https://doi.org/10.31891/2307-5732-2025-359-94
УДК 004.8:004.932

ДАРМОГРАЙ СВЯТОСЛАВ
Національний Університет Львівська Політехніка
https://orcid.org/0000-0003-3617-0215
e-mail: darmohrays@gmail.com

НАВЧАННЯ ПІД ЧАС ТЕСТУВАННЯ ДЛЯ МОНОКУЛЯРНОЇ ОЦІНКИ ГЛИБИНИ

Попри величезний технологічний прогрес методів глибокого навчання протягом останніх років, для складних завдань, як монокулярне оцінювання глибини, ці підходи все ще не можна надійно застосовувати для задач у справжньому світі. Ця проблема зумовлена труднощами отримання високоякісних еталонних даних, необхідних для сучасних моделей глибинного навчання. Щоб вирішити проблему малої кількості реальних якісно розмічених даних, найуспішніші сучасні моделі використовують штучно згенеровані набори даних, що спричиняє зсуви розподілу даних. Останнім часом з'явився сплеск досліджень методів навчання під час тестування (test-time training), які можуть успішно застосовуватись для проблемм адаптації до нових даних. Метою цієї роботи є оцінити, чи здатні методи навчання під час тестування покращити результати попередньо натренованої моделі монокулярного оцінювання глибини. Результати дослідження свідчать, що поліпшення справді можливі, але за рахунок значного збільшення часу інференсу. До того ж , хоч середні показники зростають, залишається невизначеність щодо того, чи буде досягнуто покращення на окремих зображеннях. В окремих випадках частка правильно розміщених точок може зрости з 0,1 до 0,9, але на інших зразках може знизитися з 0,9 до 0,5.

Ключові слова: тренування під час тестування, монокулярна оцінка глибини, глибоке навчання

DARMOHRAI SVIATOSLAV
Lviv Polytechnic National University

TEST-TIME TRAINING FOR MONOCULAR DEPTH ESTIMATION

Despite the significant technological advancements in deep learning models over the past few years, they still cannot be reliably applied to challenging tasks, such as monocular depth estimation, on real-world data. One of the key bottlenecks arises from difficulties in obtaining high-quality ground truth data, which is essential for current deep learning models. Supervised objectives are only as reliable as their targets: imprecise or biased labels distort the loss landscape, encouraging models to overfit spurious cues and hurting generalization. For depth estimation in particular, the data comes from LiDAR/RGB-D projections, MVS reconstructions, ordinal labels, or synthetic renderings—each with characteristic failures: LiDAR is sparse and prone to motion/occlusion ghosting; RGB-D has holes and edge artifacts on shiny/transparent or distant surfaces;. To alleviate the issue of a lack of real-world, quality-labeled data, the most successful contemporary models utilize artificially generated datasets, which introduce distribution shifts. Recently, there was a surge in research on test-time training methods, which can be successfully applied to the issue of adaptation to new data. In test-time training (TTT), the model updates a subset of its parameters (e.g., normalization statistics or lightweight heads) at inference using self-supervised auxiliary losses that do not require labels, such as consistency or reconstruction objectives. By aligning internal representations to the target distribution on-the-fly, TTT can reduce distribution shift and improve robustness without additional annotated data. This work sets as a goal to evaluate whether Time-Test Training methods could improve the results of the already pretrained monocular depth estimation model. Results suggest that improvements are indeed possible, but with a significant increase in inference time. Even though results improve on aggregate, there is uncertainty about whether the improvement will be achieved on some specific sample. In separate cases, the percentage of correctly placed points can improve from 0.1 to 0.9; however, on other samples, the score can go down from 0.9 to 0.5.

