Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна наукова праця
на правах рукопису

КОВТУНЕНКО АНДРІЙ РОМАНОВИЧ

УДК 004.83

ДИСЕРТАЦІЯ

НЕЙРОМЕРЕЖЕВІ МЕТОДИ МУЛЬТИМОДАЛЬНОЇ ПРОСТОРОВО-СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ ЗА НЕЧІТКИМ ЗАПИТОМ

Спеціальність: 122 – Комп’ютерні науки
Галузь знань: 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.
_____ А. Р. Ковтуненко

Науковий керівник:
Машталір Сергій Володимирович
доктор технічних наук, професор

Харків-2026

АНОТАЦІЯ

Ковтуненко А.Р. Нейромережеві методи мультимодальної просторово-семантичної сегментації за нечітким запитом – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки (12 – Інформаційні технології). Харківський національний університет радіоелектроніки, Міністерство освіти і науки України, Харків, 2026.

Дисертаційну роботу присвячено розробці методів та моделей мультимодальної сегментації зображень за текстовим запитом довільної форми з можливістю виділення окремих екземплярів об'єктів на основі архітектурних модифікацій нейронних мереж та удосконалених алгоритмів постобробки.

Дисертаційна робота була виконана за підтримки проєкту Horizon Europe «Підтримка європейської дослідницько-інноваційної діяльності через співпрацю стейкхолдерів та інституційну реформу» («Supporting European R&I Through stakeholder collaboration and institutional reform», INITIATE, HORIZON-WIDERA-2023-ACCESS-03, номер проєкту 101136775).

Стрімке зростання обсягів мультимедійної інформації висуває нові вимоги до методів її автоматизованої обробки та аналізу. Серед ключових напрямів такої обробки особливе місце посідає сегментація зображень як фундаментальна задача комп'ютерного зору, що полягає у розбитті зображення на семантично однорідні регіони. Класичні методи сегментації, що базуються на фіксованому наборі класів (closed-set), демонструють високу ефективність на стандартних еталонних наборах даних, проте їх застосування в реальних умовах суттєво обмежене архітектурною залежністю від попередньо визначених категорій об'єктів.

У практичних сценаріях об'єкти характеризуються варіативними властивостями, можуть описуватися через атрибути, дії або часткову

належність до категорій. Крім того, взаємодія об’єктів породжує нові контекстні ознаки, які принципово неможливо передбачити на етапі формування навчальної вибірки. Це зумовило необхідність переходу до парадигми open-set методів, де множина класів не є фіксованою і може визначатися текстовим запитом довільної форми (open-vocabulary). Такий підхід дозволяє користувачу формулювати запити природною мовою, описуючи шукані об’єкти через їх візуальні характеристики, просторове розташування або функціональне призначення.

Революційним кроком у цьому напрямку стала поява моделі CLIP (Contrastive Language-Image Pretraining), що забезпечила можливість зіставлення візуальних та текстових представлень у єдиному семантичному просторі. Завдяки контрастивному навчанню на масштабному корпусі пар «зображення-текст» модель набула здатності порівнювати візуальний контент із природномовними описами без прив’язки до конкретного набору класів. Подальший розвиток мультимодальних архітектур, зокрема SAM (Segment Anything Model) та CLIPSeg, продемонстрував потенціал підходів, у яких запит на сегментацію формулюється природною мовою.

Однак існуючі методи сегментації екземплярів за текстовим запитом переважно реалізовані як двоетапні процедури, де спочатку виконується детекція об’єктів, а потім їх сегментація. Типовим представником такого підходу є Grounded-SAM, що послідовно застосовує детектор GroundingDINO та сегментатор SAM. Така архітектура призводить до значних обчислювальних витрат, збільшує час роботи системи та не дозволяє повною мірою використати переваги мультимодального навчання. Крім того, помилки детекції на першому етапі неминуче впливають на якість фінальної сегментації.

Окремою науковою проблемою залишається вибір функції втрат для навчання сегментаційних нейронних мереж. Широко застосовувані функції, такі як Dice, попри свою практичну ефективність, не мають строгого математичного обґрунтування як метрики у класичному розумінні теорії

метричних просторів. Зокрема, ці функції не задовольняють нерівність трикутника, що обмежує можливості їх використання в ієрархічних підходах до сегментації та алгоритмах ефективного пошуку у просторі розбиттів. Нерівність трикутника є принципово важливою властивістю, оскільки дозволяє швидко відсіювати завідомо несхожі варіанти при порівнянні сегментацій

Таким чином, актуальним науковим завданням є розробка моделей, методів та алгоритмів мультимодальної просторово-семантичної сегментації за нечітким запитом на основі модифікації архітектур глибокого навчання для підвищення ефективності виділення екземплярів об’єктів в умовах open-vocabulary парадигми.

Метою дисертаційної роботи є розробка моделей, методів та алгоритмів мультимодальної просторово-семантичної сегментації за нечітким запитом на основі модифікації архітектур глибокого навчання для підвищення ефективності виділення екземплярів об’єктів у режимі open-vocabulary.

Для досягнення мети дослідження в дисертаційній роботі вирішуються такі завдання:

1) виконати аналіз проблем та підходів до вирішення задачі сегментації зображень за текстовим запитом в умовах open-set парадигми;

2) розробити модифіковану архітектуру моделі сегментації екземплярів на основі CLIPSeg для прогнозування центрів об’єктів та відстаней до меж;

3) розробити метод постобробки результатів висхідної сегментації для розділення екземплярів об’єктів;

4) розробити метричну функцію втрат для порівняння сегментацій на основі теорії розбиттів;

5) провести експериментальну перевірку розроблених методів та моделей на стандартних наборах даних.

Об’єктом дослідження є процес сегментації зображень за текстовим запитом довільної форми.

Предметом дослідження є моделі, методи та алгоритми мультимодальної просторово-семантичної сегментації на основі глибокого навчання.

Результати дисертаційного дослідження базуються на використанні: теорії глибокого навчання та згорткових нейронних мереж – при розробці архітектури InstanceCLIPSeg з декодерами для прогнозування центрів та відстаней; теорії трансформерів та механізмів уваги – при модифікації CLIP-енкодера та реалізації механізму умовної модуляції візуальних ознак на основі текстового запиту; теорії метричних просторів та розбиттів – при розробці зваженої метрики для порівняння сегментацій та доведенні її аксіоматичних властивостей; методів кластеризації та аналізу зв'язності – при розробці алгоритмів постобробки на основі керованого зростання регіонів.

Наукова новизна одержаних результатів:

1) Вперше запропоновано модель InstanceCLIPSeg для сегментації екземплярів за текстовим запитом, яка, на відміну від відомих двоетапних методів, містить модифіковану архітектуру CLIPSeg з двома декодерами для прогнозування центрів об'єктів та попіксельних відстаней до меж обмежувальної рамки, що забезпечує одноетапну open-set сегментацію екземплярів та розширює можливості системи для адаптивного розпізнавання нових класів об'єктів у режимі реального часу.

2) Вперше запропоновано зважену метрику для порівняння сегментацій зображень на основі теорії розбиттів, яка, на відміну від сучасних функцій втрат (Dice, Tversky, IoU, Lovász-Softmax), задовольняє аксіоми метричного простору включно з нерівністю трикутника та враховує геометричні та семантичні властивості регіонів через вагову функцію. Математично доведено виконання аксіом невід'ємності, рефлексивності, симетрії та нерівності трикутника, що відкриває можливості для ієрархічного пошуку у просторі розбиттів. Продемонстровано можливість використання розробленої функції для навчання нейронної мережі через апроксимацію Straight-Through Estimator.

3) Удосконалено метод постобробки результатів висхідної сегментації, який, на відміну від наявних підходів на основі кластеризації, використовує аналіз карт відстаней та адаптивне розширення регіонів від центрів об’єктів з урахуванням просторової зв’язності пікселів.

4) Набув подальшого розвитку підхід до відновлення роздільної здатності в декодерах сегментаційних мереж. Досліджено та порівняно механізми білінійної інтерполяції, транспонованої згортки та PixelShuffle. Встановлено, що використання PixelShuffle з ICNR-ініціалізацією у поєднанні з багаторівневим злиттям ознак із різних шарів енкодера дозволяє уникнути артефактів типу «шахової дошки» та забезпечує вищу точність, у порівнянні з іншими підходами.

Ключові слова: глибоке навчання, сегментація зображень, згорткові нейронні мережі, трансформери, CLIP, open-set сегментація, сегментація екземплярів, текстовий запит, функція втрат, метрика.

Список публікацій здобувача

1. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Experimental study of distance map decoder architectural configurations for instance segmentation by text query. System technologies, 2(163), 3–12. https://doi.org/10.34185/1562-9945-2-163-2026-01

2. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Metrical loss functions for image segmentation based on convolutional neural networks. Applied Aspects of Information Technology, 9(2), 172–184. https://doi.org/10.15276/aait.09.2026.12

3. Kovtunenko, A. R., & Mashtalir, S. V. (2025). Improved segmentation model to identify object instances based on textual prompts. Herald of Advanced Information Technology, 8(1), 54–66. https://doi.org/10.15276/hait.08.2025.4

4. Kovtunenko, A. R., & Mashtalir, S. V. (2025). A Review of Modern Neural Network Architectures for Image Segmentation. Management Information System and Devises, (185), 53–62. https://doi.org/10.30837/0135-1710.2025.185.043

5. Ковтуненко, А. Р. (2025). Мультимодальна висхідна сегментація об’єктів за текстовим запитом. У III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (с. 11–12). Західноукраїнський національний університет

6. Ковтуненко, А., & Машталір, В. (2024). Аналіз та порівняння функцій втрат для вирішення задачі сегментації. У Радіоелектроніка та молодь у XXI столітті. Т. 7: Конференція «Комп’ютерний зір, системний аналіз та математичне моделювання». Press of the Kharkiv National University of Radioelectronics. (с. 57–59). https://doi.org/10.30837/iyf.cvsamm.2024.057

ANNOTATION

Kovtunenko A.R. Neural networks for multimodal open-vocabulary spatio-semantic segmentation with ambiguous queries – Qualification scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in the specialty 122 – Computer Science (12 – Information Technologies). – Kharkiv National University of Radio Electronics, Ministry of Education and Science of Ukraine, Kharkiv, 2026.

The dissertation is devoted to the development of methods and models for multimodal image segmentation via arbitrary text prompts with the capability of isolating individual object instances based on architectural modifications of neural networks and improved post-processing algorithms.

The dissertation was carried out with the support of the Horizon Europe project «Supporting European R&I Through stakeholder collaboration and institutional reform» (INITIATE, HORIZON-WIDERA-2023-ACCESS-03, project number 101136775).

The rapid growth in multimedia information volumes poses new requirements for automated processing and analysis methods. Among the key directions of such processing, image segmentation holds a special place as a fundamental computer vision task that involves dividing an image into semantically homogeneous regions. Classical segmentation methods based on a fixed set of classes (closed-set) demonstrate high efficiency on standard benchmarks; however, their application in real-world conditions is significantly limited by architectural dependence on predefined object categories.

In practical scenarios, objects are characterized by variable properties and can be described through attributes, actions, or partial category membership. Moreover, object interactions generate new contextual features that are fundamentally impossible to predict during training set formation. This necessitated a transition to the open-set paradigm, where the set of classes is not fixed and can be defined by an arbitrary text query (open-vocabulary). Such an approach allows users to formulate

queries in natural language, describing target objects through their visual characteristics, spatial location, or functional purpose.

A revolutionary step in this direction was the emergence of the CLIP model (Contrastive Language-Image Pretraining), which enabled the alignment of visual and textual representations in a unified semantic space. Through contrastive learning on a massive corpus of image-text pairs, the model acquired the ability to align visual content with natural language descriptions without being tied to a specific set of classes. Further development of multimodal architectures, including SAM (Segment Anything Model) and CLIPSeg, demonstrated the potential of approaches where the segmentation query is formulated in natural language.

However, existing methods for instance segmentation via text prompts are predominantly implemented as two-stage procedures, where object detection is performed first, followed by segmentation. A typical representative of this approach is Grounded-SAM, which sequentially applies the GroundingDINO detector and SAM segmentator. Such architecture leads to significant computational costs, increases system latency, and does not fully exploit the advantages of multimodal learning. Furthermore, detection errors at the first stage inevitably affect the quality of final segmentation.

A separate scientific problem remains the choice of loss function for training segmentation neural networks. Widely used functions such as Dice, despite their practical effectiveness, lack rigorous mathematical justification as metrics in the classical sense of metric space theory. In particular, these functions do not satisfy the triangle inequality, which limits their use in hierarchical segmentation approaches and efficient search algorithms in partition space. The triangle inequality is a fundamentally important property, as it allows quickly eliminating obviously dissimilar variants when comparing segmentations.

Thus, the development of models, methods, and algorithms for multimodal spatio-semantic segmentation with ambiguous queries based on deep learning architecture modifications to improve the efficiency of object instance extraction under the open-vocabulary paradigm represents a relevant scientific task.

The aim of the dissertation is the development of models, methods, and algorithms for multimodal spatio-semantic segmentation with ambiguous queries based on deep learning architecture modifications to improve the efficiency of object instance extraction in open-vocabulary mode.

To achieve the research aim, the following objectives are addressed in the dissertation:

1) perform an analysis of problems and approaches to solving the image segmentation task via text prompts under the open-set paradigm;

2) develop a modified instance segmentation model architecture based on CLIPSeg for predicting object centers and distances to boundaries;

3) develop a post-processing method for bottom-up segmentation results to separate object instances;

4) develop a metric loss function for comparing segmentations based on partition theory;

5) perform experimental verification of the developed methods and models on standard datasets.

The object of the study is the process of image segmentation via arbitrary text prompts.

The subject of research is the models, methods, and algorithms of multimodal spatio-semantic segmentation based on deep learning.

The results of the dissertation research are based on the use of: deep learning theory and convolutional neural networks – in the development of the InstanceCLIPSeg architecture with decoders for predicting centers and distances; transformer theory and attention mechanisms – in the modification of the CLIP encoder and implementation of the conditional modulation mechanism for visual features based on text queries; metric space and partition theory – in the development of a weighted metric for comparing segmentations and proving its axiomatic properties; clustering methods and connectivity analysis – in the development of post-processing algorithms based on controlled region growing.

The scientific novelty of the obtained results:

1) For the first time, the InstanceCLIPSeg model is proposed for text-prompted instance segmentation. Unlike existing two-stage methods, it features a modified CLIPSeg architecture with two decoders for predicting object centers and per-pixel distances to bounding box boundaries. This ensures one-stage open-set instance segmentation and expands the system's capabilities for adaptive, real-time recognition of novel object classes.

