експерименту було створення двох наборів даних, які б відрізнялися виключно текстурою цільового об'єкта. Усі інші параметри – геометрія 3D-моделі, налаштування сцени, освітлення, траєкторія руху камери – залишалися абсолютно ідентичними для обох випадків.

Датасет 1 - "Textured" (Верхня половина Рис. 4): Для цього набору даних на 3D-модель качки було накладено високодеталізовану, фотореалістичну текстуру, яка імітувала природне забарвлення птаха.

Датасет 2 - "White" (Нижня половина Рис. 4): У цьому випадку та сама 3D-модель використовувалася з простою монохромною білою текстурою. Вона не несла жодної інформації про колір чи патерни, дозволяючи моделі спиратися лише на геометричну форму об'єкта, його силует та затінення.

Такий підхід дозволив нам створити дві паралельні реальності для навчання, де єдиною відмінністю була наявність або відсутність текстурних ознак.

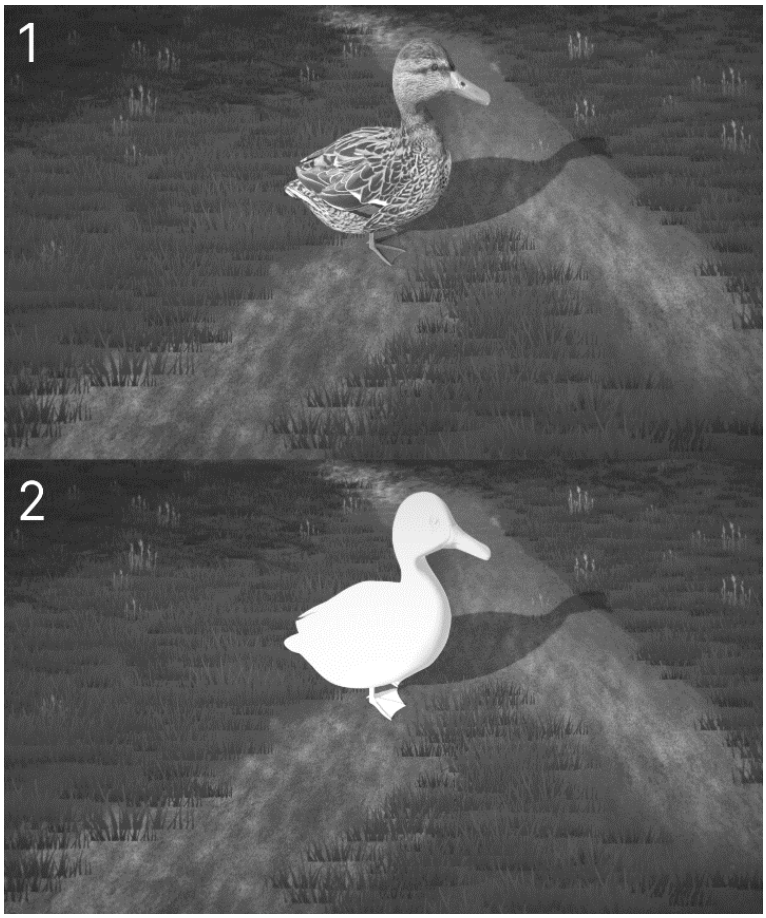

Рис. 4: Побічне порівняння ідентичних кадрів з датасету "Textured" (вгорі) та "White" (внизу)
Figure 4: Side-by-side comparison of identical frames from the "Textured" (top) and "White" (bottom) datasets

Процес розмітки даних (створення обмежувальних прямокутників, або bounding boxes) був повністю автоматизований засобами Unity. Для кожного згенерованого кадру скрипт розраховував точні екранні координати видимої частини цільового об'єкта. Це досягалося шляхом випускання променів (RayCast) з камери до кожного пікселя екрана; якщо промінь перетинав 3D-модель об'єкта, його координати використовувалися для визначення крайніх точок (верхньої, нижньої, лівої, правої), на основі яких і будувався обмежувальний прямокутник . Координати зберігалися у текстовому файлі у форматі, сумісному з YOLO.

Для кожного з двох сценаріїв ("Textured" та "White") було згенеровано по 342 унікальних зображення. Отримані дані були розділені на навчальну та валідаційну вибірки у співвідношенні приблизно 80/20:

- 273 зображення для навчання.
- 69 зображень для валідації.

Розподіл було здійснено таким чином, щоб зображення з однакових ракурсів в обох датасетах потрапляли у відповідні вибірки (навчальну до навчальної, валідаційну до валідаційної), що забезпечило повну узгодженість умов для подальшого навчання та порівняння моделей.

2.1 Архітектура та навчання моделі детектування об'єктів

Для об'єктивного порівняння впливу текстури було критично важливо, щоб обидві моделі ("textured" та "white") були ідентичними за архітектурою та навчалися за абсолютно однакових умов. Цей підрозділ описує вибір моделі, стратегію її навчання та конкретні гіперпараметри, що використовувалися для забезпечення відтворюваності експерименту.

В якості базової архітектури для вирішення задачі детектування об'єктів було обрано модель YOLOv11s з репозиторію Ultralytics. Ця модель є представником сімейства YOLO (You Only Look Once), яке добре відоме своїм балансом між швидкістю роботи та точністю. Конфігурація "s" (small) є полегшеною версією, що містить близько 7 мільйонів параметрів. Такий вибір був зумовлений прагматичними міркуваннями: малий обсяг ваг моделі значно знижує вимоги до обчислювальних ресурсів, зокрема до обсягу відеопам'яті (VRAM), що дозволило проводити серію експериментів на GPU середнього класу та прискорити ітерації дослідження.

Навчання глибокої нейронної мережі «з нуля» на нашому відносно невеликому навчальному наборі з 273 синтетичних зображень неминуче призвело б до сильного перенавчання (*overfitting*), коли модель ідеально запам’ятовує навчальні приклади, але втрачає здатність до узагальнення на нових даних. Щоб уникнути цієї проблеми, ми застосували стратегію переносу навчання (*transfer learning*).

Ми використали модель YOLOv11s, попередньо навчену на великому та різноманітному датасеті COCO (Common Objects in Context), який містить мільйони об’єктів різних класів.

Згідно з Yosinski et al. [8], ранні шари згорткових мереж формують універсальні низькорівневі ознаки (градієнти, контури, текстури), які легко переносяться між доменами.

Це пояснює, чому використання попередньо натренованих ваг YOLOv11, отриманих на великій базі COCO, сприяє кращому *transfer learning* під час навчання на синтетичних даних.

Така здатність до адаптації є ключовою для зменшення розриву *Sim2Real* та забезпечує узгодженість результатів між симульованим і реальним середовищами.