Keywords: *Test-Time Training, Monocular Depth Estimation, Deep Learning*

Стаття надійшла до редакції / Received 22.10.2025
Прийнята до друку / Accepted 15.11.2025

Постановка проблеми

Поєднання Deep Learning методів з великими розміченими масивами даних дозволило отримати моделі які досягають неймовірних результатів у багатьох задачах комп'ютерного зору. Однією з таких задач є визначення глибини сцени з одного зображення. Ця задача є неймовірно складною, оскільки єдине 2D зображення не несе у собі достатньої інформації щоб відтворити сцену у 3D: неможливо дізнатись глибину сцени - чи це малі об'єкти близько, чи великі далеко; проблема неоднозначність барельєфа[1] - різні 3D фігури можуть видавати майже ідентичні зображення за різного освітлення.

Незважаючи на перелічені складнощі, глибокі нейронні мережі показують хороші результати на популярних бенчмарках, проте при детальному огляді можна побачити певну кількість проблем. Через проблему неоднозначності масштабу сцени, більшість моделей передбачають відносну глибину. Отримання великих наборів даних з реальними значеннями глибини сцени є дорогим та складним. Заміни реальних даних на згенеровані за допомогою рендеренгу штучних 3Д сцен може допомогти збільшити кількість даних, з високоякісною розміткою, проте створює проблему зміщення розподілу даних. Ця проблема є характерною для всіх моделей глибокого навчання, вона проявляється у тому, що якість передбачень моделей погіршиться, при передбаченнях на даних, які були мало репрезентовані у тренувальній вибірці. Усі ці проблеми призводять до того, що поточні моделі для монокулярної оцінки глибини не є достатньо надійними для застосування у реальному світі у складних задачах.

Враховуючи усі обмеження методів глибокого навчання, запропоновано методи навчання під час тестування, які дозволяють адаптувати ваги моделі, до поточного прикладу. Попередньо, ці моделі були

застосовані для вирішення проблеми зміщення розподілу даних у задачах класифікації[2] та сегментації[3] зображень і ціною довшого часу передбачення отримано кращі результати.

З огляду на проблеми, описані вище, метою роботи є застосування Test-Time Training методів в задачі монокулярної оцінки глибини.

Об'єктом дослідження є цифрові зображення в задачах монокулярної оцінки глибини.

Предметом дослідження є методи визначення масштабу сцени.

Для досягнення мети, треба було вирішити такі завдання:

1. Провести аналіз існуючих методів
2. Спроектувати та імплементувати систему, для покращення передбачень уже натренованих моделей з використанням Test-Time Training методів
3. Аналіз результатів

Аналіз досліджень та публікацій

Першою важливою моделлю для монокулярного передбачення глибини можна виділити MiDaS[4]. Вони досягли найкращих результатів за допомогою комбінування різних наборів даних, у який могло відрізнятись представлення правдивих даних: щільне попіксельне представлення для стерео наборів даних, проріджені точки для даних з лідару.

Аби покращити результати попередніх моделей, автори DepthAnything[5] сфокусувалися на якості даних. Оскільки на реальних даних складно дістати ідеальну глибину сцени для кожного пікселя, було запропоновано тренувати велику модель лише на ідеальних даних згенерованих за допомогою 3Д рушіїв. Проте, тренування лише на штучних даних, не дозволило отримати хороші результати на реальних даних, тому ця модель використовувалась як учитель для генерації псевдоміток для реальних даних, на яких навчались менші моделі-студенти.

У вищезгаданих дослідженнях використовувались дані, які не завжди мали інформації про метричну глибину сцени у випадку MiDaS, і взагалі не мали її у випадку DepthAnything v2, тож моделі передбачали глибину в такому просторі, для якого є афінне перетворення, дізнавшись яке, можна було б отримати реальні метричні дані. У дослідженні MoGe2,[6] було запропоновано створити окрему голову, яка вивчала б перетворення оригінального афінно-інваріантного передбачення глибини у метричний масштаб. Окрім цього, теж була проведена робота над даними: модель натренована лише на штучних даних, використовувалась для доповнення та фільтрації реальних даних, на яких тренувались основні моделі.