2) A weighted metric for comparing image segmentations based on partition theory is proposed for the first time, which, unlike existing loss functions (Dice, Tversky, IoU, Lovász-Softmax), satisfies the axioms of a metric space including the triangle inequality and takes into account geometric and semantic properties of regions through a weighting function. Mathematical proofs of non-negativity, reflexivity, symmetry, and triangle inequality axioms are provided, opening possibilities for hierarchical search in partition space. It is demonstrated that the developed function can be used for neural network training via Straight-Through Estimator approximation.

3) The method of post-processing bottom-up segmentation results has been improved, which, unlike existing clustering-based approaches, utilizes distance map analysis and adaptive region expansion from object centers taking into account the spatial connectivity of pixels.

4) The approach to resolution recovery in segmentation network decoders has been further developed. Bilinear interpolation, transposed convolution, and PixelShuffle mechanisms were investigated and compared. It has been established that the use of PixelShuffle with ICNR initialization, combined with multi-level feature fusion from different encoder layers, eliminates checkerboard artifacts and yields superior accuracy compared to other approaches.

Keywords: deep learning, image segmentation, convolutional neural networks, transformers, CLIP, open-set segmentation, instance segmentation, text prompt, loss function, metric.

List of publications

1. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Experimental study of distance map decoder architectural configurations for instance segmentation by text query. System technologies, 2(163), 3–12. https://doi.org/10.34185/1562-9945-2-163-2026-01

2. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Metrical loss functions for image segmentation based on convolutional neural networks. Applied Aspects of Information Technology, 9(2), 172–184. https://doi.org/10.15276/aait.09.2026.12

3. Kovtunenko, A. R., & Mashtalir, S. V. (2025). Improved segmentation model to identify object instances based on textual prompts. Herald of Advanced Information Technology, 8(1), 54–66. https://doi.org/10.15276/hait.08.2025.4

4. Kovtunenko, A. R., & Mashtalir, S. V. (2025). A Review of Modern Neural Network Architectures for Image Segmentation. Management Information System and Devises, (185), 53–62. https://doi.org/10.30837/0135-1710.2025.185.043

5. Ковтуненко, А. Р. (2025). Мультимодальна висхідна сегментація об’єктів за текстовим запитом. У III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (с. 11–12). Західноукраїнський національний університет

6. Ковтуненко, А., & Машталір, В. (2024). Аналіз та порівняння функцій втрат для вирішення задачі сегментації. У Радіоелектроніка та молодь у XXI столітті. Т. 7: Конференція «Комп’ютерний зір, системний аналіз та математичне моделювання». Press of the Kharkiv National University of Radioelectronics. (с. 57–59). https://doi.org/10.30837/iyf.cvsamm.2024.057

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ОДИНИЦЬ І ТЕРМІНІВ

ASPP – просторова пірамідна агрегація з розрідженням (англ. Atrous Spatial Pyramid Pooling);

BCE – бінарна крос-ентропія (англ. Binary Cross-Entropy);

BFS – пошук у ширину (англ. Breadth-First Search);

BPE – кодування пар байтів (англ. Byte Pair Encoding);

Backbone – мережа виділення ознак;

CE – крос-ентропія (англ. Cross-Entropy);

CIoU – повна міра перетину над об’єднанням (англ. Complete Intersection over Union);

CLIP – контрастивне текстово-візуальне попереднє навчання (англ. Contrastive Language-Image Pretraining);

CLIPSeg – модель сегментації на основі CLIP;

CNN – згорткова нейронна мережа (англ. Convolutional Neural Network);

CoordConv – координатна згортка (англ. Coordinate Convolution);

DETR – трансформер для виявлення об'єктів (англ. Detection Transformer);

DIoU – дистанційна міра перетину над об’єднанням (англ. Distance Intersection over Union);

DSC – коефіцієнт Соренсена-Дайса (англ. Dice-Sørensen Coefficient);

EMD – відстань землекопа (англ. Earth Mover’s Distance);

FCN – повнозгорткова мережа (англ. Fully Convolutional Network);

FiLM – лінійна модуляція на рівні ознак (англ. Feature-wise Linear Modulation);

FN – хибнонегативний результат (англ. False Negative);

FP – хибнопозитивний результат (англ. False Positive);

FPN – піраміда ознак (англ. Feature Pyramid Network);

FPS – кадрів на секунду (англ. Frames Per Second);

FSL – навчання з малою вибіркою (англ. Few-Shot Learning);

GIoU – узагальнена міра перетину над об’єднанням (англ. Generalized Intersection over Union);

GPU – графічний процесор (англ. Graphics Processing Unit);

HQ-SAM – високоякісна модель сегментації будь-чого (англ. High-Quality Segment Anything Model);

ICNR – ініціалізована субпіксельна згортка (англ. Initialized Sub-Pixel Convolution);

IoU – міра перетину над об’єднанням (англ. Intersection over Union);

JPU – спільне пірамідне підвищення роздільної здатності (англ. Joint Pyramid Upsampling);

KNN – метод k-найближчих сусідів (англ. k-Nearest Neighbors);

LVIS – набір даних для сегментації екземплярів з великим словником (англ. Large Vocabulary Instance Segmentation);

MiCROM – зіставлення регіонів з мінімальною вартістю (англ. Minimum-Cost Region Matching);

MLP – багатошаровий перцептрон (англ. Multilayer Perceptron);

MSE – середньоквадратична помилка (англ. Mean Square Error);

NLP – обробка природної мови (англ. Natural Language Processing);

NMS – немаксимальне придушення (англ. Non-Maximum Suppression);

PRN – паноптична уточнювальна мережа (англ. Panoptic Refinement Network);

ReLU – випрямлена лінійна функція активації (англ. Rectified Linear Unit);

RoI – область інтересу (англ. Region of Interest);

SAM – модель сегментації будь-чого (англ. Segment Anything Model);

STE – оцінювач прямого проходження (англ. Straight-Through Estimator);

ViT – візуальний трансформер (англ. Vision Transformer);

ZSL – навчання з нульовою вибіркою (англ. Zero-Shot Learning).

ВСТУП

Актуальність теми. Сучасний етап розвитку систем комп'ютерного зору характеризується переходом від класичних методів детекції та сегментації з фіксованим набором класів до систем, здатних до комплексного розуміння сцени на основі запитів природною мовою. Традиційні підходи, як правило, базуються на закритому наборі класів (closed-set), визначених на етапі навчання, що накладає фундаментальне обмеження та унеможливлює роботу з об'єктами, які відрізняються від еталонних зразків навчальної вибірки.

Проте у реальних умовах об'єкти мають варіативні характеристики, атрибути і можуть взаємодіяти між собою, породжуючи нові контекстні атрибути, які неможливо передбачити заздалегідь [7, 8]. Це зумовило необхідність переходу до парадигми open-set методів, де кількість класів не є фіксованою і може бути заданою текстовим запитом довільної форми (open-vocabulary) [9, 10].

Поява моделі CLIP [11] та подальший розвиток мультимодальних архітектур, таких як SAM [12], демонструють потенціал підходів, у яких запит можна описати природною мовою. На відміну від класичного closed-set підходу, open-set дозволяє враховувати атрибути, дії або часткову належність, що забезпечує моделювання складних просторово-семантичних зв'язків, які не були явно задані при навчанні. Слід зазначити, що текстовий опис не завжди здатен вичерпно передати усі характеристики об'єкта, такі як специфічна текстура, відтінок або складна геометрія. У таких випадках мультимодальний підхід [13-15] має передбачати можливість використання візуального зразка для пошуку. Саме тому актуальним є дослідження нейромережевих методів мультимодальної просторово-семантичної сегментації за нечітким запитом.

Існуючі методи сегментації екземплярів поділяються на одноетапні та двоетапні залежно від швидкості та точності. Двоетапні методи, такі як

Grounded-SAM [16], спочатку виявляють об’єкти за допомогою детектора, а потім передають їх координати до моделі сегментації. Такий підхід є обчислювально витратним та не дозволяє повною мірою використовувати переваги мультимодального навчання. Одноетапні open-set методи сегментації екземплярів за текстовим запитом нині відсутні, що зумовлює актуальність розробки відповідних архітектур.

Окремою проблемою є вибір функції втрат для навчання сегментаційних мереж. Широко використовувані функції, такі як Dice [17] та IoU [18], не мають строгого математичного обґрунтування як власне метрики і не повністю враховують геометричну структуру розбиттів. Розробка метричних функцій втрат, що задовольняють аксіоми метричного простору, відкриває можливості для ієрархічних підходів до сегментації та ефективного пошуку розбиттів.

Таким чином, **актуальним науковим завданням** є розробка моделей, методів та алгоритмів мультимодальної просторово-семантичної сегментації за нечітким запитом для підвищення ефективності виділення екземплярів об’єктів в умовах open-vocabulary парадигми.

Зв’язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконана відповідно до плану науково-дослідних робіт Харківського національного університету радіоелектроніки та була виконана за підтримки проєкту Horizon Europe «Підтримка європейської дослідницько-інноваційної діяльності через співпрацю стейкхолдерів та інституційну реформу» («Supporting European R&I Through stakeholder collaboration and institutional reform», INITIATE, HORIZON-WIDERA-2023-ACCESS-03, номер проєкту 101136775).

Мета і завдання дослідження. Метою дисертаційної роботи є розробка моделей, методів та алгоритмів мультимодальної просторово-семантичної сегментації за нечітким запитом на основі модифікації архітектур глибокого навчання для підвищення ефективності виділення екземплярів об’єктів у режимі open-vocabulary.

Для досягнення мети дослідження в дисертаційній роботі вирішуються такі завдання:

1) виконати аналіз проблем та підходів до вирішення задачі сегментації зображень за текстовим запитом в умовах open-set парадигми;

2) розробити модифіковану архітектуру моделі сегментації екземплярів на основі CLIPSeg [19] для прогнозування центрів об'єктів та відстаней до меж;

3) розробити метод постобробки результатів висхідної сегментації для розділення екземплярів об'єктів;

4) розробити метричну функцію втрат для порівняння сегментацій на основі теорії розбиттів;

5) виконати експериментальну перевірку розроблених методів та моделей на стандартних наборах даних.

Об'єкт дослідження – процес сегментації зображень за текстовим запитом довільної форми.

Предмет дослідження – моделі, методи та алгоритми мультимодальної просторово-семантичної сегментації на основі глибокого навчання.

Методи дослідження: результати дисертаційного дослідження базуються на використанні: теорії глибокого навчання та згорткових нейронних мереж – при розробці архітектури InstanceCLIPSeg з декодерами прогнозування центрів (centers head) та зміщень (offset/distance head); теорії трансформерів та механізмів уваги – при модифікації CLIP-енкодера для роздільної здатності 352×352 пікселів; теорії метричних просторів та розбиттів – при розробці зваженої метрики для порівняння сегментацій; методів кластеризації та аналізу зв'язності – при розробці алгоритмів постобробки для виділення екземплярів.

Наукова новизна одержаних результатів. На основі проведених теоретичних та експериментальних досліджень отримано такі нові наукові результати.

1) Вперше запропоновано модель InstanceCLIPSeg для сегментації екземплярів за текстовим запитом, яка, на відміну від відомих двоетапних методів [16, 20], містить модифіковану архітектуру CLIPSeg з двома декодерами для прогнозування центрів об’єктів та попіксельних відстаней до меж обмежувальної рамки, що забезпечує одноетапну open-set сегментацію екземплярів та розширює можливості системи для адаптивного розпізнавання нових класів об'єктів у режимі реального часу.

2) Вперше запропоновано зважену метрику для порівняння сегментацій зображень на основі теорії розбиттів, яка, на відміну від сучасних функцій втрат (Dice, Tversky [21], IoU, Lovász-Softmax [22]), задовольняє аксіоми метричного простору включно з нерівністю трикутника та враховує геометричні та семантичні властивості регіонів через вагову функцію. Математично доведено виконання аксіом невід’ємності, рефлексивності, симетрії та нерівності трикутника, що відкриває можливості для ієрархічного пошуку у просторі розбиттів. Продемонстровано можливість використання розробленої функції для навчання нейронної мережі через апроксимацію Straight-Through Estimator [23].

3) Удосконалено метод постобробки результатів висхідної сегментації, який, на відміну від наявних підходів на основі кластеризації, використовує аналіз карт відстаней та адаптивне розширення регіонів від центрів об’єктів з урахуванням просторової зв’язності пікселів.

4) Набув подальшого розвитку підхід до відновлення роздільної здатності в декодерах сегментаційних мереж. Досліджено та порівняно механізми білінійної інтерполяції, транспонованої згортки та PixelShuffle [24]. Встановлено, що використання PixelShuffle з ICNR-ініціалізацією у поєднанні з багаторівневим злиттям ознак із різних шарів енкодера дозволяє уникнути артефактів типу «шахової дошки» та забезпечує вищу точність, у порівнянні з іншими підходами.

Практичне значення одержаних результатів. Практичне значення отриманих у дисертаційній роботі результатів полягає у розробці

інструментарію для систем автоматизованого моніторингу, відеоспостереження та робототехніки, який дозволяє майже в режимі реального часу здійснювати пошук об'єктів за довільним текстовим запитом або еталонним зображенням. Завдяки запропонованій одноетапній архітектурі суттєво знижуються обчислювальні витрати порівняно з існуючими двоетапними рішеннями, а відсутність прив'язки до фіксованого словника (open-vocabulary) дозволяє адаптувати систему до нових умов та непередбачуваних сценаріїв без необхідності ресурсоємного перенавчання моделі.

Крім того, застосування запропонованої метрики дозволяє виконувати ієрархічний пошук, оптимізувати процес навчання моделей та здійснювати швидке відсіювання завідомо несхожих варіантів при зіставленні сегментацій, що є недосяжним для існуючих функцій втрат.

Отримані результати були впроваджені в ТОВ «Сайтосс» (акт впровадження від 30.04.2026) та в освітній процес Харківського національного університету радіоелектроніки (акт впровадження від 12.05.2026).

Особистий внесок здобувача. Основні наукові результати, що виносяться на захист, отримані автором самостійно. У роботах, опублікованих у співавторстві, здобувачеві належить: [1] – розробка архітектур та проведення експериментів; [2] – адаптація метрики для використання як функції втрат та проведення експериментів; [3] – розробка архітектури моделі, алгоритму постобробки, проведення експериментів; [4] – аналіз існуючих методів сегментації; [5] – розробка алгоритму постобробки; [6] – аналіз та порівняння існуючих функцій втрат.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися на наукових конференціях:

1) III Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі»;

2) 28-му Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті».

Публікації. Результати дисертаційних досліджень опубліковано у 4 наукових працях, які містять 3 статті у фахових періодичних наукових виданнях України категорії «Б», 1 у фахових періодичних наукових виданнях України категорії «А».

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел. Загальний обсяг дисертаційної роботи становить 161 сторінку, у тому числі 136 сторінок основного тексту, 67 ілюстрацій, 10 таблиць, 3 додатки.