Цей підхід ґрунтується на фундаментальному спостереженні, що ранні згорткові шари нейронної мережі навчаються розпізнавати загальні низькорівневі та середньорівневі ознаки, такі як краї, градієнти, кольорові плями та базові текстури. Ці ознаки є універсальними і корисними для широкого спектра задач комп’ютерного зору. Натомість, більш глибокі шари, що знаходяться ближче до виходу мережі, спеціалізуються на виявленні високоспецифічних ознак, що стосуються конкретних класів із початкового датасету (наприклад, COCO).

Таким чином, стратегія полягала в тому, щоб зберегти корисні універсальні фільтри з «кістяка» (*backbone*) моделі та адаптувати лише її «голову» (*head*) до нашої специфічної задачі – детектування одного класу «качка».

Для забезпечення стабільного та ефективного навчання було розроблено двоетапну стратегію, яка поєднувала "заморозку" шарів та тонке налаштування.

Етап 1: "Розігрів" голови детектора (Head Warm-up). На цьому етапі метою було адаптувати лише вихідні, класифікаційні шари моделі до нового завдання, не зачіпаючи стабільні ваги попередньо навченого кістяка. Для цього перші 10 шарів нейронної мережі були "заморожені" (параметр freeze=10), тобто їхні ваги не оновлювалися під час градієнтного спуску. Навчання тривало протягом 5 епох з відносно високою швидкістю навчання (learning rate) 10^{-4}. Такий підхід, рекомендований Ultralytics, дозволяє швидко адаптувати модель до нового класу без ризику руйнування універсальних фільтрів.

Етап 2: Тонке налаштування всієї мережі (Full Network Fine-tuning). Після того, як "голова" моделі була адаптована, ми "розморозили" всю мережу і продовжили її донавчання протягом 80 епох. На цьому етапі швидкість навчання було значно знижено до $3*10^{-5}$. Мета цього етапу – дозволити всім шарам моделі, включно з ранніми, тонко підлаштуватися під специфіку наших синтетичних даних, зберігаючи при цьому стабільність градієнтів завдяки низькій швидкості навчання. Вибір такої комбінації (короткий "розігрів" + тривале тонке налаштування) спирався на результати досліджень, які показали, що такий підхід підвищує стабільність та кінцеву точність моделі, особливо в умовах значного розриву між доменами.

В Таблиці 1 наведено параметри доетапного навчання.

Таблиця 1. Параметри двоетапного навчання
Table 1. Parameters of two-stage training

Етап	Шари для навчання	Епохи	Швидкість навчання (LR)	Заморожені шари (Freeze)	Мета
1	Головні детекторні head-шари	5	$1*10^{-4}$	0-9	Адаптувати класифікатор до класу duck без руйнування універсальних фільтрів.
2	Уся мережа	80	$3*10^{-5}$	–1	Тонко підлаштувати усі рівні, зберігаючи стабільність градієнтів.

Для гарантування повної відтворюваності результатів усі експерименти проводилися у детермінованому режимі.

Було зафіксовано початкове значення генератора псевдовипадкових чисел (seed = *42*) для всіх етапів навчання, що забезпечило ідентичну ініціалізацію ваг і послідовність подачі даних при кожному запуску моделі.

Такий підхід відповідає базовим принципам наукової відтворюваності в машинному навчанні, описаним у роботі Picard [19], де наголошується на важливості контролю стохастичних процесів для коректного порівняння результатів між експериментами.

У ході навчання застосовувались такі основні гіперпараметри:

- Batch size: 32
- Learning rate: $1*10^{-4}$ (етап 1) / $3*10^{-5}$ (етап 2)
- Оптимізатор: Adam [20]
- Функція втрат: комбінація *Binary Cross-Entropy* та *Complete IoU Loss (CIoU)* [21]
- Розмір вхідних зображень: 640×640 пікселів
- Freeze: 10 шарів на етапі попередньої адаптації
- Epochs: 5 + 80 (двоступеневе навчання)

Оптимізатор Adam (Adaptive Moment Estimation) було обрано через його ефективність і швидку збіжність при донавчанні моделей на невеликих наборах даних.

Як показано у роботі Kingma та Ba [20], цей алгоритм адаптивно коригує швидкість навчання для кожного параметра, поєднуючи переваги AdaGrad і RMSProp, що забезпечує швидшу стабілізацію градієнтів.

Функція втрат поєднує *Binary Cross-Entropy (BCE)* для класифікаційної складової та *Complete IoU Loss (CIoU)* для регресії обмежувальних рамок.

Згідно з дослідженням Zheng et al. [21], CIoU враховує не лише площу перетину рамок, а й відстань між їх центрами та співвідношення сторін, що дозволяє моделі ефективніше локалізувати об’єкти в кадрі порівняно зі звичайним IoU.

Додатково використовувалась стратегія плавного зниження швидкості навчання за косинусним законом (Cosine Annealing Scheduler) [22], реалізована параметром cos_lr=True.

Метод був уперше описаний у роботі Loshchilov і Hutter [22] і довів свою ефективність у запобіганні «перестрибуванню» через локальні мінімуми, забезпечуючи більш плавну збіжність і кращу генералізацію на пізніх етапах навчання.

Також застосовувався механізм «прогріву» (Warm-up) [23], протягом перших двох епох (warmup_epochs=2), коли швидкість навчання поступово збільшувалася від нуля до встановленого значення.

Як показано у роботі Goyal et al. [23], такий підхід стабілізує оптимізацію, особливо під час початкових етапів тренування моделей із великим розміром батчу.

У сукупності ці процедури – детермінізація, адаптивна оптимізація, комбінована функція втрат, плавна зміна швидкості навчання та етап прогріву – забезпечили стабільну збіжність, стійкість до коливань градієнтів і відтворюваність результатів експериментів.

2.2 Експериментальна валідація та метрики оцінювання

Після завершення навчання двох моделей – "textured" та "white" – наступним кроком стала їхня всебічна валідація та порівняльний аналіз. Для отримання об'єктивних та надійних результатів було розроблено методологію оцінювання, яка включала використання двох спеціально підготовлених тестових наборів даних та набір стандартних метрик якості для задач детектування об'єктів.

Щоб оцінити продуктивність моделей у різних умовах, ми підготували два незалежних тестових датасети, кожен з яких мав свою специфічну мету.

Тестовий датасет 1 - змішаний набір для базової оцінки. Цей набір містив 30 синтетичних зображень, взятих порівну з валідаційних вибірок обох навчальних датасетів, та 12 реальних фотографій цільових об'єктів (качок).

Основна мета цього датасету – виконати "перевірка адекватності" (sanity check). По-перше, оцінка на знайомих синтетичних даних дозволила переконатися, що обидві моделі успішно засвоїли свої відповідні домени ("in-domain performance"). По-друге, невелика кількість реальних зображень дала змогу провести первинну, "м'яку" перевірку здатності моделей до узагальнення (generalization) та перенесення знань на реальний світ.

Тестовий датасет 2 - реальний набір для повноцінної Cross-Domain валідації. Цей набір даних був розроблений для жорсткого тестування моделей в умовах, максимально наближених до практичного застосування, і повністю складався з реальних фотографій.