Деякі дослідження напряму розглядають проблему зсуву розподілу даних у монокулярній оцінці глибини, навчаючи моделі бути стійкими до несприятливих умов, таких як погана погода та слабке освітлення. DepthAnything-AC[7] дотреновує існуючу потужну модель за допомогою фреймворку узгодженої регуляризації. На противагу цьому, Robust-Depth[8] пропонує нову методологію самоконтрольованого навчання з нуля, яка використовує двонаправлений псевдоконтроль між парами аугментованих та неаугментованих зображень. Однак, ці підходи теж вимагають високоякісних розмічених даних.

Для вирішення проблеми зміщення розподілу даних та онлайн адапдації до них, нова активна галузь досліджень це Test-Time Training підходи. З їх допомогою вдалось покращити якість класифікації зображень[2], сегментації[3]. Окрім цього було запропоновано архітектуру, яку можна використати для будь-якого завдання[9].

Виклад основного матеріалу

У класичній задачі керованого навчання, ми припускаємо що маємо доступ до пар даних (x_i, y_i) на яких проводиться навчання з функцією витрат, яка залежить від обох x_i та y_i.

$$\min_{\Theta} \frac{1}{n} \sum_{i=1}^{n} l_m(x_i, y_i, \Theta)$$

Після того, як модель буде натренована, ваги моделі будуть зафіксованими, і не будуть мінятись при її використанні, що виключає можливість адаптації до нових даних.

У парадигмі Test-Time Training під час тестування на вході доступний лише x_i (за потреби — його аугментовані варіанти $A(x_i)$. Оновлення параметрів відбувається кілька ітерацій, використовуючи лише вхідні дані у задачі самокерованого навчання, яка має допомогти моделі вивчити кращі ознаки, які можуть бути використані для головної задачі.

Головна задача у цьому дослідженні це монокулярне передбачення глибини сцени. Щоб уникнути другого етапу перетренування моделі і щоб сфокусуватись лише на дослідженні застосування Test-Time Training підходу, дослідження були проведені на натренованій моделі без жодних змін до її архітектури. Для цього було вирішено застосувати підхід Учитель-Студент, де передбачення учителя використовуються як ціль для студента. Цей підхід дозволяє створювати різні варіанти вхідного зображення, і ціллю студента буде зробити усі передбачення узгодженими між собою.

```
y_teacher = TeacherPredict(data_sample)
for iteration in range(1, STEPS):
    x_gen = GenerateSamples(data_sample, y_teacher)
    y_student = StudentPredict(x_gen)
    StudentUpdate(y_student, y_teacher)
    if (iteration % TEACHER_UPDATE_N) == 0:
        EMAUpdateTeacher(teacher, student)
        y_teacher = TeacherPredict(data_sample)
final_output = TeacherPredict(data_sample)
```

Рис 1. Псевдокод алгоритму

Цей підхід дозволяє моделі уникати перенавчанню на своїх помилках, оскільки модель не буде оптимізованою напряму: ваги учителя заморожені, і до них не повертатиметься градієнт.

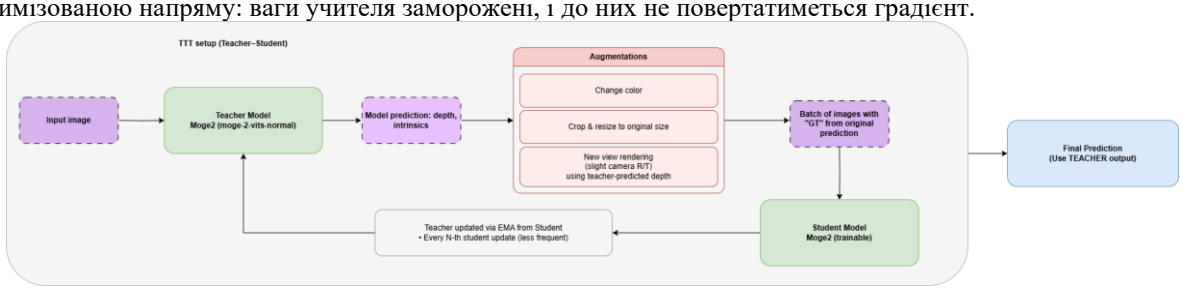

Рис 2. Схематичне зображення алгоритму

До створення варіантів вхідного зображення теж потрібно підійти правильно, прості зміни кольору можуть не надати достатньо інформації для моделі, щоб покращити передбачення глибин, тому було вирішено додати обов'язкові геометричні аугментації: випадкове обрізання зображення, випадкове віддзеркалення.