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ ЗА ТЕКСТОВИМ ЗАПИТОМ

1.1 Аналіз процесу сегментації зображень та її типів

Однією з фундаментальних функцій комп'ютерного зору є розуміння та інтерпретація навколишнього простору – зображень та відео. Універсальний підхід до вирішення цієї задачі наразі відсутній, і процес пошуку продовжується. Кількість мультимедійної інформації, що стрімко зросла, вимагає суттєвого розвитку методів її швидкої обробки. При цьому одним із напрямів обробки є попередній аналіз і виділення характерних ознак із зображень для стиснення інформації, необхідної для подальших завдань. Одним із видів такого стиснення інформації є сегментація зображень.

Парадигма сучасних систем комп'ютерного зору зміщується від класичних методів детекції, сегментації та розпізнавання до систем, здатних до комплексного розуміння сцени. Традиційні підходи, як правило, базуються на закритому наборі класів (closed-set), визначених ще на етапі навчання. Такий підхід накладає фундаментальне обмеження, унеможливлюючи роботу з об'єктами, що відрізняються від еталонних зразків навчальної вибірки. Проте у реальних умовах об'єкти мають варіативні характеристики, атрибути і можуть взаємодіяти між собою, внаслідок чого виникають нові контекстні атрибути, які неможливо передбачити заздалегідь.

Процес сегментації є поділом даних або зображень на логічні та взаємопов'язані частини, що представляють об'єкти, області або категорії. Головна мета сегментації полягає у спрощенні або трансформації даних у більш зрозумілу та легшу для аналізу форму, виділяючи лише найбільш релевантні об'єкти та деталі. Сегментація зображень є фундаментальною задачею комп'ютерного зору, що вимагає точного піксельного розмежування об'єктів і розуміння сцени.

Семантична сегментація (semantic segmentation) – це задача комп'ютерного зору, в якій кожному пікселю зображення присвоюється мітка класу. На відміну від класифікації зображень, яка визначає клас усього зображення, семантична сегментація забезпечує попіксельне розуміння сцени. Результатом роботи алгоритму семантичної сегментації є маска, де кожен піксель має відповідну мітку класу [25, 26].

Основними характеристиками семантичної сегментації є:

1) присвоєння кожному пікселю мітки класу з попередньо визначеного набору;

2) відсутність розрізнення між окремими екземплярами одного класу;

3) формування щільної карти передбачень розміром, що відповідає вхідному зображенню.

Семантична сегментація знаходить широке застосування в автономному водінні для розуміння дорожньої сцени, медичній візуалізації для виділення анатомічних структур, аналізі супутникових знімків для картографування земельного покриву та інших галузях.

Сегментація екземплярів (instance segmentation) є задачею комп'ютерного зору, що поєднує семантичну сегментацію та виділення меж кожного об'єкта. На відміну від семантичної сегментації, яка присвоює мітку класу кожному пікселю без розрізнення окремих об'єктів, сегментація екземплярів розглядає кожне входження об'єкта класу як окрему сутність. Це забезпечує більш детальне розуміння візуальних сцен. Такий підхід є незамінним у сценаріях, де розуміння взаємозв'язків, розділення об'єктів одного класу та їх взаємодії є критично важливим [27].

Паноптична сегментація (panoptic segmentation) об'єднує семантичну сегментацію та сегментацію екземплярів в єдину задачу. Вона передбачає присвоєння кожному пікселю зображення як семантичної мітки класу, так і ідентифікатора екземпляра для об'єктів, що підлягають підрахунку (things), при цьому аморфні області (stuff) отримують лише семантичну мітку [28, 29].

Формально, для паноптичної сегментації визначаються два типи категорій:

1) things – об’єкти, що мають чітко визначену форму та можуть бути підраховані (люди, автомобілі, тварини);

2) stuff – аморфні області без чіткої форми (небо, трава, дорога).

Підходи до сегментації екземплярів можна класифікувати за кількома критеріями. За кількістю етапів обробки виділяють одноетапні та двоетапні методи.

Одноетапні методи (one-stage) спрямовані на прогнозування об’єктів та їх масок за один крок, без використання додаткових методів для знаходження областей інтересу або пошуку об’єктів. До таких методів належать: YOLACT [30], SOLO [31], PolarMask [32], MEInst [33], CenterMask [34]. Такий підхід має перевагу у швидкості, проте може поступатися у точності.

Двоетапні методи (two-stage) розділяють задачу на дві окремі фази: детекцію областей інтересу та подальшу сегментацію для уточнення піксельної маски для кожного екземпляра об’єкта. До них належать: Mask R-CNN [35], RefineMask [36], Mask Transfiner [37], TensorMask [38], Polytransform [39], BCNet [40]. Натомість, ці методи є точнішими, але потребують більше обчислювальних ресурсів.

За напрямком обробки виділяють:

1) top-down – спочатку виявляються цільові об’єкти, а потім уточнюються їх маски сегментації;

2) bottom-up – спочатку ідентифікуються відповідні індивідуальні ознаки, які потім групуються в екземпляри або сегменти, що належать одному об’єкту.

Іншими словами, у підході top-down метою є спочатку знайти об’єкти, а потім присвоїти їм унікальні ідентифікатори та уточнити межі, якщо об’єкти були знайдені через детекцію. У підході bottom-up метою є спочатку ідентифікувати ознаки, які дають зрозуміти, що це об’єкт інтересу, а потім

об’єднати ці ознаки в різні об’єкти. Цей підхід добре ілюструється пошуком ознак для кожного пікселя з подальшою їх кластеризацією.

Також можливе поєднання цих методів та підходів для вирішення поставленої задачі. Як правило, методи bottom-up поступаються у точності методам top-down, особливо на наборах даних зі складними формами та великою їх різноманітністю. Натомість, методи top-down мають проблеми з об’єктами малого розміру та можуть генерувати множинні маски з перекриттям, що зумовлює необхідність додаткової постобробки.

Таблиця 1.1 – Порівняння характеристик одноетапних та двоетапних методів сегментації екземплярів

Характеристика	**One-stage**	**Two-stage**
Швидкість	Висока	Нижча
Точність	Нижча	Висока
Обчислювальні ресурси	Менше	Більше
Робота з малими об’єктами	Краща	Гірша
Обробка перекриттів	Складніша	Простіша

1.2 Еволюція архітектур сегментації

Розвиток архітектур нейронних мереж для задач сегментації пройшов значний шлях від простих повнозгорткових моделей до сучасних архітектур на основі трансформерів [41-43]. Розуміння цієї еволюції є важливим для аналізу сучасного стану проблеми та визначення напрямків подальших досліджень.

Архітектура FCN [44] стала першою, що дозволила вирішувати задачу сегментації для зображень довільної роздільної здатності без необхідності постобробки результатів. Основна ідея полягала у заміні повнозв’язних шарів на згорткові та додаванні шарів для відновлення розмірності. Проте,

незважаючи на ці переваги, архітектура показала незадовільні результати для сегментації складних сцен, малих об’єктів та виділення меж (рис. 1.1).

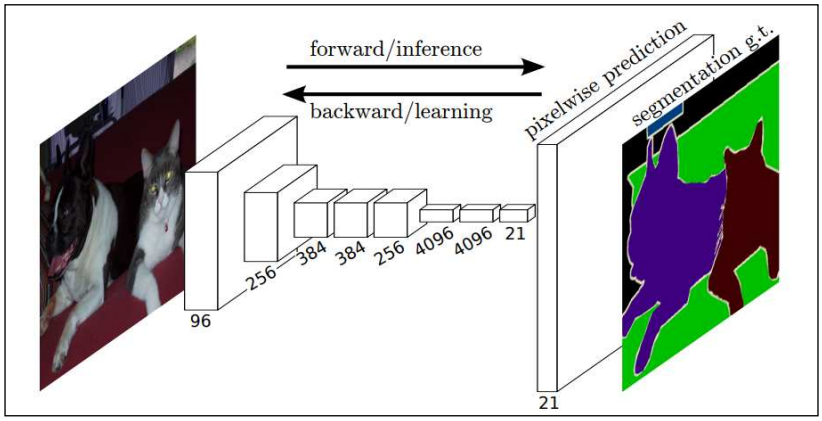

Рисунок 1.1 – Архітектура FCN: пряме та зворотне проходження

Поява U-Net [45] вирішила частину проблем FCN, особливо в контексті обробки дрібних деталей на зображеннях та виділення меж. U-Net також є повнозгортковою нейронною мережею, яка вже має skip-з’єднання, що дозволяє моделі отримувати інформацію про об’єкти з різних рівнів представлень та відновлювати їх просторову інформацію. Введення зваженої функції втрат зі збільшеними коефіцієнтами для граничних пікселів дозволило точно виділяти межі об’єктів (рис. 1.2).

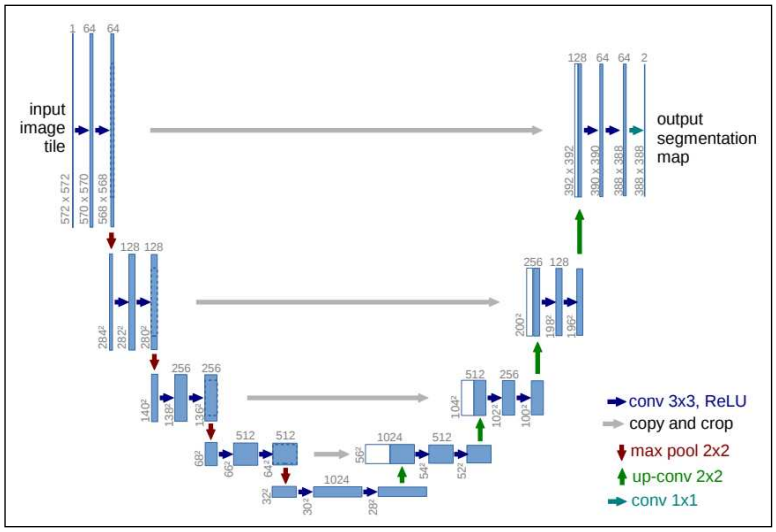

Рисунок 1.2 – Архітектура U-Net з skip-з’єднаннями

Архітектура була спочатку розроблена для медичних зображень та малих обсягів вихідних даних. Застосування аугментації до даних дозволило досягти високої точності розпізнавання, але слід зазначити, що при недостатній варіативності набору даних існує ризик перенавчання моделі. До недоліків U-Net належать обчислювальні витрати, особливо при збільшенні розміру вхідного зображення, та обмежене рецептивне поле, що може ускладнити аналіз структур, які потребують більшого контексту.

SegNet [46] порівняно з U-Net дозволяє обробляти зображення високої роздільної здатності з меншими ресурсними витратами, при цьому достатньо точно виділяючи межі об’єктів. Вона була спеціально розроблена для систем, що потребують високошвидкісної обробки даних, навіть у реальному часі. SegNet також використовує двочастинну структуру енкодер-декодер, але застосовує модифіковане підвищення дискретизації в модулі декодера (рис. 1.3).

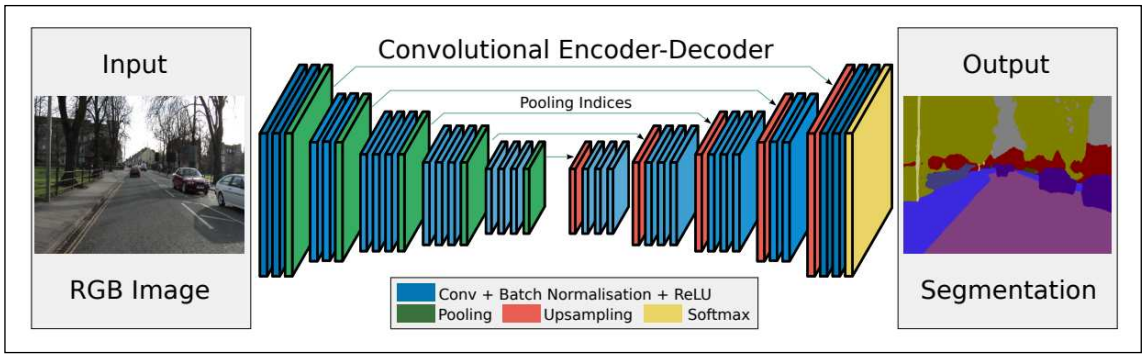

Рисунок 1.3 – Архітектура згорткового енкодеру-декодеру SegNet

Просторова роздільна здатність відновлюється за допомогою індексів max pooling, збережених на відповідних рівнях енкодера під час операції max pooling. При цьому карти ознак безпосередньо не передаються, що суттєво зменшує кількість параметрів моделі. Оскільки skip-з’єднання не застосовуються, це може призвести до часткової втрати інформації про глобальний контекст зображення, що потенційно знижує ефективність сегментації у сценаріях зі складними просторовими структурами.

Сімейство моделей DeepLab (DeepLabV1 [47], DeepLabV2 [48], DeepLabV3 [49], DeepLabV3+ [50]) представляє послідовну еволюцію архітектур, спрямовану на покращення точності семантичної сегментації. Кожна ітерація моделі внесла свої доповнення та покращення.

DeepLabV1 застосував Atrous Convolutions, що характеризуються введенням специфічних проміжків між елементами згорткового ядра. Такий підхід дозволяє збільшувати рецептивне поле без зміни роздільної здатності карти ознак. DeepLabV2 розвиває концепцію розширених згорток, вводячи Atrous Spatial Pyramid Pooling (ASPP) – набір згорток з різними значеннями параметрів розширення. Це рішення суттєво покращило якість сегментації об’єктів різних масштабів на зображеннях.

У DeepLabV3 архітектура ASPP була доповнена компонентом глобального усереднення (global pooling). Його переваги полягають у тому, що він охоплює всю карту ознак і зводить її до єдиного вектора, фіксує контекст на рівні зображення, що допомагає зрозуміти загальну сцену, та доповнює локальні ознаки, захоплені згортковими шарами. Також розширені згортки були додані до всіх частин моделі (рис. 1.4).