Він містив:

- 50 позитивних прикладів - реальні фотографії качок у різних середовищах, позах та умовах освітлення.
- 50 негативних прикладів, які, у свою чергу, були поділені на дві групи для перевірки стійкості до різних типів помилок:
- 25 "важких" негативних прикладів: фотографії інших птахів (гусей, лебедів, чайок), які мають схожий силует або знаходяться у схожому контексті. Це дозволило оцінити здатність моделей відрізняти цільовий клас від схожих нецільових об'єктів.
- 25 "легких" негативних прикладів: фотографії природних пейзажів (ліс, озеро) без будь-яких птахів чи тварин. Ця група була призначена для перевірки моделей на схильність до хибних спрацювань (false positives) на складних фонах.

Цей датасет є основним інструментом для відповіді на наше дослідницьке питання. Оцінка на ньому дозволяє виміряти реальну ефективність перенесення знань з симуляції в реальність (Sim2Real), а також оцінити надійність та практичну придатність кожної з моделей.

Для кількісної оцінки та порівняння продуктивності навчених моделей було використано набір стандартних метрик для задач детектування об'єктів. Якість детекції оцінювалася за метриками Precision (Точність), Recall (Повнота) та інтегрального показника mean Average Precision (mAP).

Для всебічного аналізу ми розраховували дві варіації mAP, що відрізняються за вимогами до точності локалізації:

- mAP@0.5 (mAP50): Ця метрика оцінює здатність моделі правильно класифікувати та в цілому локалізувати об'єкт, використовуючи поріг Intersection over Union (IoU) 0.5.
- mAP@0.5-0.95 (стандартна метрика COCO): Ця більш сувора метрика оцінює точність обмежувальних прямокутників, усереднюючи mAP по діапазону порогів IoU від 0.5 до 0.95.

Такий набір метрик дозволив нам провести комплексний аналіз, оцінивши як загальну здатність моделей до виявлення об'єктів, так і точність їхньої локалізації.

3. Результати

У цьому розділі представлено кількісні результати експериментів, проведених згідно з методологією, описаною в попередньому розділі. Дані викладено об'єктивно та послідовно, щоб продемонструвати ефективність обох навчених моделей – "textured" та "white" – на різних етапах валідації. Спочатку наведено результати навчання на синтетичних даних, що встановлюють базовий рівень продуктивності моделей у межах їхніх навчальних доменів. Далі представлено порівняльні результати тестування на змішаному та повністю реальному датасетах, що дозволяє оцінити здатність кожної моделі до перенесення знань та її ефективність в умовах зсуву домену.

3.1. Результати навчання та валідації на синтетичних даних

Першочерговим етапом аналізу була перевірка якості навчених моделей на валідаційних вибірках, що походили з тих самих синтетичних доменів, що й навчальні дані. Цей крок був необхідний для того, щоб встановити базовий рівень продуктивності та переконатися, що обидві моделі – "textured" та "white" – успішно засвоїли візуальні закономірності своїх відповідних наборів даних.

Результати внутрішньодоменної валідації (in-domain validation) продемонстрували, що обидві моделі досягли майже ідеальних та практично ідентичних показників. Для обох конфігурацій інтегральний показник mAP@0.5 склав приблизно 0.995. Це свідчить про те, що ні недонавчання (underfitting), ні суттєві відмінності в ємності засвоєної інформації не були факторами в експерименті. Обидві моделі повністю опанували свої синтетичні домени, що створило надійну основу для подальшого порівняння їхньої здатності до перенесення знань на реальні дані.

Висока якість навчання підтверджується візуальними метриками. Нормалізовані матриці сплутування для обох моделей демонструють 100% точність на валідаційній вибірці, де всі об'єкти класу "качка" були правильно ідентифіковані, а хибні спрацювання на фоні були відсутні (Рис. 5, Рис. 6).

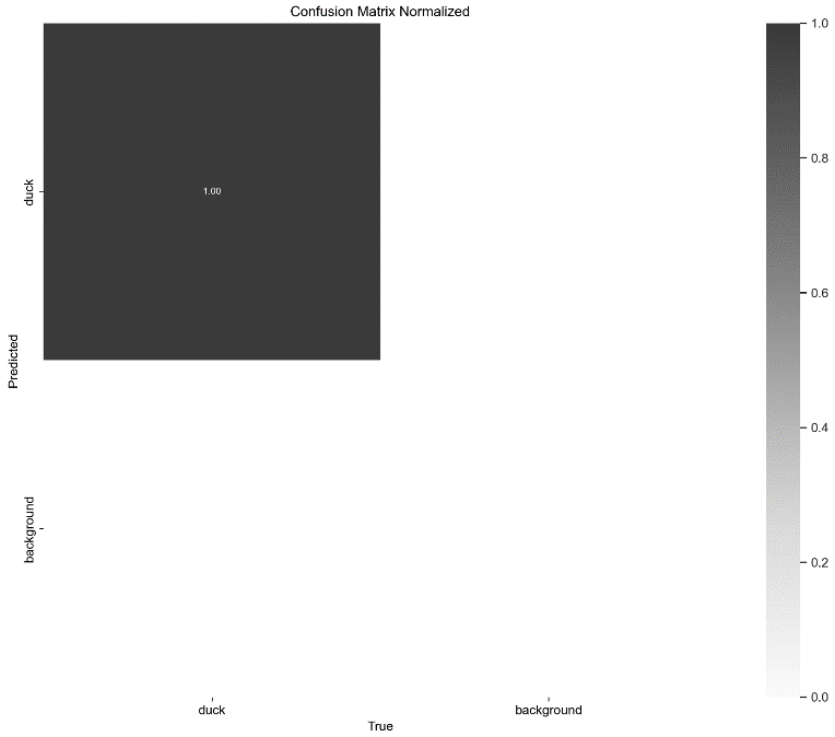

Рис. 5: Нормалізована матриця сплутування для моделі "textured" на синтетичному валідаційному датасеті

Figure 5: Normalized confusion matrix for the "textured" model on a synthetic validation dataset