Враховуючи особливість задачі - те, що ми маємо передбачення 3Д глибини сцени, це також можна використати для генерування нових варіантів вхідних даних: 3Д сцена може буде зарендерена в 3Д рушії з відмінною позицією камери. Проте виникає проблема, оскільки можуть з'являтись пропущені місця за об'єктами, для таких випадків генерується маска, яка показує лише пікселі, які належать об'єктам в сцені. Проте використання таких аугментацій ще більше зменшує час однієї ітерації.

Рис 3. Приклади варіації вхідного зображення. Перші два стовпці - застосовано лише віддзеркалення та обрізання. Два останні стовпці: рендер з іншою позицією камери. На всіх прикладах застосовувались випадкові аументації кольору

Опис даних

Якість моделі було обраховано на тестовому наборі даних, використаному авторами Moge-2, що включає такі набори даних: NYUv2[10], KITTI[11], ETH3D[12], iBims-1[13], GSO[14], Sintel[15], DDAD[16], DIODE[17], Spring[18], Hammer[19]. Щоб пришвидшити експериментальний цикл, було відібрано приклади зі значенням метрики points_scale_invariant_delta1 на базовій моделі меншими ніж 0. Тож фінальна вибірка, на які було проведено всі експерименти складається з 50 прикладів, і саме до них застосовувався метод TTT.

Для всіх експериментів було використано модель moge-2-vits-normal. Всі експерименти були проведені на Nvidia RTX 3080 12 GB відеокарті. Щоб модель помістилась повністю у відеопам'ять для всіх експериментів був встановлений розмір батчу 2. Для того, щоб вирішити проблему шуму від малого розміру батчу, було застосовано акумуляцію градієнтів протягом 32-ох ітерацій. Для усіх деталей, перегляньте імплементацію в https://github.com/Darmohrays/MoGe_TT.

Опис функції втрат та метрик

Функцію втрат було перевикористано з Moge:

$$L_g = \sum_{i \in M} \frac{1}{z_i} ||s \times p_{i,p} + t - p_{i,g}||,$$

Де s та t - оптимальні параметри зміщення, знайдені за допомогою ROE алгоритму запропонованого авторами; M - маска валідних пікселів; $\frac{1}{z_i}$- зважує помилку з врахуванням відстані до неї від камери.

Для оцінки результатів було використано метрики з репозиторію Moge, де *rel* вимірює середню відносну похибку, а δ_1 - частку пікселів/точок із «достатньою точністю», це означає, що прогноз лежить у певній межі від правдивого значення. Аби мати краще розуміння передбачень моделі, до її передбачень застосовується вирівнювання, яке розділяє абсолютну метричну точність і якість структури. Параметри для вирівнювання обраховуються за допомогою методу найменших квадратів між передбаченням і правдивим значенням.

Загальні метрики:

Глибина *rel:* $\frac{1}{N}\sum \frac{|d_p - d_g|}{d_g - \varepsilon}$

Глибина δ_1: $\frac{1}{N}\sum(max(\frac{d_g}{d_p}, \frac{d_p}{d_g}) < \rightleftarrows 1.25 \rightleftarrows)$

3Д точки *rel:* $\frac{1}{N}\sum \frac{|||p_p - p_g||}{||p_g|| + \varepsilon}$

3Д точки δ_1: $\frac{1}{N}\sum(||p_p - p_g|| < 0.25 \times min(||p_g||, ||p_p||))$

Вирівнювання

depth_metric: без вирівнювання

depth_scale_invariant: $d_p^{'} = sd_p$

depth_affine_invariant: $d_p^{'} = sd_p + b$

points_metric: $p_p^{'} = p_p + t$ (лише зсув)

points_scale_invariant: $p_p^{'} = sp_p$ (лише масштаб)

points_affine_invariant: $p_p^{'} = sp_p + t$ (масштаб та зсув)