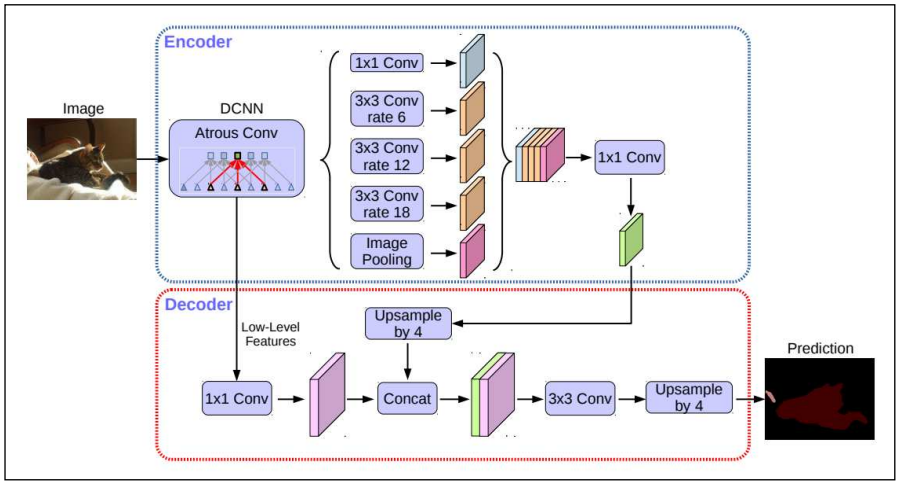

Рисунок 1.4 – Архітектура DeepLab V3+: енкодер з ASPP та декодер

У DeepLabV3+ архітектуру було змінено на енкодер-декодер, подібно до U-Net. Мережі ResNet, що використовувалися в попередніх версіях, або

модифікована архітектура Xception використовуються як енкодер-декодер у цій архітектурі. Підвищення дискретизації було зроблено у два етапи для покращення гранулярності. Складність архітектури покращила якість сегментації, особливо меж об’єктів, але природно призвела до збільшення обчислювальних витрат.

Архітектура Mask R-CNN [35] є архітектурною еволюцією підходів, закладених у Faster R-CNN [51], але з розширеною функціональністю для вирішення задачі сегментації екземплярів, де потрібно розділяти кожен об’єкт одного класу. Ця архітектура реалізує двоетапний підхід: перший етап передбачає прогнозування областей інтересу (RoI) на вхідному зображенні, після чого RoIAlign точно витягує відповідні регіони з карт ознак, згенерованих мережею виділення ознак (backbone). На другому етапі виділені регіони класифікуються, і паралельно окрема гілка моделі генерує маски сегментації для кожного ідентифікованого об’єкта (рис. 1.5).

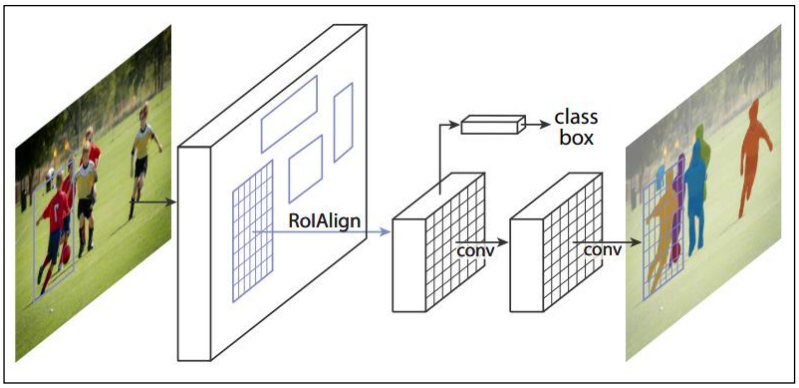

Рисунок 1.5 – Архітектура Mask R-CNN

Цей підхід став стандартом на тривалий період і був здатний передбачати точні маски навіть для об’єктів, що перекриваються. Однак цей підхід є обчислювально витратним і все ще показує неоптимальні результати для об’єктів малого розміру. Важливою перевагою Mask R-CNN є її універсальність – вона може бути застосована до класифікації, детекції та сегментації.

Некоректне виділення меж об’єктів, особливо у випадках їх взаємного перекриття, залишається одним із фундаментальних обмежень алгоритмів семантичної сегментації. У методі Gated-SCNN [52] автори спрямували зусилля на вирішення цієї проблеми. Для цього вони використали дві гілки в архітектурі. Основна backbone-гілка займається виділенням ознак з вхідного зображення, які потім передаються через Gated Convolution Layer у другу гілку, що працює з оригінальним зображенням, попередньо обробленим фільтром Канні – тобто з градієнтами (рис. 1.6).

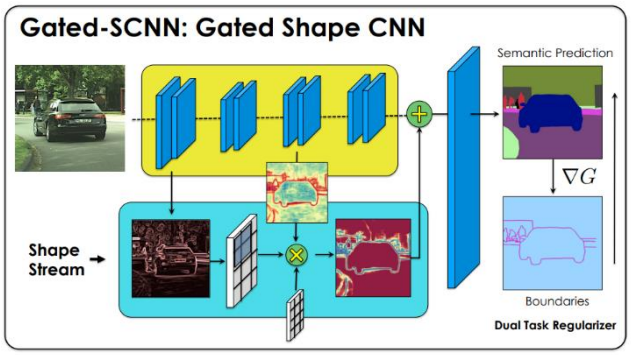

Рисунок 1.6 – Архітектура Gated-SCNN: Gated Shape CNN

Результати цих двох гілок потім об’єднуються через ASPP для формування кінцевого результату сегментації. Слід зазначити, що модуль Gated Convolution Layer, запропонований у цій архітектурі, концептуально еквівалентний сучасним механізмам уваги. Однак основним внеском цієї роботи є нова функція втрат – dual task loss, яка надає можливість порівнювати передбачені межі об’єктів з істинними.

FastFCN [53] є ще одним підходом до семантичної сегментації, в якому запропоновано новий блок Joint Pyramid Upsampling (JPU) для підвищення дискретизації. У цьому блоці вхідні карти ознак спочатку обробляються стандартною згорткою, а потім кількома роздільними згортками (separable convolution) з різними коефіцієнтами розширення. Автори дослідження використовують останні три згорткові шари з backbone-мережі, хоча кількість згорткових шарів може варіюватися (рис. 1.7).

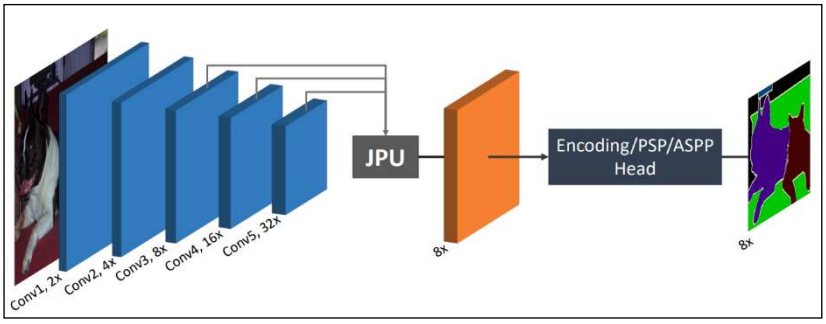

Рисунок 1.7 – Архітектура FastFCN з блоком JPU

Такий підхід покращує реконструкцію дрібних деталей та суттєво зменшує обчислювальні витрати. Після JPU застосовується модуль мультимасштабного/глобального контексту для генерації фінальної карти сегментації, такий як ASPP. Основними перевагами цього підходу є збільшена обчислювальна швидкість та можливість застосування JPU до інших вже відомих архітектур або їх модулів, наприклад DeepLabV3 або інших окремих backbone-мереж.

MaskFormer [54] є моделлю сегментації, яка інтегрує механізм уваги у свою архітектуру. Її фундаментальна концептуальна відмінність полягає не у попіксельній класифікації чи виділенні областей інтересу, а у первинній генерації набору масок об’єктів, які потім класифікуються. Для реалізації цього підходу backbone витягує embeddings, які обробляються піксельним декодером та трансформерним декодером для генерації бінарних масок об’єктів. Архітектура декодерів є box-free DETR [55] (рис. 1.8).

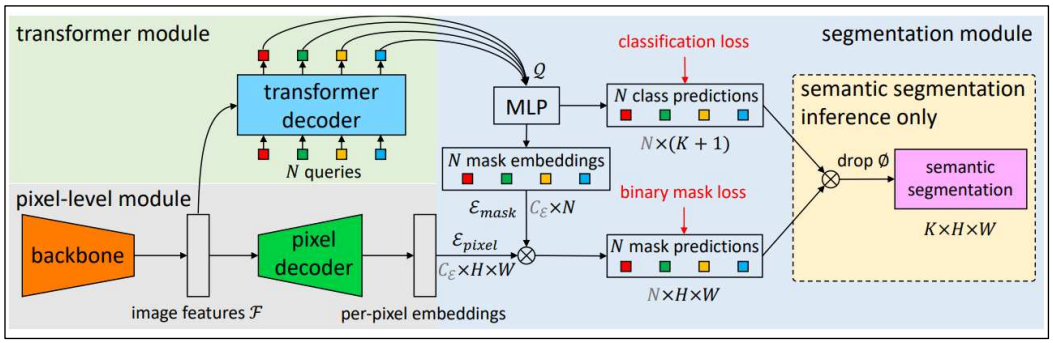

Рисунок 1.8 – Архітектура MaskFormer

Ця запропонована архітектура демонструє високу ефективність та сумісність з різними backbone-мережами. Наприклад, порівняно з DeepLabV3+, MaskFormer потребує меншої кількості параметрів при використанні того самого backbone, досягаючи при цьому вищої точності. До недоліків належить потреба у значних обсягах навчальних даних та ретельний підбір параметрів, оскільки трансформери мають тенденцію досягати кращих результатів при навчанні на великих обсягах даних.

Архітектура SegFormer [56] є моделлю семантичної сегментації типу енкодер-декодер. Її кодер реалізований як ієрархічний трансформер, який обробляє зображення фрагментами 4×4, поступово зменшуючи просторову роздільну здатність та генеруючи багаторівневі ознаки. Така конструкція забезпечує здатність обробляти вхідні зображення різних розмірів без необхідності позиційного кодування (рис. 1.9).

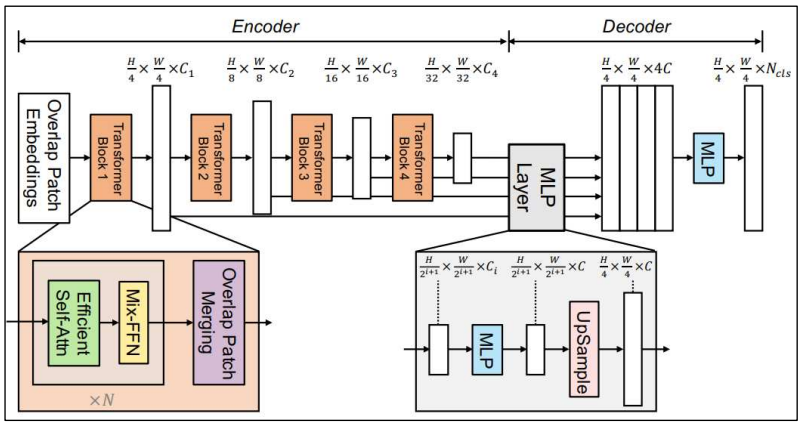

Рисунок 1.9 – Архітектура SegFormer: енкодер та декодер

Основна увага архітектури SegFormer зосереджена на її мінімалістичному декодері, який складається виключно з багатошарового перцептрона (MLP) та шарів підвищення дискретизації, що суттєво знижує обчислювальну складність. Декодер агрегує карти ознак, створені енкодером, вирівнюючи їх до уніфікованої роздільної здатності для отримання кінцевого результату.

Mask2Former [57] є розвитком концепцій, представлених у MaskFormer. Основна архітектурна інновація полягає у заміні cross-attention на masked-attention та додаванні карт ознак різних роздільних здатностей до піксельного декодера, подібно до Feature Pyramid Network (FPN). Masked attention дозволяє моделі ефективніше фокусуватися на об’єктах або областях інтересу, що прискорює процес збіжності (рис. 1.10).

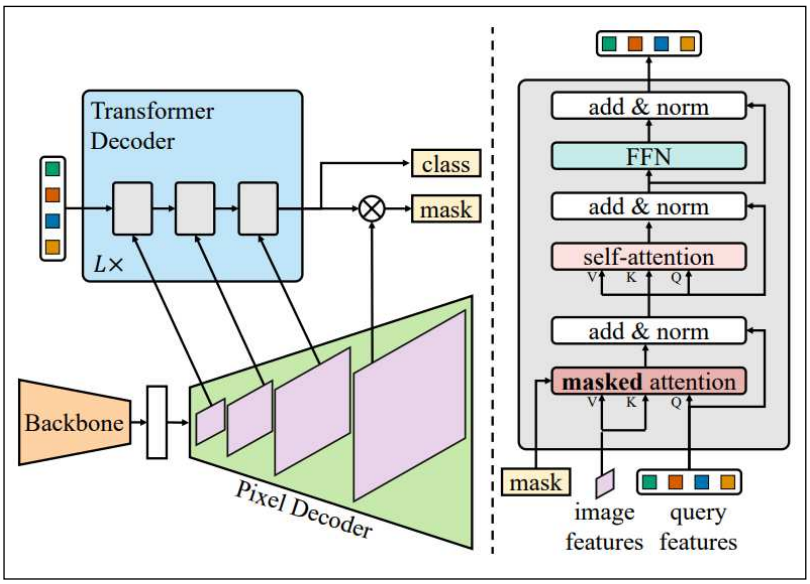

Рисунок 1.10 – Архітектура Mask2Former з masked-attention

Використання мультимасштабних ознак високої роздільної здатності у піксельному декодері дозволяє моделі сегментувати малі об’єкти або регіони. Для оптимізації використання пам’яті під час навчання автори запропонували підхід, в якому зіставляється лише випадково обрана підмножина точок ground-truth маски, а не всі пікселі.

OneFormer [58] є логічним продовженням архітектур MaskFormer та Mask2Former, пропонуючи уніфікований підхід до різних задач сегментації, включаючи семантичну, екземплярну та паноптичну сегментацію в рамках однієї моделі. Ключовою відмінністю OneFormer є використання task-conditional токена, який визначає тип виконуваної задачі (рис. 1.11).

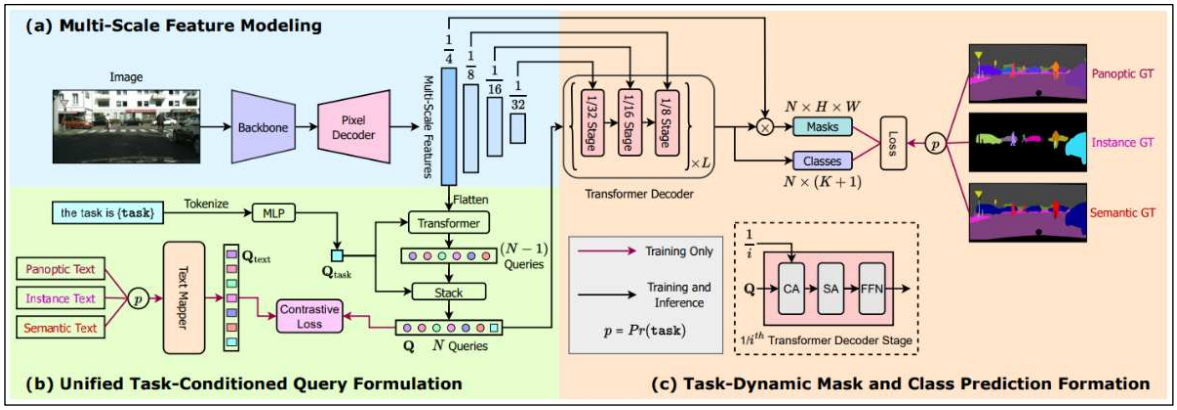

Рисунок 1.11 – Архітектура OneFormer: уніфікований підхід до сегментації

Архітектура моделі складається з трьох основних компонентів: енкодера зображень на основі Swin Transformer [59], task-токена та декодера з механізмами уваги на декількох рівнях. Процес сегментації в OneFormer охоплює виділення візуальних ознак з вхідного зображення, додавання task-токена, генерацію об'єктних запитів у декодері залежно від типу задачі та прогнозування бінарних масок разом з відповідними класами.