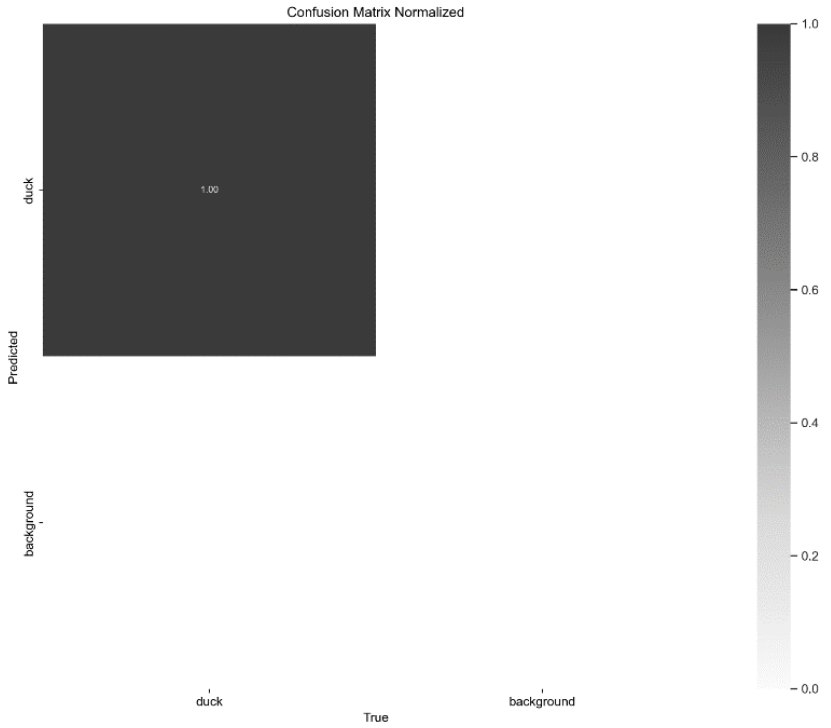

Рис. 6: Нормалізована матриця сплутування для моделі "white" на синтетичному валідаційному датасеті
Figure 6: Normalized confusion matrix for the "white" model on a synthetic validation dataset

Аналогічно, криві точності-повноти (Precision-Recall curves) для обох моделей мають прямокутну форму (Рис. 7, 8), що є характерним для детектора з дуже високою продуктивністю, де і точність, і повнота близькі до 1.0 у широкому діапазоні порогів впевненості.

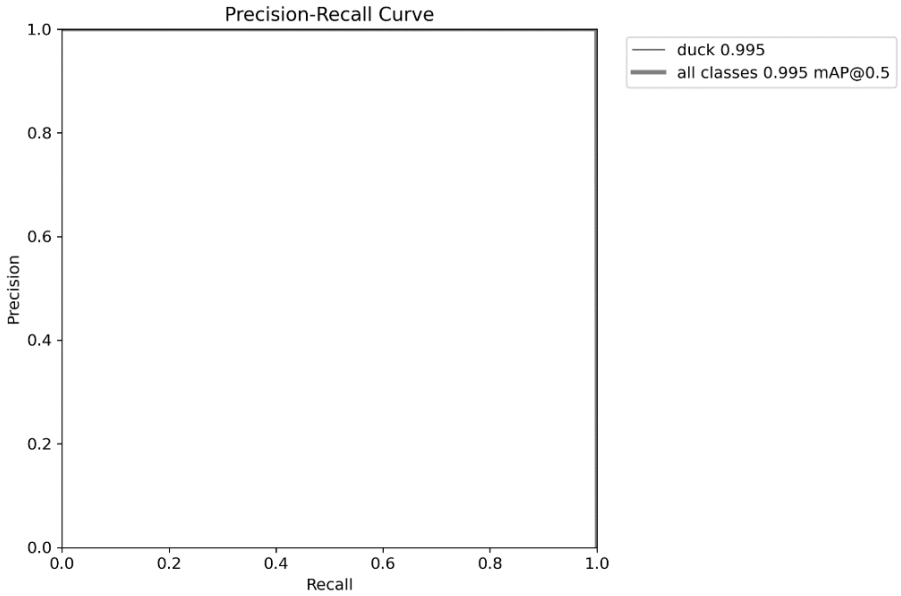

Рис. 7: Крива PR для моделі "textured" на синтетичному валідаційному датасеті
Figure 7: PR curve for the "textured" model on a synthetic validation dataset

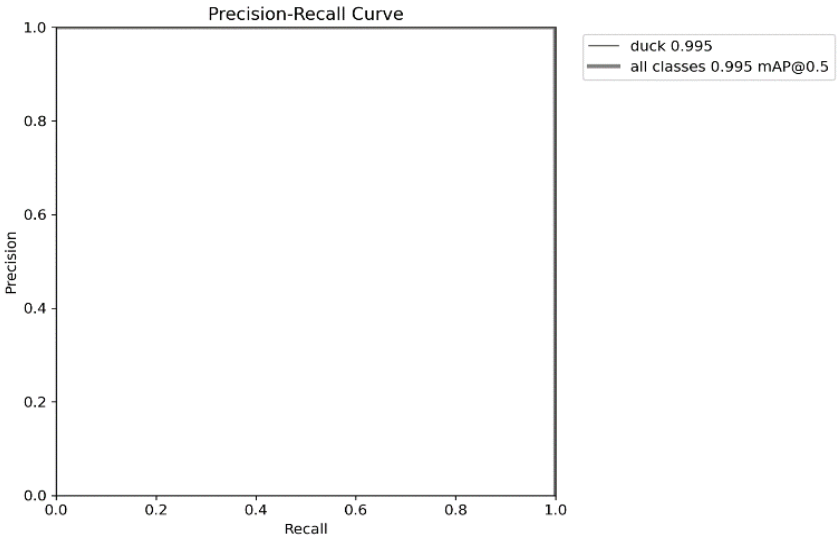

Рис. 8: Крива PR для моделі "white" на синтетичному валідаційному датасеті
Figure 8: PR curve for the "white" model on a synthetic validation dataset

Таким чином, результати цього етапу підтвердили, що обидві моделі були успішно та стабільно навчені за однакових умов, що дозволяє впевнено стверджувати, що будь-які подальші розходження в їхній продуктивності на реальних даних будуть зумовлені виключно впливом досліджуваного фактора – наявністю або відсутністю фотореалістичної текстури.

3.2. Порівняльна оцінка на змішаному тестовому датасеті

Наступним етапом було тестування обох моделей на змішаному датасеті, що складався як із синтетичних, так і з реальних зображень. Цей експеримент дозволив оцінити продуктивність моделей в умовах "м'якого" зсуву домену, коли детектор стикається з першими прикладами з реального світу.

Модель, навчена на даних з фотореалістичною текстурою, продемонструвала досить високу продуктивність. Показник mAP@0.5 досяг 0.682, що свідчить про спроможність моделі узагальнювати отримані знання. Показник точності (precision) залишився на високому рівні 0.965, однак повнота (recall) знизилася до 0.534. Це вказує на те, що модель, хоч і робила впевнені та правильні детекції, пропускала майже половину цільових об'єктів на реальних зображеннях.