Опис експериментів

Загалом було проведено 11 експеременті для фінальної аблації результатів, які представлені у таблиці 1. Першим було вирішено перевірити, яку частину ваг слід заморозити під час тренування. Модель сладається з окремих модулів: енкодер, шия, та окремі голови. У експерименті freeze_encoder було заморожено енкодер, розморожено шию та голови, та навпаку у експерименті unfreeze_encoder. Порівнячи результати, можна спостерігати, що замороження лише енкодера не дає великого приросту по метриках, порівняно з розмороженим, тож наступні експеременти було вирішено проводити з розмороженим енкодером. Як базовий алгоритм оптимізації використовувався AdamW, проте було проведено декілька експериментів, щоб визначити оптимальний коефіцієнт швидкості навчання(у експериментах lower_lr =1e-5, higher_lr = 1e-3, стандартний = 5e-4). Окрім цього, також було перевірено SGD алгоритм, проте він не надав приросту в якості, і з великим значенням навчального кроку лише погіршив результати. Також важливим було перевірити, наскільки довжина тренування впливає на якість результатів, для чого були проведені експерименти shorter - 256 ітерацій, longer - 1024, стандартний - 512. Ці експерименти показали, що навіть з меншою кількістю ітерацій можна отримати кращі результати, і експеримент з меншою кількістю ітерацій один з двох, якому вдалось покращити результати на всіх метриках, хоч і з меншим приростом ніж у деяких інших експериментах. Натомість довше тренування, не тільки вдвічі збільшує час отримання передбачень, але призводить до розходження моделі та отримання гірших результатів.

Таблиця 1

Результати експериментів

Experiment	Depth						Points						Time, sec
	metric		scale invariant		affine invariant		metric		scale invariant		affine invariant		
	Rel↓	δ1↑	Rel↓	δ1↑	Rel↓	δ1↑	Rel↓	δ1↑	Rel↓	δ1↑	Rel↓	δ1↑	
Original	0,42	0,35	0,44	0,34	0,37	0,45	0,35	0,39	0,50	0,18	0,44	0,32	0,039
freeze_encoder	0,42	0,35	0,37	0,34	0,21	0,67	0,35	0,39	0,38	0,32	0,31	0,43	238
unfreeze_encoder	0,49	0,44	0,31	0,49	0,22	0,66	0,37	0,53	0,33	0,48	0,27	0,58	248
unfreeze_encoder_normal_loss	0,47	0,41	0,33	0,46	0,23	0,64	0,38	0,51	0,35	0,45	0,28	0,54	248
unfreeze_encoderSGD	0,47	0,37	0,32	0,48	0,22	0,66	0,36	0,51	0,34	0,44	0,27	0,56	248
unfreeze_encoderSGD_lower_lr	0,42	0,36	0,36	0,34	0,21	0,67	0,35	0,40	0,38	0,32	0,30	0,44	248
unfreeze_encoderSGD_higher_lr	2,84	0,10	0,57	0,19	0,52	0,26	1,40	0,12	0,62	0,08	0,57	0,18	248
unfreeze_encoder_lower_lr	0,39	0,36	0,36	0,34	0,19	0,70	0,32	0,42	0,38	0,32	0,29	0,45	248
unfreeze_encoder_higher_lr	0,39	0,36	0,36	0,34	0,19	0,70	0,33	0,42	0,38	0,32	0,29	0,45	248
unfreeze_encoder_shorter	0,41	0,41	0,34	0,42	0,21	0,65	0,34	0,44	0,35	0,38	0,28	0,52	117
unfreeze_encoder_longer	0,64	0,31	0,34	0,45	0,26	0,61	0,50	0,41	0,38	0,36	0,31	0,51	461

Незважаючи на те, що середні результати покращились, ще однією важливою метрикою для застосування цього підходу є його надійність, для цього було вирішено переглянути розподіл для кожного окремого зображення(рис 4). По рисунках, ми можемо бачити, що приріст в позитивну сторону є в рази більшим, проте в половині випадків, результати, хоч і не значно, але погіршуються.