Таблиця 1.2 – Порівняння архітектур для сегментації

Архітектура	**Тип**	**Особливості**	**Недоліки**
FCN	Семант.	Перша end-to-end	Грубі межі
U-Net	Семант.	Skip-зв'язки	Обмежене рецептивне поле
DeepLabV3+	Семант.	ASPP module	Повільна
Mask R-CNN	Екземпл.	2-етапна (RoI)	Ресурсоємна
MaskFormer	Універс.	Query-based	Вимоглива до даних
Mask2Former	Універс.	Masked-attn	Складна у навчанні

На рисунку 1.12 систематизовані моделі за класом та часом появи.

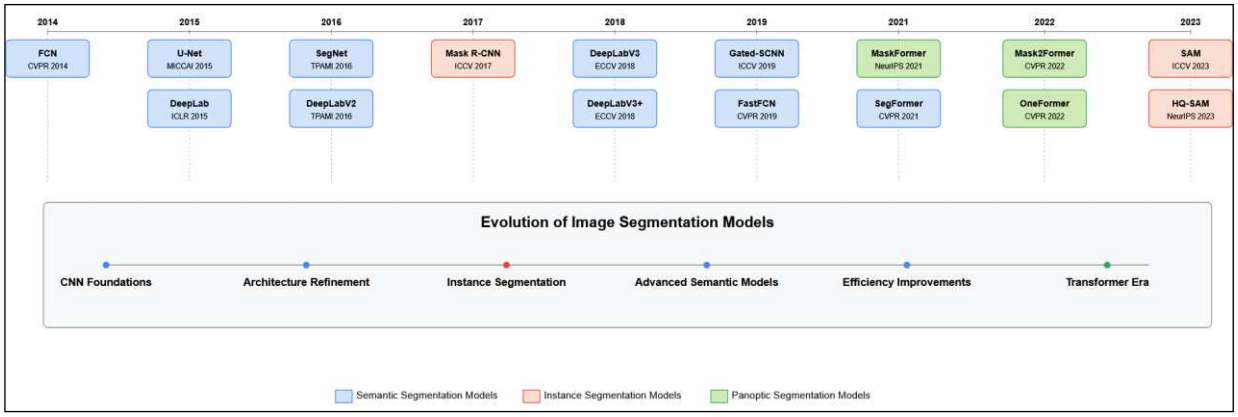

Рисунок 1.12 – Систематизація моделей за часом появи та типом

1.3 Аналіз особливостей open-set та open-vocabulary підходів

Розглянуті вище методи були навчені на фіксованому наборі даних і потребують додаткового навчання при зміні формату або типу вхідних даних, наприклад, при додаванні нового класу для пошуку. Для того щоб модель була стійкішою до змін, її слід навчати на великому варіативному наборі даних, що робить процес навчання тривалим та витратним.

Класичні методи сегментації з фіксованою кількістю класів (closed-set) досягли вражаючих успіхів. Такий підхід задовольняє більшість потреб, і методи стають state-of-the-art на загальновизнаних наборах даних для порівняння. Проте архітектурна залежність від фіксованої кількості класів обмежує їх повторне використання у реальному світі, де варіацій об’єктів для пошуку безліч. Вони можуть взаємодіяти між собою, мати просторово-семантичні зв’язки, які моделі необхідно враховувати.

Основними обмеження closed-set підходів є:

1) неможливість розпізнавання об’єктів, що не входили до навчальної вибірки;

2) необхідність повторного навчання при додаванні нових класів;

3) відсутність гнучкості у визначенні атрибутів та характеристик об’єктів;

4) неможливість врахування контекстних зв’язків між об’єктами.

Такі задачі зумовили необхідність переходу до парадигми open-set методів, де кількість класів не є фіксованою і може бути заданою текстовим чи іншим мультимодальним запитом. На відміну від класичного closed-set підходу, open-set дозволяє враховувати атрибути, дії або часткову належність. Така гнучкість саме дозволяє враховувати просторово-семантичний контекст.

Zero-shot learning (ZSL) – це підхід у машинному навчанні, при якому модель може вирішувати задачі, пов'язані з категоріями або завданнями, для яких вона не була явно навчена [60, 61]. Це означає, що модель здатна розпізнавати об'єкти, що належать до нових класів, які не зустрічалися у навчальних даних, але на основі додаткової інформації модель може це зрозуміти [62].

Few-shot learning (FSL) – підхід, подібний до ZSL, спрямований на забезпечення здатності моделі розпізнавати раніше невідомі об'єкти шляхом навчання на невеликій кількості даних [63-65].

Open-vocabulary підхід оперує необмеженим словником природної мови [66-69]. Це дає можливість формулювати запити у довільній формі, де атрибути або ознаки формуються з вже відомих лексичних одиниць, забезпечуючи гнучкість та адаптацію системи до невідомих раніше об'єктів.

Слід зазначити, що текстовий опис не завжди здатен вичерпно передати усі характеристики об'єкта, такі як специфічна текстура, відтінок або складна геометрія. У таких випадках мультимодальний підхід має передбачати можливість використання візуального зразка (visual prompt) для пошуку.

Мультимодальне навчання є парадигмою машинного навчання, що передбачає інтеграцію інформації з різних модальностей (текст, зображення, аудіо, відео) для створення якісніших та гнучких моделей [70-72]. У контексті сегментації зображень за текстовим запитом мультимодальний підхід дозволяє поєднувати візуальну інформацію із семантичним змістом текстових описів.

З розвитком методів обробки природної мови (NLP) багато підходів були успішно адаптовані до завдань комп'ютерного зору, дозволяючи більш

інтуїтивно описувати сцени за допомогою природної мови. На відміну від традиційних моделей, обмежених фіксованим набором класів, підходи на основі NLP дозволяють шукати об’єкти на основі атрибутів, що розширює їх застосування [73-76].

Архітектура типової мультимодальної моделі для сегментації складається з:

1) візуальний енкодер – витягує ознаки з вхідного зображення;
2) текстовий енкодер – обробляє текстовий запит;
3) модуль злиття – поєднує візуальні та текстові ознаки;
4) декодер – генерує фінальну маску сегментації.

Основні переваги open-vocabulary підходу:

1) можливість пошуку об’єктів за довільним текстовим описом;
2) врахування атрибутів, властивостей та контекстних зв’язків;
3) відсутність необхідності перенавчання при появі нових категорій;
4) природна інтеграція з мультимодальними системами.

Такий підхід реалізує модель SAM [12]. Її метою є надання універсального рішення для сегментації об’єктів на зображеннях, незалежно від їх категорії. Це стало можливим завдяки навчанню моделі на величезному наборі даних SA-1B, що містить понад 1 мільярд масок.

Архітектура SAM складається з трьох основних компонентів:

1) image encoder – енкодер зображень на основі модифікованого ViT, що отримує зображення 1024×1024 на вхід;
2) prompt encoder – енкодер запитів, здатний обробляти різні типи вхідних даних (текст, точки, обмежувальні рамки, маски);
3) mask decoder – легкий декодер для генерації масок сегментації (рис. 1.13).

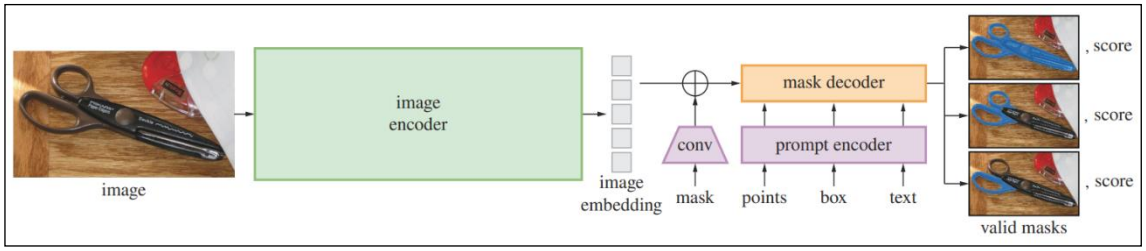

Рисунок 1.13 – Архітектура SAM: image encoder, prompt encoder та mask decoder

Спочатку зображення конвертується в embeddings за допомогою image encoder. Потім до цих даних додається текстовий опис об’єктів, що шукаються, або їх розташування, і за допомогою легкого декодера отримуються маски об’єктів. Також були розроблені модифікації SAM для вирішення проблем неточного виділення меж об’єктів та швидкості виконання:

HQ-SAM [77] – покращена версія SAM, розроблена для підвищення якості передбачуваних масок, особливо в контексті точного виділення меж об’єктів. Хоча SAM навчався на 1B масок, результати сегментації можуть бути неоптимальними при роботі з об’єктами зі складною структурою. HQ-SAM вирішує дві основні проблеми оригінального SAM: неточні межі масок для малих або тонких об’єктів та некоректні прогнозування масок або фрагментовані результати у складних сценах.

FastSAM [20] – оптимізована версія SAM, спрямована на підвищення швидкості виконання. FastSAM використовує архітектуру на основі YOLOv8 [78] для швидкої генерації масок-кандидатів, які потім фільтруються відповідно до запиту.

Grounded-SAM [16] – двоетапний метод, де спочатку об’єкти виділяються з текстового опису за допомогою GroundingDINO [79], а їх розташування потім передається до SAM для отримання масок.

Таблиця 1.3 – Порівняння моделей сімейства SAM

Модель	**Параметри**	**Час виконання**	**Особливості**
SAM ViT-B	362,1M	0,101s	Базова модель
HQ-SAM ViT-B	358M	0,11s	Покращена якість меж
FastSAM	72M	0,04s	Висока швидкість
Grounded-SAM	590,3M	0,269s	Текстовий пошук

На цей момент не існує реалізації SAM, де об’єкти можна було б задавати безпосередньо текстовим запитом в рамках одноетапного підходу. У двоетапному методі Grounded-SAM об’єкти спочатку виділяються з текстового опису за допомогою GroundingDINO, розташування яких потім передається до SAM для отримання масок.

1.4 Постановка завдання дослідження

На основі проведеного аналізу стану сегментації зображень за текстовим запитом виявлено такі ключові проблеми, що потребують вирішення.

1) Обмеженість сучасних архітектур, а саме відсутність одноетапної open-set моделі для сегментації екземплярів. Наразі не існує одноетапної open-set моделі для вирішення задачі сегментації екземплярів за текстовим запитом. Існуючі рішення, такі як Grounded-SAM, є двоетапними, що збільшує час виконання та обчислювальні витрати.

2) Проблеми з постобробкою. Методи bottom-up потребують ефективних алгоритмів постобробки для групування пікселів в окремі екземпляри. Існуючі підходи не завжди дають задовільні результати.

Метою дисертаційної роботи є розробка методів та моделей мультимодальної просторово-семантичної сегментації за текстовим запитом.

Для досягнення поставленої мети необхідно вирішити нижченаведені завдання:

1) провести аналіз сучасних методів та моделей сегментації зображень, зокрема open-set та open-vocabulary підходів, та визначити напрямки їх удосконалення;

2) розробити модифіковану архітектуру з інтеграцією підходів для прогнозування центрів об’єктів та відстаней до меж обмежувальних рамок;

3) розробити метод постобробки результатів висхідної сегментації для розділення знайдених об’єктів на окремі екземпляри;

4) дослідити та обґрунтувати зважену метрику для порівняння розбиттів зображень з доведенням аксіом метричного простору.

Визначено обмеження для запропонованого підходу:

1) текстовий запит для пошуку об’єктів має описувати об’єкти, що належать до однієї шуканої категорії;

2) запит не повинен містити суперечливих ознак;

3) якщо необхідно знайти об’єкти різних семантик, слід використовувати декілька окремих запитів;

4) мінімальний розмір об’єкта для детекції становить 100 пікселів² після масштабування до робочої роздільної здатності моделі.

1.5 Висновки за розділом

У першому розділі проведено комплексний аналіз стану проблеми сегментації зображень за текстовим запитом. За результатами аналізу сформульовано такі висновки:

1) Проаналізовано типи сегментації зображень (семантична, екземплярна, паноптична) та класифіковано підходи до сегментації екземплярів за критеріями кількості етапів обробки (одноетапні, двоетапні) та напрямку обробки (top-down, bottom-up). Встановлено, що методи bottom-up

потенційно мають переваги у швидкості, але потребують ефективних алгоритмів постобробки.

2) Проведено порівняльний аналіз архітектур для семантичної сегментації та сегментації екземплярів. Розглянуто еволюцію від FCN та U-Net до сучасних архітектур на основі трансформерів (MaskFormer, Mask2Former, OneFormer). Виявлено тенденцію до інтеграції механізмів уваги та уніфікації підходів до різних типів сегментації.

3) Досліджено особливості open-set та open-vocabulary підходів. Показано, що парадигма open-set дозволяє подолати обмеження closed-set методів щодо фіксованого набору класів.

4) Проаналізовано модель SAM та її модифікації (HQ-SAM, FastSAM, Grounded-SAM). Встановлено, що на сьогодні не існує одноетапної реалізації SAM з можливістю визначення об’єктів безпосередньо текстовим запитом.

5) На основі проведеного аналізу сформульовано мету та завдання дисертаційного дослідження, визначено об’єкт та предмет дослідження, окреслено наукову новизну та практичне значення очікуваних результатів.

2 ФУНКЦІЇ ВТРАТ ДЛЯ ВИРІШЕННЯ РІЗНИХ ТИПІВ ЗАДАЧ СЕГМЕНТАЦІЇ

2.1 Аналіз відомих функцій втрат

У попередньому розділі було проаналізовано еволюцію архітектур нейронних мереж для сегментації – від простих повнозгорткових моделей до сучасних архітектур на основі трансформерів. Проте разом з еволюцією архітектур, функції втрат також пройшли власну еволюцію, оскільки вибір функції втрат є не менш важливим аспектом для ефективного розв'язання задачі та навчання моделі [80, 81].