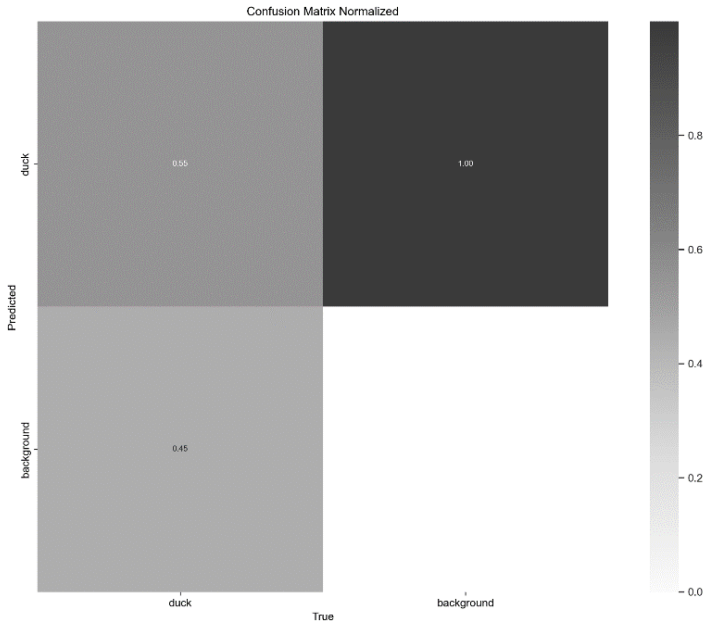

Рис. 9: Нормалізована матриця сплутування для моделі "textured" на змішаному датасеті
Figure 9: Normalized confusion matrix for the "textured" model on a mixed dataset

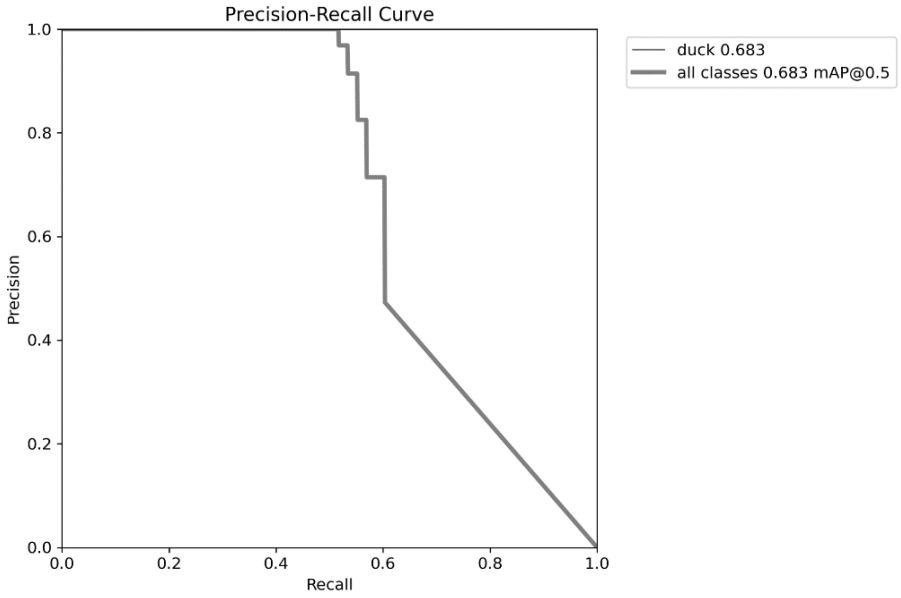

Рис. 10: Крива PR для моделі "textured" на змішаному датасеті
Figure 10: PR curve for the "textured" model on a mixed dataset

Модель, навчена на даних з монохромною білою текстурою, показала значно нижчі результати за всіх ключових метриках. Показник mAP@0.5 склав лише 0.421, а більш суворий mAP@0.5-0.95 – 0.359. Особливо помітним було падіння повноти (recall), яка становила всього 0.310. Це означає, що модель, яка спиралася переважно на геометричну форму, змогла правильно ідентифікувати менше третини цільових об'єктів у нових умовах.

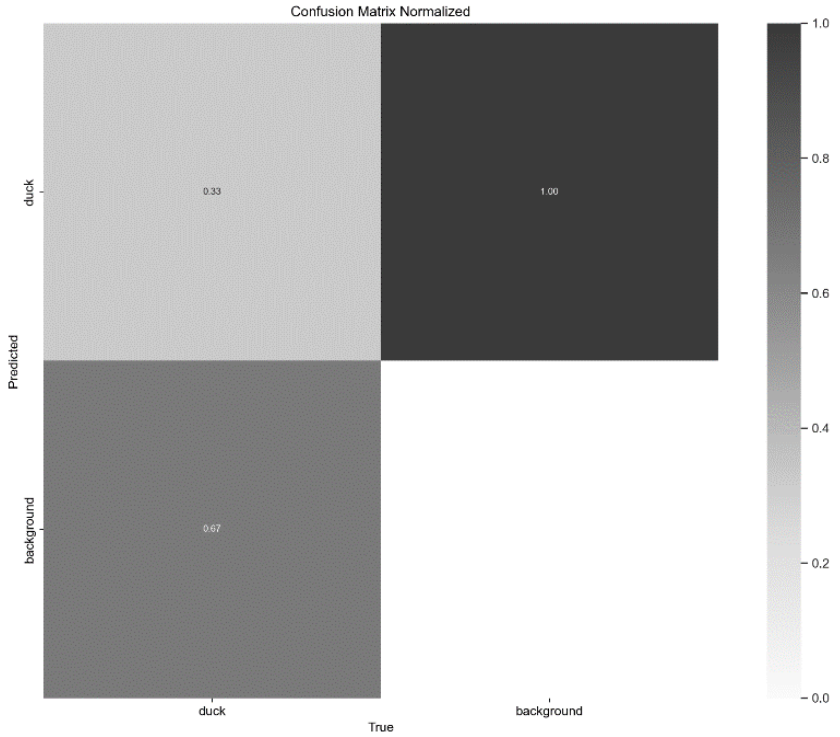

Рис. 11: Нормалізована матриця сплутування для моделі "white" на змішаному датасеті
Figure 11: Normalized confusion matrix for the "white" model on a mixed dataset

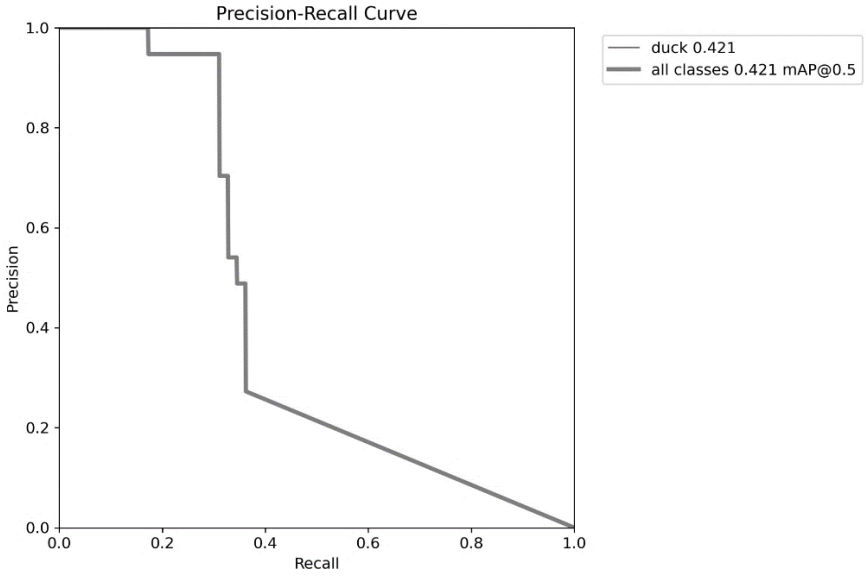

Рис. 12: Крива PR для моделі "white" на змішаному датасеті
Figure 12: PR curve for the "white" model on a mixed dataset

Для наочного порівняння продуктивності обох моделей на змішаному тестовому датасеті, ключові метрики зведено в Таблицю 2.