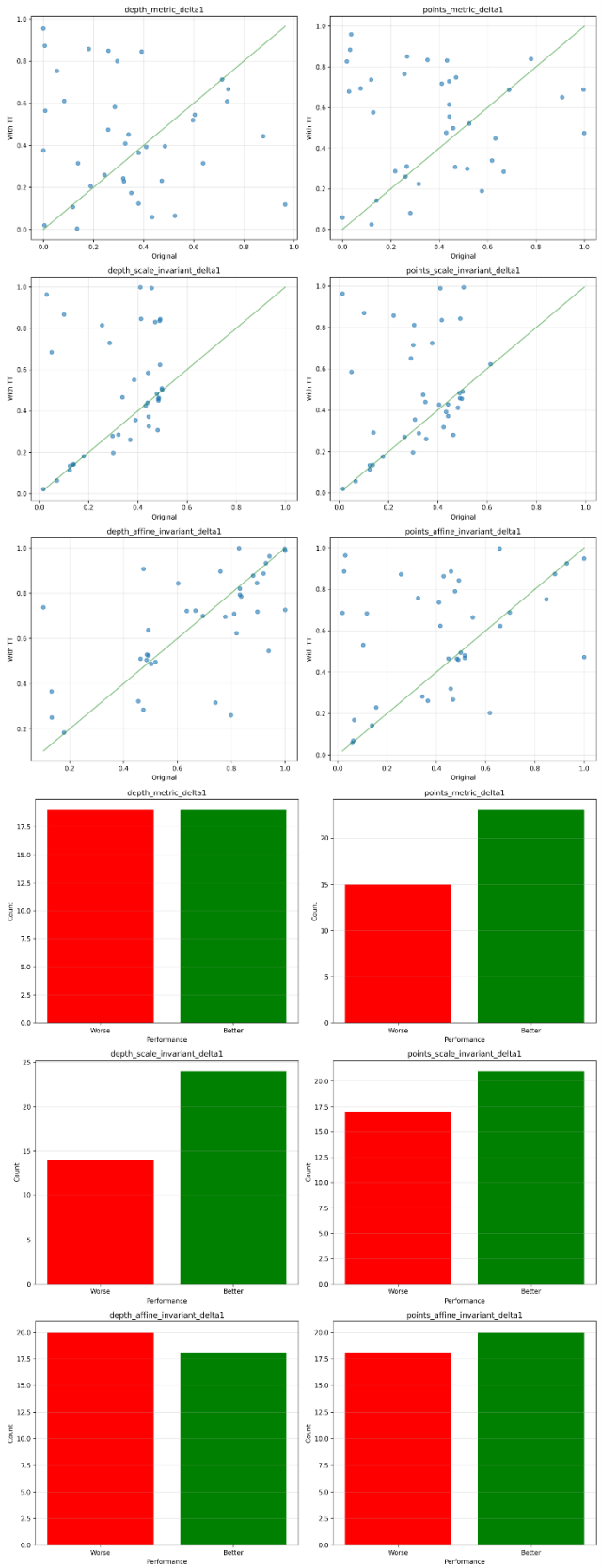

Рис 4. Зліва-порівняння передбачень на окремих тестових прикладах справа - рисунок зображує у скількох випадках вдалось отримати результати кращі порівняно з попередньою моделлю, у скількох гірші

Висновки

При застосуванні запропонованої архітектури, отримано покращення ефективності передбачення за всіма усередненими на датасеті метриками. Проте, при детальному розгляді окремих прикладів даних, можна помітити, що для окремих зображень, його застосування може і погіршити результат на окремих прикладах даних.

Також обмеженням методу є вимоги по ресурсах: час отримання передбачення збільшується з 0.039 до 248 секунд, та в рази збільшуються вимоги до відеопам'яті системи - потрібно не менше 12 гігабайт відеопам'яті. Це не дозволяє використовувати розроблений підхід для аналізу даних в реальному часі, чи на довільному пристрої користувача.