Функція втрат – це функція, призначена для обчислення похибки між передбаченими значеннями, згенерованими моделлю, та еталонними значеннями цільової змінної. Основна мета процесу навчання моделі полягає у мінімізації значення функції втрат шляхом ітеративного коригування параметрів моделі. Формально, якщо f_θ – модель з параметрами θ, x – вхідні дані, y – еталонні значення, то функція втрат $\mathcal{L}$ визначає величину:

$$\mathcal{L}(f_\theta(x), y) \to \min_\theta .$$

Для сегментації зображень вибір функції втрат має особливе значення, оскільки необхідно враховувати специфіку попіксельного прогнозування. На відміну від класичної класифікації, де оцінюється належність всього зображення до певного класу, сегментація вимагає коректного присвоєння мітки кожному пікселю, що суттєво збільшує розмірність задачі оптимізації.

Крім того, сегментація часто характеризується значним дисбалансом класів – об'єкти інтересу можуть займати лише невелику частину зображення, тоді як фон домінує. Така особливість вимагає спеціальних підходів до конструювання функцій втрат, що враховують цей дисбаланс.

Для задач комп'ютерного зору виділимо такі підкласи функцій втрат: функції на основі розподілів (distribution-based) – для задач класифікації; функції на основі областей (region-based) – для оцінки відповідності областей сегментації; функції на основі меж (boundary-based) – для точного виділення контурів об'єктів; функції на основі векторного представлення (embedding-based) – для метричного навчання та представлення ознак.

Взаємозв'язок між архітектурою мережі та функцією втрат є двостороннім. З одного боку, архітектура визначає формат виходу моделі та накладає обмеження на тип функції втрат. З іншого боку, вибір функції втрат може впливати на архітектурні рішення, наприклад, необхідність додаткових компонент самої моделі. Архітектури, розглянуті у першому розділі (Mask R-CNN, MaskFormer, SAM тощо), використовують комбінації різних функцій втрат для одночасної оптимізації класифікації, локалізації та сегментації.

Функції втрат на основі розподілів використовуються в задачах, де необхідно віднести зображення або його частини до заздалегідь визначених класів. Ці функції оцінюють розбіжність між передбаченим розподілом ймовірностей класів та істинним значенням. У контексті сегментації вони застосовуються для попіксельної класифікації, де кожен піксель розглядається як окремий зразок для класифікації.

Cross-entropy loss є фундаментальною функцією втрат для задач класифікації. Вона вимірює розбіжність між двома розподілами ймовірностей та базується на концепції ентропії з теорії інформації. Для бінарної класифікації cross-entropy визначається як:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right],$$

де $y_i \in 0,1$ – істинна мітка для i-го пікселя, $\hat{y}_i \in (0,1)$ – передбачена ймовірність приналежності до позитивного класу, N – загальна кількість пікселів у зображенні.

Для багатокласової класифікації, що є типовою для семантичної сегментації, формула узагальнюється:

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}),$$

де C – кількість класів, $y_{i,c}$ – індикатор приналежності i-го пікселя до класу c (one-hot кодування), $\hat{y}_{i,c}$ – передбачена ймовірність приналежності i-го пікселя до класу c.

Градієнт cross-entropy відносно прогнозування має просту форму:

$$\frac{\partial L_{CE}}{\partial \hat{y}i,c} = -\frac{yi,c}{\hat{y}_{i,c}},$$

що забезпечує стабільне навчання. Проте при комбінації з softmax-активацією градієнт набуває ще простішої форми: $\hat{y}_{i,c} - y_{i,c}$, що робить цю комбінацію особливо ефективною для оптимізації.

Незважаючи на широке використання, cross-entropy стикається з суттєвими проблемами при роботі з незбалансованими наборами даних, що є типовою ситуацією для задач сегментації. Коли один клас (наприклад, фон) домінує над іншими, модель може навчитися тривіальному рішенню – передбачати домінуючий клас для всіх пікселів, що мінімізує функцію втрат, але не є корисним результатом. Це призвело до розробки модифікацій cross-entropy.

Balanced Cross-Entropy вводить коефіцієнти, що враховують частоту появи кожного класу в наборі даних:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \cdot y_{i,c}\log(\hat{y}_{i,c}),$$

де вага класу визначається як $w_c = \frac{N}{C \cdot n_c}$, а n_c – кількість пікселів класу c у навчальній вибірці. Таким чином, рідкісні класи отримують більші ваги, що компенсує їх меншу представленість. Альтернативний підхід до визначення ваг використовує обернену частоту: $w_c = \frac{1}{n_c}$ з подальшою нормалізацією.

Weighted Cross-Entropy розширює цю концепцію, вводячи ваги для різних класів незалежно від їх частоти:

$$L_{WCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\alpha_c \cdot y_{i,c}\log(\hat{y}_{i,c}),$$

де α_c – задані вручну ваги для кожного класу, що визначаються експертом на основі важливості кожного класу для конкретної задачі. Це забезпечує додаткову гнучкість для специфічних задач, наприклад, коли помилка на певному класі є критичнішою за інші.

Focal Loss [82] була запропонована для вирішення проблеми дисбалансу класів у задачах детекції об’єктів і відтоді широко застосовується у сегментації. Основна ідея полягає у зменшенні внеску правильних відповідей з високою ймовірністю (легких прикладів), дозволяючи моделі концентруватися на складних випадках:

$$L_{FL} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\alpha_c(1-\hat{y}_{i,c})^{\gamma} \cdot y_{i,c}\log(\hat{y}_{i,c}),$$

де $\gamma \geq 0$ – параметр фокусування, що контролює швидкість зниження ваги для коректно класифікованих прикладів. При $\gamma = 0$ Focal Loss еквівалентна стандартній Weighted Cross-Entropy. Типові значення γ лежать в діапазоні від 1 до 5, з оптимальним значенням близько 2 для багатьох задач.

Модулювальний фактор $(1-\hat{y}_{i,c})^{\gamma}$ має важливу властивість. Коли приклад класифіковано правильно з високим рівнем упевненості $(\hat{y}_{i,c} \to 1)$, то цей фактор наближається до нуля, суттєво зменшуючи внесок такого прикладу. Навпаки, для складних прикладів з низькою впевненістю модулювальний фактор залишається близьким до одиниці.

Порівняння функцій втрат на основі розподілів за ключовими характеристиками наведено у таблиці 2.1.

Таблиця 2.1 – Порівняння функцій втрат на основі розподілів

Назва	**Стійкість**	**Диф-сть**	**Складність**	**Чутливість**	**Параметри**
Cross-Entropy (CE)	Низька	Так	Низька	Висока	Ні
Balanced CE	Висока	Так	Низька	Середня	Ні (або ваги класів)
Weighted CE	Висока	Так	Низька	Середня	Так (α)
Focal Loss	Дуже висока	Так	Середня	Контрольована (γ)	Так (α, γ)

На відміну від функцій на основі розподілів, які оцінюють кожен піксель незалежно, функції на основі областей розглядають сегментацію як цілісну структуру та оцінюють ступінь перекриття між передбаченою та еталонною областями. Такий підхід є більш природним, оскільки безпосередньо оптимізує метрики якості сегментації.

Dice Loss (також відома як F1 Loss або Sørensen–Dice Loss) [17] спроектована для максимізації коефіцієнта Дайса між двома множинами –

результуючою маскою сегментації та еталонною множиною. Коефіцієнт Дайса визначається як:

$$DSC = \frac{2 | X \cap Y |}{| X | + | Y |},$$

де X – множина пікселів передбаченої маски, Y – множина пікселів еталонної маски. Відповідно, Dice Loss визначається як:

$$L_{Dice} = 1 - DSC = 1 - \frac{2 | X \cap Y |}{| X | + | Y |}.$$

Для диференційовної реалізації з м'якими прогнозуваннями (soft predictions) формула набуває вигляду:

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + \epsilon},$$

де $\hat{y}_i \in [0,1]$ – передбачена ймовірність для i-го пікселя, $y_i \in \{0,1\}$ –еталонне значення, ϵ – мала константа для запобігання ділення на нуль (типово 10^{-5}).

Ключовою перевагою Dice Loss є її інваріантність до дисбалансу класів [83]. Оскільки функція оцінює відносне перекриття, а не абсолютну кількість правильних передбачень, вона природно враховує розмір об'єкта. Це робить Dice Loss особливо популярною у медичній сегментації, де об'єкти інтересу часто займають малу частку зображення.

Проте градієнти Dice Loss є нестабільними, особливо коли перетин передбаченої та істинної масок дуже малий. Градієнт відносно x_i має вигляд:

$$\frac{\partial L_{Dice}}{\partial x_i} = -\frac{2y_i(\sum_j x_j + \sum_j y_j) - 2\sum_j x_j y_j}{(\sum_j x_j + \sum_j y_j)^2}.$$

При малих значеннях знаменника градієнт може стати дуже великим, що призводить до нестабільності навчання. Також можливе вибухове зростання градієнтів на початкових етапах навчання, коли модель ще не навчилася виділяти об’єкти.

Для подолання недоліків негладкості та проблем з градієнтами була запропонована комбінована функція Log-Cosh Dice Loss [84]:

$$L_{LogCoshDice} = \log(\cosh(L_{Dice})).$$

Функція $\log(\cosh(x))$ апроксимує |x| для великих значень x, але є гладкою при $x = 0$, що забезпечує більш стабільні градієнти. Це особливо корисно на початкових етапах навчання, коли значення Dice Loss є високими.

Tversky Loss [21] є узагальненням Dice Loss, що додає можливість контролювати баланс між помилками false positives (FP) та false negatives (FN):

$$L_{Tversky} = 1 - \frac{|X \cap Y|}{|X \cap Y| + \alpha |X \setminus Y| + \beta |Y \setminus X|},$$

де α та β – параметри, що контролюють штраф за FP та FN відповідно. При $\alpha = \beta = 0.5$ Tversky Loss еквівалентна Dice Loss. При $\alpha = \beta = 1$ отримуємо Jaccard Loss (IoU Loss).

Гнучкість Tversky Loss дозволяє налаштовувати функцію під конкретну задачу:

1) при $\alpha > \beta$ модель штрафується більше за false positives, що корисно, коли важливо уникнути хибних спрацьовувань;

2) при $\alpha < \beta$ більший штраф за false negatives, що важливо для задач, де пропуск об'єкта є критичнішим за хибне спрацьовування.

Для м'яких передбачень формула набуває вигляду:

$$L_{Tversky} = 1 - \frac{\sum_i x_i y_i + \epsilon}{\sum_i x_i y_i + \alpha \sum_i x_i (1 - y_i) + \beta \sum_i (1 - x_i) y_i + \epsilon}.$$

Варто зазначити, що Tversky Loss може бути нестабільною на початковій фазі навчання, особливо при екстремальних значеннях параметрів α та β.

Focal Tversky Loss поєднує ідеї Focal Loss та Tversky Loss для підвищення уваги до складних прикладів:

$$L_{FocalTversky} = (1 - TI)^{\gamma},$$

де TI – індекс Тверського, γ – параметр фокусування. Ця модифікація особливо ефективна для сегментації малих об'єктів.

IoU Loss (Jaccard Loss) [18] є поширеною метрикою для оцінки якості сегментації, яка також може використовуватись як функція втрат. Intersection over Union (IoU) визначається як:

$$IoU = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}.$$

Відповідно, IoU Loss:

$$L_{IoU} = 1 - IoU = 1 - \frac{\sum_i x_i y_i}{\sum_i x_i + \sum_i y_i - \sum_i x_i y_i}.$$

Проте IoU має проблему при відсутності перетину – функція стає розривною, коли $|X \cup Y| = 0$. Для диференціювання необхідно застосувати метод диференційовної апроксимації, наприклад, додається згладжувальний параметр ϵ або використовується прийом log-sum-exp для покращення чисельної стабільності.

Для подолання недоліків стандартного IoU були запропоновані модифікації, спочатку розроблені для задач детекції, але застосовні і до сегментації:

Generalized IoU (GIoU) [85] розширює стандартний IoU, враховуючи випадки, коли передбачена та еталонна області не перетинаються:

$$GIoU = IoU - \frac{|C \setminus (X \cup Y)|}{|C|},$$

де C – найменша опукла область, що містить обидві області X та Y. Другий член штрафує за «порожній простір» між областями, що забезпечує градієнт навіть при відсутності перетину.

Distance IoU (DIoU) [86] додатково враховує відстань між центрами передбаченої та еталонної областей:

$$DIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2},$$

де ρ – евклідова відстань, b та b^{gt} – центроїди передбаченої та еталонної областей, c – діагональ найменшого обмежувального прямокутника.

Complete IoU (CIoU) [87] додатково враховує узгодженість форми:

$$CIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v,$$

де v вимірює узгодженість форми (наприклад, співвідношення сторін для прямокутників), α – балансувальний параметр.

Lovász-Softmax Loss [22] пропонує принципово інший підхід до оптимізації IoU. Оскільки IoU є почастинно-лінійною функцією від бінарних передбачень, її неможливо безпосередньо оптимізувати градієнтними методами. Lovász extension дозволяє отримати опуклу оболонку будь-якої функції на множині підмножин, що робить можливою градієнтну оптимізацію.

Для функції втрат Jaccard

$$\Delta_J(e) = 1 - \frac{|Y|}{|Y| + e},$$

де e – вектор помилок.

Lovász extension визначається як:

$$\overline{\Delta_J}(m) = \sum_{i=1}^{p} m_{\pi(i)} (g_{\pi(i)} - g_{\pi(i-1)}),$$

де π – перестановка, що сортує m за спаданням, $g_i = \Delta_J(e_{\pi(1)}, ..., e_{\pi(i)})$.

Для багатокласової сегментації Lovász-Softmax Loss обчислюється як середнє по класах:

$$L_{Lovasz} = \frac{1}{|C|} \sum_{c \in C} \overline{\Delta_{J_c}}(m(c)).$$

Перевагою Lovász-Softmax є пряма оптимізація IoU, що часто призводить до кращих результатів на цій метриці. Проте обчислювальна складність є вищою через необхідність сортування.

Порівняння регіон-базованих функцій втрат наведено у таблиці 2.2.

Таблиця 2.2 – Порівняння регіон-базованих функцій втрат

Назва	**Стійкість**	**Диф-сть**	**Складність**	**Чутливість**	**Контроль FP/FN**
Dice Loss	Висока	Так	Низька	Середня	Умовний
Tversky Loss	Дуже висока	Так	Низька	Контрольована (α, β)	Так
IoU Loss	Середня	Часткова	Середня	Середня	Ні
Lovász-Softmax	Середня	Так	Висока	Висока	Так

У той час як функції на основі областей фокусуються на оцінці загальної відповідності областей сегментації, функції на основі меж зосереджуються на точності меж об'єктів. Це особливо важливо у багатьох практичних задачах, де точне виділення контурів є критичним – наприклад, у медичній візуалізації для планування хірургічних втручань або в автономному водінні для точного визначення меж дороги.