Таблиця 2. Порівняння продуктивності моделей на змішаному тестовому датасеті
Table 2. The comparison of model performance on the mixed test dataset

Метрика	Модель "Textured"	Модель "White"	Перевага "Textured" моделі (%)
mAP@0.5	0.682	0.421	+61.9%
mAP@0.5-0.95	0.612	0.359	+70.5%
Precision	0.965	0.855	+12.9%
Recall	0.534	0.310	+72.3%

Представлені дані чітко демонструють, що навіть за умов незначного зсуву домену модель, навчена з використанням фотореалістичної текстури, показує суттєву перевагу за всіма ключовими показниками, особливо за здатністю виявляти об'єкти (recall) та загальною якістю детекції (mAP).

3.3. Порівняльна оцінка на реальному тестовому датасеті

Ключовим етапом дослідження була валідація моделей на тестовому наборі, що складався виключно з реальних фотографій. Цей експеримент був розроблений для оцінки продуктивності в умовах повного розриву між доменами (cross-domain), що дозволяє визначити реальну практичну цінність кожного з підходів до генерації синтетичних даних.

При тестуванні на реальних даних продуктивність моделі, навченої на текстурованих зображеннях, очікувано значно знизилася порівняно з попередніми тестами. Це падіння є прямим наслідком "domain gap". Тим не менш, модель зберегла певну здатність до детектування об'єктів. Показник **mAP@0.5 склав 0.059**, а повнота (recall) – **0.123**. Це означає, що модель все ще змогла правильно ідентифікувати приблизно 12% цільових об'єктів у абсолютно новому для неї візуальному домені.

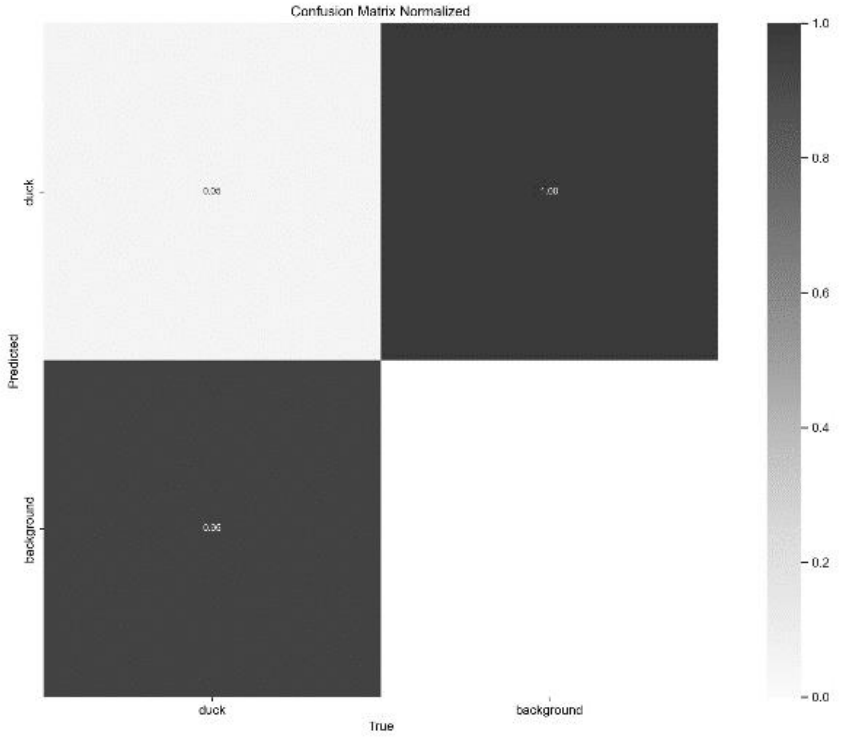

Рис. 13: Нормалізована матриця сплутування для моделі "textured" на реальному датасеті
Figure 13: Normalized confusion matrix for the "textured" model on a real dataset

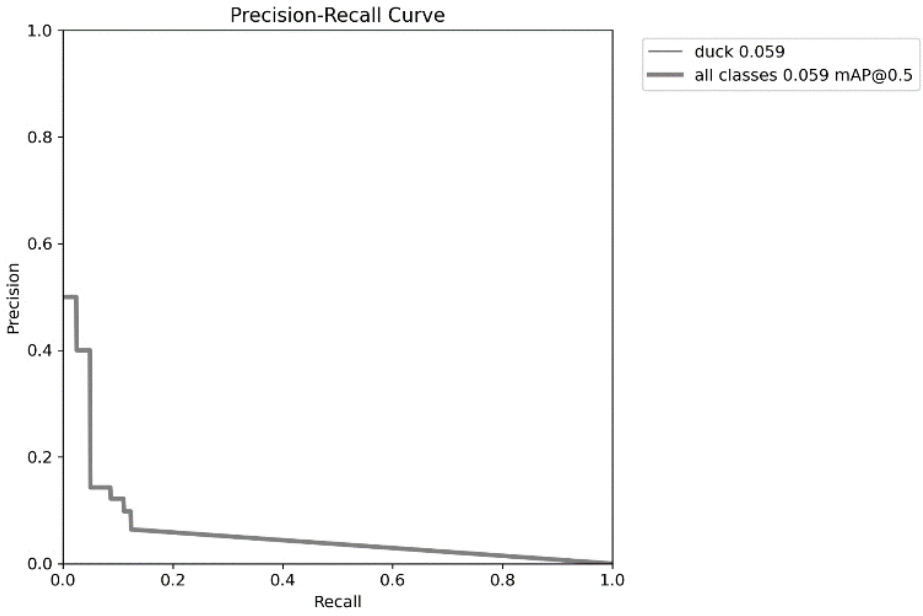

Рис. 14: Крива PR для моделі "textured" на реальному датасеті
Figure 14: PR curve for the "textured" model on a real dataset

Модель, навчена на даних без текстури, продемонструвала майже повну нездатність переносити знання на реальні зображення. Її продуктивність виявилася на порядок нижчою, ніж у текстурованої моделі, а ключові метрики наблизилися до нуля. Показник **mAP@0.5 склав лише 0.005**, а повнота (recall) – **0.012**. Фактично, модель змогла знайти лише близько 1% цільових об'єктів, що свідчить про те, що знання про суто геометричну форму об'єкта виявилися майже марними при переході до реального світу з його різноманіттям кольорів, освітлення та фонів.