У метричному просторі покращення були найменш надійними - лише в 3 експериментах вдалось покращити результати за обраними метриками . Проте, як показали попередні дослідження, покращення метричних передбачень також є можливим, за наявності метричних референсів на зображенні, які дадуть реальну відстань від сенсору, до деяких точок на зображенні.

Для майбутніх досліджень залишається відкритою проблема надійності TTT методу, оскільки він може як і суттєво покращити, так і суттєво погіршити передбачення на окремому зображенні.

Література

1. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L. The Bas-Relief Ambiguity. International Journal of Computer Vision 35, 33–44. 1999. DOI: 10.1023/A:1008154927611

2. Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. July 2020. [Електронний ресурс]. arXiv:1909.13231. DOI: 10.48550/arXiv.1909.13231

3. Yossi Gandelsman, Yu Sun, Xinlei Chen, Alexei A. Efros. Test-Time Training with Masked Autoencoders. September 2022. [Електронний ресурс]. arXiv:2209.07522. DOI: 10.48550/arXiv.2209.07522

4. René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. August 2020. [Електронний ресурс]. arXiv:1907.01341. DOI: 10.48550/arXiv.1907.01341

5. Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, Hengshuang Zhao. Depth Anything V2. October 2024. [Електронний ресурс]. arXiv:2406.09414. DOI: 10.48550/arXiv.2406.09414

6. Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details; July 2025. [Електронний ресурс]. arXiv:2507.02546. DOI 10.48550/arXiv.2507.02546

7. Boyuan Sun, Modi Jin, Bowen Yin, Qibin Hou. Depth Anything at Any Condition. July 2025. [Електронний ресурс]. arXiv:2507.01634. DOI: 10.48550/arXiv.2507.01634

8. Johannes Kopf, Xuejian Rong, Jia-Bin Huang. Robust Consistent Video Depth Estimation. June 2021. [Електронний ресурс]. arXiv:2012.05901. DOI: 10.48550/arXiv.2012.05901

9. Qin Wang, Olga Fink, Luc Van Gool, Dengxin Dai. Continual Test-Time Domain Adaptation. March 2022. [Електронний ресурс]. arXiv:2203.13591. DOI: 10.48550/arXiv.2203.13591

10. Silberman, N., Hoiem, D., Kohli, P., Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7576. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-33715-4_54

11. Menze. M., Geiger A. Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR). 2015. DOI: 2010.1109/CVPR.2015.7298925

12. Schops T., Schonberger J. L., Galliani S. et al. A Multi-view Stereo Benchmark with High-Resolution Images and Multi-camera Videos. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (Honolulu, HI, 07.2017). Honolulu, HI : IEEE, 2017. DOI:10.1109/CVPR.2017.272. P. 2538–2547.

13. Koch T., Liebel L., Fraundorfer F. et al. Evaluation of CNN-based Single-Image Depth Estimation Methods. May 2018. [Електронний ресурс]. DOI:10.48550/arXiv.1805.01328.

14. Downs L., Francis A., Koenig N. et al. Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items. April 2022. [Електронний ресурс] DOI:10.48550/arXiv.2204.11918.

15. Kopf J., Rong X., Huang J.-B. Robust Consistent Video Depth Estimation. June 2021. [Електронний ресурс]. DOI:10.48550/ARXIV.2012.05901.

16. Guizilini V., Ambrus R., Pillai S. 3D. Packing for Self-Supervised Monocular Depth Estimation. March 2020. [Електронний ресурс]. DOI:10.48550/arXiv.1905.02693.

17. Vasiljevic I., Kolkin N., Zhang S. et al. DIODE: A Dense Indoor and Outdoor DEpth Dataset. August 2019. [Електронний ресурс]. DOI:10.48550/arXiv.1908.00463.