Boundary Loss [88] була запропонована для вирішення проблеми сегментації сильно незбалансованих даних, де об'єкт інтересу займає малу частку зображення. Основна ідея полягає у формулюванні функції втрат через інтеграл по межі, а не по області:

$$L_{Boundary} = \int_{\Omega} \phi_G(q) s_\theta(q) dq ,$$

де Ω – простір зображення, ϕ_G – карта рівнів (level set) функція для істинної межі, $s_\theta(q)$ – softmax вихід мережі для пікселя q.

Карта рівнів ϕ_G визначається як знакова функція відстані до межі:

$$\phi_G(q) = \begin{cases} -D_G(q), & \text{якщо } q \text{ всередині } G \\ D_G(q), & \text{якщо } q \text{ зовні } G \end{cases},$$

де $D_G(q)$ – евклідова відстань від пікселя q до найближчої точки межі ∂G. На практиці для дискретних зображень це апроксимується як:

$$L_{Boundary} = \sum_{q \in \Omega} s_\theta(q) \cdot \phi_G(q).$$

Boundary Loss часто використовується в комбінації з функціями на основі областей, такими як Dice Loss:

$$L_{combined} = \alpha L_{Dice} + (1-\alpha) L_{Boundary}.$$

Параметр α зазвичай зменшується протягом навчання – на початку домінує функція на основі областей для забезпечення грубої локалізації, а потім Boundary Loss уточнює межі.

Hausdorff Distance Loss [89] базується на відстані Гаусдорфа – класичній метриці для порівняння форм у геометрії. Відстань Гаусдорфа визначається як максимум з двох односторонніх відстаней:

$$d_H(X,Y) = \max h(X,Y), h(Y,X).$$

Де одностороння відстань Гаусдорфа:

$$h(X,Y) = \max_{x \in \partial X} \min_{y \in \partial Y} |x - y|,$$

де ∂X та ∂Y – межі областей X та Y.

Відстань Гаусдорфа вимірює найгірший випадок невідповідності меж – максимальну відстань від будь-якої точки однієї межі до найближчої точки

іншої. Це робить її чутливою до локальних відхилень, що може бути як перевагою (гарантія точності), так і недоліком (чутливість до викидів).

Оскільки класична відстань Гаусдорфа не є диференційовною через операції max та min, для використання її як функцію втрат потрібна апроксимація. Одна з поширених апроксимацій:

$$L_{Hausdorff} = \frac{1}{|\partial X|}\sum_{x\in\partial X} d_Y^{\alpha}(x) + \frac{1}{|\partial Y|}\sum_{y\in\partial Y} d_X^{\alpha}(y),$$

де $d_Y(x) = \min_{y\in\partial Y}|x-y|$ – відстань від точки x до найближчої точки межі Y, $\alpha-$ параметр, що контролює чутливість до викидів. При $\alpha = 2$ отримуємо середньоквадратичну відстань.

Для ефективного обчислення використовується попередньо розрахована карта відстаней (distance transform), що дозволяє обчислити $d_Y(x)$ для всіх пікселів за лінійний час.

Active Contour Loss [90] базується на класичній теорії активних контурів (snakes), адаптованій для глибокого навчання. Функція містить як регіональні, так і граничні компоненти:

$$L_{AC} = \lambda_{length} L_{length} + \lambda_{region} L_{region}.$$

Компонент довжини штрафує за надмірну довжину контуру:

$$L_{length} = \int |\nabla H_{\epsilon}(\phi)|,$$

де H_{ϵ} – згладжена функція Гевісайда, ϕ – функція рівня, що представляє контур.

Регіональний компонент базується на моделі Чана-Везе [91]:

$$L_{region} = \int (I - c_1)^2 H_\epsilon(\phi) + \int (I - c_2)^2 (1 - H_\epsilon(\phi)),$$

де I – інтенсивність зображення, c_1 та c_2 – середні інтенсивності всередині та зовні контуру.

Порівняння функцій втрат на основі меж наведено у таблиці 2.3.

Таблиця 2.3 – Порівняння boundary-based функцій втрат

Назва	**Стійкість (дисб.)**	**Диф-сть**	**Складність**	**Чутливість (межі)**	**Стійкість (шум)**
Boundary Loss	Висока	Так	Середня	Середня	Висока
Hausdorff Loss	Середня	Часткова	Висока	Висока	Низька
Active Contour	Висока	Так	Середня	Низька	Висока

Функції втрат на основі векторного представлення працюють з векторними представленнями (embeddings) об’єктів, розміщуючи їх у багатовимірному просторі таким чином, щоб семантично близькі об’єкти були ближче один до одного, а різні об’єкти – віддалені один від одного. Цей підхід застосовується у representation learning – парадигмі машинного навчання, спрямованій на отримання корисних ознак з даних або їх трансформацію в інший простір ознак, де певні характеристики стають більш явними.

У контексті сегментації embedding-based підходи використовуються для групування пікселів, що належать одному об’єкту. Замість прямого прогнозування мітки класу, мережа генерує векторне представлення для кожного пікселя, а потім пікселі групуються на основі близькості їх представлення у просторі ознак.

Triplet Loss [92] є функцією втрат, яка навчає мережу формувати простір ознак, в якому об’єкти одного класу розташовані ближче один до одного, а об’єкти різних класів – далі один від одного. Навчання базується на триплетах – наборах з трьох елементів:

1) якір – опорний зразок;

2) позитивний приклад – зразок того ж класу, що й позитивний приклад;

3) негативний приклад – зразок іншого класу.

Функція втрат визначається як:

$$L_{Triplet} = \max(0, d(a, p) - d(a, n) + m),$$

де a, p, n – якір, позитивний та негативний зразки відповідно, d – функція відстані (зазвичай евклідова або косинусна), $m > 0$ – відступ (margin), що визначає мінімальну різницю між відстанями.

Функція мінімізує відстань між якорем та позитивним прикладом, одночасно максимізуючи відстань між якорем та негативним прикладом. Відступ m забезпечує, що негативні приклади будуть не просто далі за позитивні, а далі на певну величину, що покращує розділення класів.

Вирішальне значення для ефективності навчання має відбір триплетів. До основних стратегій цього процесу належать:

1) random sampling – випадковий вибір триплетів;

2) hard negative mining – вибір найскладніших негативних прикладів (найближчих до anchor);

3) semi-hard negative mining – вибір негативних прикладів, які далі за позитивний, але ближче за margin.

У контексті сегментації triplet loss може використовуватися для навчання векторного представлення пікселів, де пікселі одного екземпляра

об’єкта є позитивними прикладами один для одного, а пікселі різних екземплярів – негативними.

Center Loss [93] є технікою регуляризації, спрямованою на генерацію компактних представлень класів у просторі ознак. На відміну від Triplet Loss, яка працює з парами/триплетами зразків, center loss мінімізує відстань кожного зразка до центру його класу:

$$L_{Center} = \frac{1}{2}\sum_{i=1}^{N} |x_i - c_{y_i}|_2^2,$$

де x_i – векторне представлення зразка i, c_{y_i} – центр класу y_i, N – кількість зразків у пакеті.

Центри класів c_k оновлюються протягом навчання як ковзне середнє векторного представлення відповідного класу:

$$c_k^{(t+1)} = c_k^{(t)} - \eta \cdot \Delta c_k,$$

де $\Delta c_k = \frac{\sum_{i:y_i=k} (c_k - x_i)}{1 + |i : y_i = k|}$, η – швидкість оновлення центрів.

Center Loss зазвичай комбінується з класифікаційною функцією втрат:

$$L = L_{Softmax} + \lambda L_{Center},$$

де λ – балансувальний коефіцієнт (типово $\lambda \in [0.001, 0.1]$). Softmax loss забезпечує розділення класів, а center loss – компактність представлень всередині класу.

Contrastive Loss [94] є ще однією популярною функцією основі векторного представлення, що працює з парами зразків:

$$L_{Contrastive} = \frac{1}{2N}\sum_{n=1}^{N}(y \cdot d^2 + (1-y)\cdot \max(0, m-d)^2),$$

де $y \in 0,1$ – мітка пари (1 – однаковий клас, 0 – різні класи), d – відстань між векторними представленнями, m – відступ.

Порівняння функцій втрат на основі векторного представлення наведено у таблиці 2.4.

Таблиця 2.4 – Порівняння функцій втрат на основі векторного представлення.

Назва	**Стійкість (дисб.)**	**Диф-сть**	**Складність**	**Чутливість (вибірка)**	**Параметри**
Triplet Loss	Середня	Так	Висока	Дуже висока	Так (margin)
Center Loss	Висока	Так	Середня	Низька	Так (λ)
Contrastive Loss	Середня	Так	Середня	Висока	Так (margin)

На практиці часто використовуються комбіновані функції втрат, що поєднують переваги різних підходів та компенсують індивідуальні недоліки окремих функцій. Комбіновані функції зазвичай є зваженою сумою кількох функцій втрат:

$$L_{combined} = \sum_{i=1}^{K}\omega_i L_i,$$

де L_i – окремі функції втрат, ω_i – їх ваги, K – кількість компонентів.

Типовою комбінацією для семантичної сегментації є поєднання Dice Loss та Cross-Entropy:

$$L = \alpha L_{Dice} + (1-\alpha) L_{CE} .$$

Така комбінація використовує переваги обох функцій: Dice Loss забезпечує стійкість до дисбалансу класів та пряму оптимізацію метрики сегментації, тоді як Cross-Entropy забезпечує стабільні градієнти та швидшу збіжність на початкових етапах навчання.

Для задач, де важлива точність меж, застосовується комбінація функції на основі областей та функцій на основі меж:

$$L = \alpha L_{Dice} + \beta L_{Boundary} .$$

Типова стратегія полягає у динамічному налаштуванні ваг протягом навчання – на початку домінує функція на основі областей, для грубої локалізації, а потім поступово збільшується вага функції для уточнення меж.

Для сегментації екземплярів, як у розглянутих у першому розділі архітектурах Mask R-CNN та Panoptic-DeepLab, використовуються багатозадачні функції втрат, що поєднують втрати для різних компонентів:

$$L_{total} = L_{cls} + L_{box} + L_{mask} ,$$

де L_{cls} – функція втрат для класифікації, L_{box} – для регресії обмежувальної рамки, L_{mask} – для сегментаційної маски.

У моделі InstanceCLIPSeg, яка буде детально розглянута у наступних розділах, використовується комбінація втрат для модуля прогнозування центрів (centers head) та модуля прогнозування зсувів/відстаней (offset/distance head):

$$L_{total} = L_{centers} + L_{offsets} ,$$

де $L_{centers}$ – зважена MSE для прогнозування центрів об'єктів, $L_{offsets}$ – зважена L1 для прогнозування відстаней до меж.

2.2 Зважена метрика для аналізу розбиття результатів сегментації

Розглянуті функції втрат є ефективними для різних підходів до сегментації, однак залишається актуальною проблема подолання семантичної прірви (semantic gap) між низькорівневими ознаками та сприйняттям людини. Врахування просторового контексту зображення та сукупної структури сімейств областей є кроком уперед порівняно з аналізом окремих сегментів. Отже, обґрунтованою є необхідність вивчення методів порівняння розбиттів множин, оскільки вони слугують моделями довільної кластеризації.

У контексті роботи з множинними сегментаціями – зокрема при порівнянні алгоритмів, пошуку балансу між недостатньою та надмірною сегментацією, а також при навчанні моделей для семантичної, інстанс- (сегментації екземплярів) та паноптичної сегментації – виникає необхідність урахування метричних властивостей порівнянь фактор-множин. Завдяки консолідованій геометричній структурі вони залишаються стійкими до змінних умов зйомки, часткових перекриттів та кольорових трансформацій.

Перехід від мір подібності, що мають лише властивості рефлексивності та симетрії, до метрик, які додатково задовольняють нерівність трикутника, відкриває можливості для ефективної попередньої обробки даних. Це забезпечує значне підвищення обчислювальної ефективності. Зокрема, використання нерівності трикутника в алгоритмах пошуку дає змогу швидко відсіювати суттєво відмінні елементи, уникаючи повного перебору.

Нехай U – довільна скінченна множина з потужністю N. Чітка кластеризація генерує різні розбиття $X = [x]_1, [x]_2, \ldots, [x]_m$, де $[x]i \neq \varnothing$, $U = \bigcup i = 1^m [x]_i$, $\forall i \neq j \Rightarrow [x]_i \cap [x]_j = \varnothing$, та $Y = [y]_1, [y]_2, \ldots, [y]_n$.

Факторні множини X/P, X/Q фактично генеруються або різними алгоритмами, або одним і тим же алгоритмом з різними параметрами. Розглянемо відомі метрики з метою їх модифікації для використання в сегментації зображень.

Метрика van Dongen [95] належить до типу метрик, заснованих на пошуку перетинів класів еквівалентності з максимальними потужностями:

$$\rho(X,Y) = 2N - \sum_{i=1}^{m} \max_{j \in 1,2,...,n} |[x]i \cap [y]j| - \sum j = 1^{n} \max i \in 1,2,...,m |[x]_i \cap [y]_j| .$$

Слід підкреслити, що пошук екстремумів може значно підвищити вплив відповідних викидів або аномалій, тим самим спотворюючи результат порівняння розбиттів у цілому.

Міра подібності Larsen [96] та її трансформація у метрику:

$$\rho(X,Y) = \frac{d(X,Y) + d(Y,X)}{2},$$

де

$$d(X,Y) = \frac{1}{N} \sum_{i=1}^{m} \max_{j \in 1,2,...,n} \frac{2|[x]_i \cap [y]_j|}{|[x]_i| + |[y]_j|} - 1 .$$

Метрика Міркіна [97], отримана узагальненням метрики Геммінга, представляє безсумнівний інтерес:

$$\rho(X,Y) = \sum_{i=1}^{m} |[x]i|^2 + \sum j = 1^{n} |[y]j|^2 - 2\sum i = 1^{m} \sum_{j=1}^{n} |[x]_i \cap [y]_j|^2 .$$

Метрика Мейли [98], заснована на ентропії та відома як варіація інформації, знайшла широке застосування:

$$\rho(X,Y) = H(X) + H(Y) - 2I(X,Y),$$

де ентропія розбиття:

$$H(X) = -\sum_{i=1}^{m} p([x]_i) \log p([x]_i), \quad p([x]_i) = |[x]_i| / N,$$

та взаємна інформація:

$$I(X,Y) = \sum_{i=1}^{m}\sum_{j=1}^{n} p([x]_i,[y]_j) \log \frac{p([x]_i,[y]_j)}{p([x]_i)p([y]_j)}, \quad p([x]_i,[y]_j) = |[x]_i \cap [y]_j| / N.$$

Earth Mover's Distance (EMD) [99], спочатку визначена як вартість трансформації одного розподілу в інший, може використовуватись для порівняння розбиттів. Для наборів описів ознак $P = (p_1, w_{p_1}),\ldots,(p_m, w_{p_m})$ та $Q = (q_1, w_{q_1}),\ldots,(q_n, w_{q_n})$, сформованих на розбиттях, де w_{p_i} та w_{q_j} – ваги, обчислення EMD є розв'язанням задачі лінійного програмування:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} d(p_i,q_j)c(p_i,q_j) \to \min.$$

За обмежень:

$$c(p_i,q_j) \geq 0, \quad \sum_{j=1}^{n} c(p_i,q_j) \leq w_{p_i}, \quad \sum_{i=1}^{m} c(p_i,q_j) \leq w_{q_j},$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c(p_i,q_j) = \min\left\{\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\right\},$$

де $c(p_i,q_j)$ – кількість «продукту, що транспортується» з кластера p_i до кластера q_j, $d(p_i,q_j)$ – витрати на транспортування одиниці.