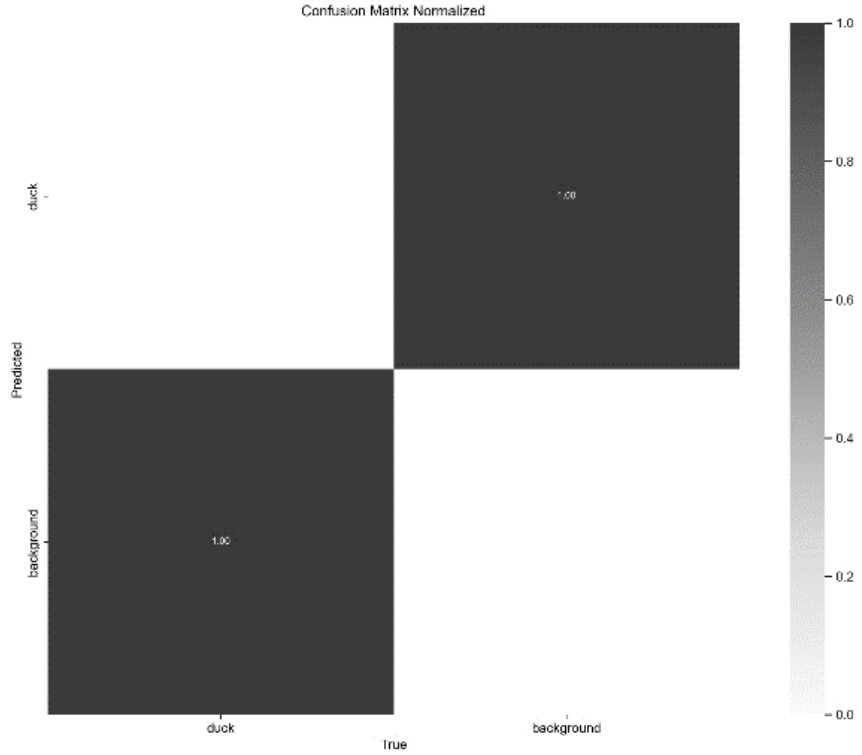

Рис. 15: Нормалізована матриця сплутування для моделі "white" на реальному датасеті
Figure 15: Normalized confusion matrix for the "white" model on a real dataset

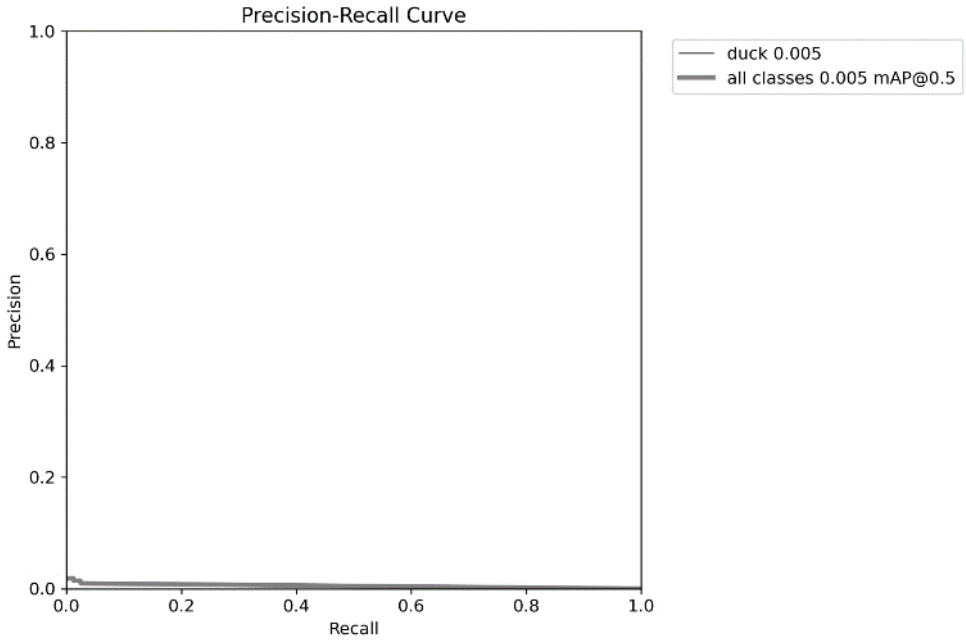

Рис. 16: Крива PR для моделі "white" на реальному датасеті
Figure 16: PR curve for the "white" model on a real dataset

Для максимально наочного порівняння ефективності обох моделей в умовах реального застосування, їхні ключові метрики зведено в Таблицю 3.

Таблиця 3. Порівняння продуктивності моделей на реальному тестовому датасеті
Table 3. The comparison of model performance on the real-world test dataset

Метрика	Модель "Textured"	Модель "White"	Перевага "Textured" моделі
mAP@0.5	0.058	0.005	Показник вищий у 11.6 разів
mAP@0.5-0.95	0.018	0.001	Показник вищий у 18.0 разів
Precision	0.064	0.015	Показник вищий у 4.3 рази
Recall	0.123	0.012	Показник вищий у 10.3 рази

Отримані дані однозначно свідчать про те, що за умов повного розриву між доменами модель, яка навчалася на синтетичних даних з фотореалістичною текстурою, демонструє на порядок вищу продуктивність. У той час як ефективність моделі, що покладалася лише на силует, виявилася близькою до нуля. Ці кількісні результати слугують емпіричною основою для подальшого аналізу та обговорення.

4. Висновок

У межах цього дослідження було проведено кількісну оцінку впливу фотореалістичної текстури цільових об’єктів у синтетичних датасетах на ефективність детектування моделями архітектури YOLOv11s при переході від симуляції до реальності (Sim2Real).

Результати експерименту повністю підтвердили висунуту гіпотезу: фотореалістична текстура є не декоративним елементом, а критично важливою структурною складовою, що забезпечує переносимість ознак та подолання «розриву між доменами» (domain gap).

Хоча обидві моделі («Textured» та «White») продемонстрували майже ідеальну та ідентичну точність на синтетичних даних (mAP@0.5 ≈ 0.995), на реальних фотографіях модель, навчена на текстурованих зображеннях, продемонструвала у 11.6 разів вищий mAP@0.5 та у 10.3 рази вищий показник повноти (recall). Це доводить, що знання про суто геометричну форму об'єкта є недостатніми для розпізнавання у реальному середовищі.

Отримані результати узгоджуються з попередніми дослідженнями робастності інтелектуальних систем у прикладних задачах [24], де показано, що стійкість моделей до зсуву вхідних розподілів є критичною для їх практичного застосування.