18. Mehl L., Schmalfuss J., Jahedi A. et al. Spring: A High-Resolution High-Detail Dataset and Benchmark for Scene Flow, Optical Flow and Stereo. March 2023. [Електронний ресурс]. DOI:10.48550/arXiv.2303.01943.

References

1. Belhumeur, P.N., Kriegman, D.J., Yuille, A.L. The Bas-Relief Ambiguity. International Journal of Computer Vision 35, 33–44. 1999. DOI: 10.1023/A:1008154927611

2. Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, Moritz Hardt. Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. July 2020. [Elektronnyi resurs]. arXiv:1909.13231. DOI: 10.48550/arXiv.1909.13231

3. Yossi Gandelsman, Yu Sun, Xinlei Chen, Alexei A. Efros. Test-Time Training with Masked Autoencoders. September 2022. [Elektronnyi resurs]. arXiv:2209.07522. DOI: 10.48550/arXiv.2209.07522

4. René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. August 2020. [Elektronnyi resurs]. arXiv:1907.01341. DOI: 10.48550/arXiv.1907.01341

5. Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, Hengshuang Zhao. Depth Anything V2. October 2024. [Elektronnyi resurs]. arXiv:2406.09414. DOI: 10.48550/arXiv.2406.09414

6. Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, Jiaolong Yang. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details; July 2025. [Elektronnyi resurs]. arXiv:2507.02546. DOI 10.48550/arXiv.2507.02546

7. Boyuan Sun, Modi Jin, Bowen Yin, Qibin Hou. Depth Anything at Any Condition. July 2025. [Elektronnyi resurs]. arXiv:2507.01634. DOI: 10.48550/arXiv.2507.01634

8. Johannes Kopf, Xuejian Rong, Jia-Bin Huang. Robust Consistent Video Depth Estimation. June 2021. [Elektronnyi resurs]. arXiv:2012.05901. DOI: 10.48550/arXiv.2012.05901

9. Qin Wang, Olga Fink, Luc Van Gool, Dengxin Dai. Continual Test-Time Domain Adaptation. March 2022. [Elektronnyi resurs]. arXiv:2203.13591. DOI: 10.48550/arXiv.2203.13591

10. Silberman, N., Hoiem, D., Kohli, P., Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. Computer Vision – ECCV 2012. ECCV 2012. Lecture Notes in Computer Science, vol 7576. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-33715-4_54

11. Menze. M., Geiger A. Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR). 2015. DOI: 2010.1109/CVPR.2015.7298925

12. Schops T., Schonberger J. L., Galliani S. et al. A Multi-view Stereo Benchmark with High-Resolution Images and Multi-camera Videos. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (Honolulu, HI, 07.2017). Honolulu, HI : IEEE, 2017. DOI:10.1109/CVPR.2017.272. P. 2538–2547.

13. Koch T., Liebel L., Fraundorfer F. et al. Evaluation of CNN-based Single-Image Depth Estimation Methods. May 2018. [Elektronnyi resurs]. DOI:10.48550/arXiv.1805.01328.

14. Downs L., Francis A., Koenig N. et al. Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items. April 2022. [Elektronnyi resurs] DOI:10.48550/arXiv.2204.11918.

15. Kopf J., Rong X., Huang J.-B. Robust Consistent Video Depth Estimation. June 2021. [Elektronnyi resurs]. DOI:10.48550/ARXIV.2012.05901.

16. Guizilini V., Ambrus R., Pillai S. 3D. Packing for Self-Supervised Monocular Depth Estimation. March 2020. [Elektronnyi resurs]. DOI:10.48550/arXiv.1905.02693.

17. Vasiljevic I., Kolkin N., Zhang S. et al. DIODE: A Dense Indoor and Outdoor DEpth Dataset. August 2019. [Elektronnyi resurs]. DOI:10.48550/arXiv.1908.00463.

18. Mehl L., Schmalfuss J., Jahedi A. et al. Spring: A High-Resolution High-Detail Dataset and Benchmark for Scene Flow, Optical Flow and Stereo. March 2023. [Elektronnyi resurs]. DOI:10.48550/arXiv.2303.01943.