Висока ефективність текстурованого підходу пояснюється здатністю ранніх шарів згорткової нейронної мережі формувати універсальні низькорівневі ознаки (градієнти, кольорові переходи, мікропатерни), які є інваріантними до зміни доменів. Відсутність таких ознак у моделі без текстури призводить до її «візуальної сліпоти» на реальних об’єктах.

Найважливішим практичним висновком є те, що якісне текстурування 3D-моделей слід розглядати як стратегічний пріоритет процесу генерації даних, а не як допоміжний етап візуалізації. Інвестиції в реалістичність текстур на мікрорівні прямо конвертуються у стабільність та надійність систем комп’ютерного зору в промислових умовах.

На відміну від хаотичної доменної рандомізації (DR), метод контрольованого фотореалізму забезпечує баланс між варіативністю та фізичною правдоподібністю, створюючи синтетичні сцени з природною статистикою, що зменшує потребу в додатковій адаптації до реальних даних.

Перспективним напрямом подальших робіт є поєднання фотореалістичного текстурування з методами доменної адаптації та рандомізації матеріалів. Це дозволить забезпечити оптимальну робастність моделей для широкого спектра практичних застосувань – від автономних транспортних засобів до робототехнічних систем спостереження.

СПИСОК ЛІТЕРАТУРИ

1. Man, K.; Chahl, J. A Review of Synthetic Image Data and Its Use in Computer Vision. *J. Imaging* 2022, 8, 310.
2. Mumuni, A.; Mumuni, F. A Survey of Synthetic Data Augmentation Methods in Computer Vision. *arXiv preprint arXiv:2403.10075*, 2024.
3. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. *In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 23-30).
4. Jackson, D., Gokhale, V., & Wyatt, J. L. (2019). Quantifying the Use of Domain Randomization for Object Localization. *arXiv preprint arXiv:1910.03438*.
5. Csurka, G. (2017). Domain Adaptation for Visual Applications: A Comprehensive Survey. *arXiv preprint arXiv:1702.05374*.
6. Wang, M., & Deng, W. (2018). Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 312, 135-153.
7. Hinterstoisser, S., Pauly, O., Heibel, H., Marek, M., & Bokeloh, M. (2019). An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Instance Detection. *In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9779-9789).
8. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. *Advances in neural information processing systems*, 27.
9. Borkman, S., et al. (2021). Unity Perception: Generate Synthetic Data for Computer Vision. *arXiv preprint arXiv:2107.04259*.
10. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 779-788).
11. Koirala, A., et al. (2021). Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning. *Journal of Intelligent & Robotic Systems*, 103(4), 67.
12. Truong, J., Chernova, S., & Batra, D. (2021). Bi-directional Domain Adaptation for Sim2Real Transfer of Embodied Navigation Agents. *IEEE Robotics and Automation Letters (RA-L)*, 6(2), 2634–2641.
13. Kadian, A., Chhabra, T., Gupta, K., & Kumar, S. (2023). A Survey of Sim-to-Real Methods in RL: Progress, Prospects, and Challenges with Foundation Models. *arXiv preprint arXiv:2302.09337*.
14. Hashemifar, S., et al. (2024). Recent Advances in Deep Learning for Protein-Protein Interaction: A Review. *International Journal of Molecular Sciences*, 25(11), 5949.
15. Awais, M., et al. (2023). Don't freeze: Finetune encoders for better Self-Supervised HAR. *In Proceedings of the 2023 ACM International Symposium on Wearable Computers*.
16. Finlayson, G. D., et al. (2023). Impact of Exposure and Illumination on Texture Classification Based on Raw Spectral Filter Array Images. *Sensors*, 23(12), 5649.
17. Chung, E., et al. (2023). Inclusive Portrait Lighting Estimation Model Leveraging Graphic-Based Synthetic Data. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.
18. Nikolenko, S. I. (2021). *Synthetic Data for Deep Learning*. Springer Nature.
19. Picard, R. W. (2021). The Reproducibility Crisis in ML/AI: An Overview. *IEEE Open Journal of Signal Processing*, 2, 407–414.
20. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.
21. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, J. (2020). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07), 12993-13000.
22. Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv preprint arXiv:1608.03983*.

23. Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., ... & He, K. (2017). Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv preprint arXiv:1706.02677*.
24. Uzlov, D., Strukov, V., Hudilin, V., & Vlasov, O. (2023). Problematic issues of machine learning technology in law enforcement. Computer Science and Cybersecurity, 2, 6-15. URL:https://doi.org/10.26565/2519-2310-2023-2- 01

Korshenko Vladyslav *PhD student at the Department of Cybersecurity of Information Systems, Networks and Technologies, senior lecturer of Department of Mathematical Modeling and Data Analysis, V. N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv, Ukraine, 61077*
e-mail: v.korshenko@karazin.ua
https://orcid.org/0000-0003-2197-072X

Uzlov Dmytro *Candidate of Technical Sciences, Director of Educational and Scientific Institute of Computer Science and Artificial Intelligence, V.N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv, Ukraine, 61077*
e-mail: dmytro.uzlov@karazin.ua*;*
https://orcid.org/0000-0003-3308-424X

Assessment of the impact of photorealistic textures on the accuracy of computer vision models using synthetic datasets

Relevance. The current development of computer vision faces the problem of high cost and labor intensity of collecting real annotated data. The use of synthetic data generated in graphics engines is an effective alternative, but the main obstacle remains the "domain gap," which reduces the accuracy of models on real images.
The goal of this work is to quantitatively assess the impact of the photorealistic texture of the target object on the detection efficiency of YOLO models when transitioning from simulation to reality (Sim2Real).
The research methodology is based on a controlled experiment in the Unity environment, where two identical synthetic datasets were generated, differing only in the type of 3D model texture: highly detailed photorealistic ("Textured") and monochrome white ("White"). The models were trained based on the YOLOv11s architecture using a transfer learning strategy and a two-step fine-tuning process. The results were validated on an independent set of exclusively real photographs.
Results. Both models, trained on two datasets ("Textured" and "White"), achieved almost identical accuracy on synthetic validation data (mAP@0.5 ≈ 0.995). However, on real photos, the "Textured" model demonstrated 11.6 times higher mAP@0.5 compared to the "White" model. The recall for the textured model was 10.3 times higher than for the model that relied solely on geometric shape.
Conclusions. Photorealistic texture is a critical factor for successful Sim2Real transfer. It ensures the formation of universal low-level features in the early layers of the neural network, which are necessary for recognizing objects in a real environment. High-quality texturing of 3D assets should be considered a strategic priority rather than an auxiliary stage of visualization.
***Keywords**: synthetic data, computer vision, object detection, domain gap, model robustness, domain shift resilience.*

Надійшла у першій редакції 11.01.2026; рекомендована до друку 12.03.2026; опублікована 03.04.2026.
The first version has been received on 11.01.2026; was recommended for printing on 12.03.2026; published 03.04.2026.