Слід зазначити, що якщо $\sum_{i=1}^{m} w_{p_i} = \sum_{j=1}^{n} w_{q_j} = 1$ і $d(p_i,q_j)$ є метрикою, то EMD також є метрикою.

MiCROM (Minimum-Cost Region Matching) [100] є розвитком EMD і моделює порівняння сегментованих зображень як задачу мінімального вартісного потоку в мережі. Перевагою цих метрик є використання вагових коефіцієнтів, що адаптує їх до сегментації зображень, тоді як недоліком є висока обчислювальна складність при великій кількості класів еквівалентності.

Аналіз відомих функцій втрат виявив, що широко використовувані функції, такі як Dice та IoU, не мають строгого математичного обґрунтування як власне метрики і не повністю враховують геометричну структуру розбиттів. Вони не задовольняють нерівності трикутника, що обмежує можливості їх використання для ієрархічного пошуку та ефективного порівняння сегментацій.

Нехай U – довільна вимірна множина з мірою $\mu(U)<\infty$, тобто для будь-якої $A\subseteq U$ існує деяке число $\mu(A)$, яке є мірою (довжина, площа, об’єм, розподіл маси, розподіл ймовірності, потужність тощо). Відома базова метрика для порівняння розбиттів:

$$\rho(X,Y)=\sum_{k=1}^{m}\sum_{l=1}^{n}\mu([x]_k \triangle [y]_l)\mu([x]_k \cap [y]_l), \qquad (2.1)$$

де $[x]_k \triangle [y]_l = ([x]_k \setminus [y]_l)\cup([y]_l \setminus [x]_k)$ – симетрична різниця.

Ця метрика одночасно містить міри подібності та відмінності класів еквівалентності з простотою обчислення. Масштабована формалізована процедура обчислення функціоналу така. Нехай $S_{\triangle}(X,Y)=(S_k^{kl})$, $S_{\cap}(X,Y)=(S_{kl}^{k})$, де $k=\overline{1,m}$, $l=\overline{1,n}$, $S_{\triangle}^{kl}=\mu([x]_k \triangle [y]_l)$, $S_{\cap}^{kl}=\mu([x]_k \cap [y]_l)$. Тоді поелементне матричне множення $Q_{\triangle}^{kl}=S_{\triangle}^{kl}S_{\cap}^{kl}$ дає $(m\times n)$ матрицю $Q_{\triangle\cap}$, сума елементів якої є $\rho(X,Y)$.

На основі проведеного аналізу базова метрика видається найбільш прийнятною для порівняння сегментацій зображень, але існує потреба

розширити розуміння зв'язку просторового вмісту із зображенням. Припустимо, що U є скінченним полем зору, тоді пропонується зважена метрика:

$$\rho(X,Y)=\sum_{k=1}^{m}\sum_{l=1}^{n}\varphi([x]_k,[y]_l)\,|[x]_k \triangle [y]_l|\,|[x]_k \cap [y]_l|, \qquad (2.2)$$

де $\varphi([x]_k,[y]_l)$ – функція, що характеризує яскравість, колірні або текстурні властивості класів еквівалентності.

Запропонований функціонал враховує не лише форму регіонів (окремих елементів розбиття) та їх взаємне просторове розташування, але й різні візуальні властивості у стислій формі, тобто:

$$X=(\alpha_1,[x]_1),(\alpha_2,[x]_2),\ldots,(\alpha_m,[x]_m),$$

де α_i – характеристика (вага) i-го елемента розбиття.

2.3 Доведення метричних властивостей зваженої метрики

Для доведення того, що запропонований функціонал є метрикою, достатньо показати, що $\rho(X,Y)$ є невід'ємним і задовольняє аксіомам тотожності (рефлексивності), симетрії та нерівності трикутника.

Позначимо $\alpha_{ij}=\varphi([x]_i,[y]_j)$. Нехай $\alpha_{ij}=\alpha_{ji}>0$, цю умову легко виконати на етапі вибору яскравісних, колірних або текстурних характеристик. Якщо будь-яке $\alpha_{ij}=0$, то X, Y перестають бути розбиттями, оскільки не покривають U. Таким чином, функціонал очевидно невід'ємний і задовольняє аксіомі симетрії завдяки комутативності симетричної різниці та перетину.

Доведення рефлексивності. Необхідно показати, що $\rho(X,Y)=0\Leftrightarrow X=Y$ Спочатку доведемо достатність. Відповідно до властивості симетрії:

$$\rho(X,X)=\sum_{k=1}^{m}\sum_{l=1}^{m}\alpha_{kl}\,|[x]_k\triangle[x]_l\,||[x]_k\cap[x]_l\,|=$$
$$=\sum_{k=1}^{m}\alpha_{kk}\,|[x]_k\triangle[x]_k\,||[x]_k\cap[x]_k\,|+2\sum_{\substack{k,l=1\\k>l}}^{m}\alpha_{kl}\,|[x]_k\triangle[x]_l\,||[x]_k\cap[x]_l\,|. \qquad (2.3)$$

Вираз складається з m членів з однаковими індексами та (m^2-m) членів з різними елементами розбиття. Перша група членів очевидно складається з нулів, оскільки $|[x]_k\triangle[x]_k|=0$ для всіх $k=\overline{1,m}$, а друга група також дає нульові члени, оскільки $|[x]_k\cap[x]_l|=0$ для всіх $k,l=\overline{1,m}$ за умови $k\neq l$.

Тепер доведемо необхідність. Припустимо, що для двох розбиттів $X\neq Y$ виконується рівність функціоналу нулю. Оскільки всі члени функціоналу невід'ємні, він дорівнює нулю лише якщо кожен з них дорівнює нулю.

Виберемо елемент $[x]^*\in X$, він входить до множини «нульових» членів $\alpha_l\cdot|[x]^*\triangle[y]_l|\cdot|[x]^*\cap[y]_l|$, де $l=\overline{1,n}$.

Припустимо, що $[x]^*$ не належить до Y, тоді для всіх $[y]_l\in Y$ виконується нерівність $|[x]^*\triangle[y]_l|\neq 0$. Тоді для всіх індексів $l=\overline{1,n}$ має бути істинною рівність $|[x]^*\cap[y]_l|=0$. Проте це можливо лише якщо $[x]^*=\varnothing$, оскільки $[x]^*\in U$ і сімейство $Y=[y]_1,[y]_2,\ldots,[y]_n$ покриває множину U. Але $[x]^*$ належить до розбиття X тієї ж множини U, і $[x]^*\neq\varnothing$.

Тоді існують елементи $[y]_1^*[y]_2^*\ldots,[y]_p^*\in Y$, які покривають підмножину $[x]^*\subset U$ і мають непорожній перетин з нею, тобто $|[x]^*\cap[y]_r^*|\neq 0$, $r=\overline{1,p}$. Отримуємо суперечність – будь-який елемент $[x]^*\in X$ є елементом з Y, тобто $X\subset Y$. В силу симетрії $Y\subset X$ і остаточно $X=Y$, що доводить рефлексивність.

Доведення нерівності трикутника. Для доведення розглянемо декілька допоміжних фактів. Для будь-якого зображення U на булеані πU виділимо Π_U – множину скінченних розбиттів. Кожен клас еквівалентності $[x]_k\in X\in\Pi_U$ характеризується деяким числом:

$$X = (\alpha_1,[x]_1),(\alpha_2,[x]_2),\ldots,(\alpha_m,[x]_m).$$

Для довільної області $D \subset U$ та будь-якого розбиття $X = [x]_1,[x]_2,\ldots,[x]_m$, оскільки розбиття X ділить множину D на неперетинні підмножини, адитивність потужності тягне за собою:

$$|D| = \sum_{i=1}^{m} |D \cap [x]_i|.$$

З урахуванням ваг:

$$|D| = \sum_{i=1}^{m} \alpha_i |D \cap [x]_i|. \qquad (2.4)$$

Існує можливість переписати функціонал в еквівалентній формі. Для будь-яких множин $A, B \in U$:

$$|A \triangle B| = |A| + |B| - 2|A \cap B|.$$

Справді, з рівностей $A = (A \setminus B) \cup (A \cap B)$ та $B = (B \setminus A) \cup (A \cap B)$, де множини $A \setminus B$ та $A \cap B$ не перетинаються, з адитивності потужності: $|A| = |A \setminus B| + |A \cap B|$ та $|B| = |B \setminus A| + |A \cap B|$. Підсумовуючи ці рівності та враховуючи, що $A \triangle B = (A \setminus B) \cup (B \setminus A)$ маємо $|A| + |B| = |A \setminus B| + |B \setminus A| + 2|A \cap B| = |A \triangle B|$.

Тоді функціонал набуває вигляду:

$$\rho(X,Y) = \sum_{k=1}^{m} |[x]_k|^2 + \sum_{l=1}^{n} |[y]l|^2 - 2\sum_{k=1}^{m}\sum_{l=1}^{n} \alpha_{kl} |[x]_k \cap [y]_l|^2. \qquad (2.5)$$

Розглянемо перетини трьох довільних розбиттів $X, Y, Z \in P_U$ з відповідними вагами. Позначимо $d_{kli} = |[x]_k \cap [y]_l \cap [z]_i|$, $k = \overline{1,m}$, $l = \overline{1,n}$, $i = \overline{1,p}$.

Нерівність трикутника виконується тоді і тільки тоді, коли:

$$\sum_{i=1}^{p}\left(\sum_{k=1}^{m}\sum_{l=1}^{n} d_{kli}\right)^2 + \sum_{k=1}^{m}\sum_{l=1}^{n}\lambda_{kl}\left(\sum_{i=1}^{p} d_{kli}\right)^2 \geq \sum_{k=1}^{m}\sum_{i=1}^{p}\mu_{ki}\left(\sum_{l=1}^{n} d_{kli}\right)^2 + \sum_{l=1}^{n}\sum_{i=1}^{p}\theta_{li}\left(\sum_{k=1}^{m} d_{kli}\right)^2,$$

де λ_{kl}, μ_{ki}, θ_{li} – відповідні вагові коефіцієнти для пар розбиттів.

При фіксації $p = 1$ і позначенні $g_{kl} = d_{k|l}$ ($g_{kl} \geq 0$), структура лівої частини нерівності:

$$2\sum_{k=1}^{m}\sum_{l=1}^{n}(1+\lambda_{kl})(g_{kl})^2 + \sum_{(k,l)\neq(k',l')} g_{kl}g_{k'l'},$$

і правої частини:

$$2\sum_{k=1}^{m}\sum_{l=1}^{n}(\mu_{k1}+\theta_{l1})(g_{kl})^2 + \sum_{k=1}^{m}\mu_{k1}\sum_{l\neq l'} g_{kl}g_{kl'} + \sum_{l=1}^{n}\theta_{l1}\sum_{k\neq k'} g_{kl}g_{k'l}.$$

Порівнюючи ці вирази, ліва частина містить усі попарні добутки елементів матриці G з нерівними індексами, а права частина – лише попарні добутки з одного рядка або стовпця. Кількість членів у правій частині менша, і всі вони містяться в лівій частині, тобто нерівність виконується для одиничних ваг.

Загалом нерівність виконується при умовах:

$$\lambda_{kl} + 1 \geq \mu_{k1} + \theta_{l1}.$$

Таким чином, метричний простір формується на певному обмеженні $P'_U \subset P_U$, де нерівність трикутника виконується при визначених умовах на

вагові коефіцієнти, і зважений функціонал є метрикою для зіставлення сегментацій зображень.

2.4 Експериментальне дослідження запропонованої метрики

Для перевірки практичної застосовності запропонованої метрики було проведено експериментальне дослідження. Використано архітектуру SegNet [46], яка розв'язує задачу попіксельної регресії. Як набір даних створено синтетичний набір з п'яти фігур: коло, прямокутник, трикутник, п'ятикутник, зірка, що не перекриваються на зображенні 100×100 пікселів. Кількість фігур розподілена нормально, від двох до п'яти на зображення. Модель видає одноканальне зображення, і метою є розбиття зображення на класи, включаючи фон.

Першою проблемою є виділення елементів для розбиття з урахуванням зв'язності та недискретності виходу. Критично важливо, щоб операція виділення була диференційовною – інакше градієнти будуть втрачені і навчання мережі стане неможливим.

Для виділення використано KNN з перевіркою зв'язності елементів, оскільки кількість елементів не є фіксованою. На початку навчання кількість кластерів може бути великою, що вимагає значних обчислювальних ресурсів (рис. 2.1). Оскільки алгоритм KNN не є диференційовним, застосовано прийом STE [23], який дозволяє передавати градієнти назад [101].

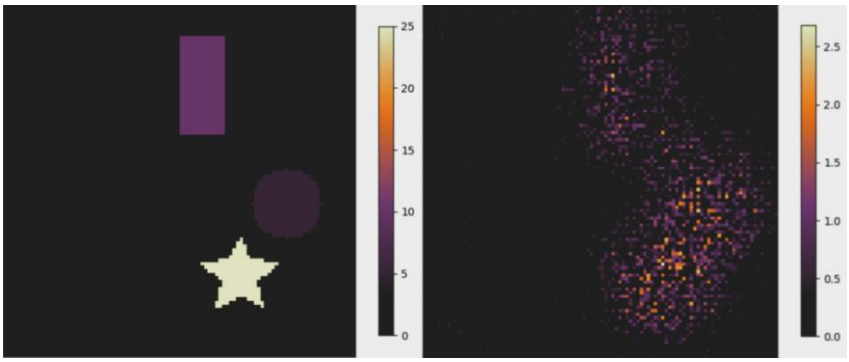

Рисунок 2.1 – Вхідні та вихідні дані на перших кроках

Як експеримент досліджено різні варіанти апроксимації: згортку 1×1, Gumbel-Softmax [102] та soft masks (рис. 2.2). Soft masks описують ймовірнісне значення приналежності пікселя до жорстких центроїдів. Результати показують, що функція не навчає мережу сегментувати об'єкти за класами, але дозволяє знаходити об'єкти та підкреслювати розділення областей з урахуванням зв'язності (рис. 2.3). Це робить метрику перспективним кандидатом для комбінованих функцій втрат.

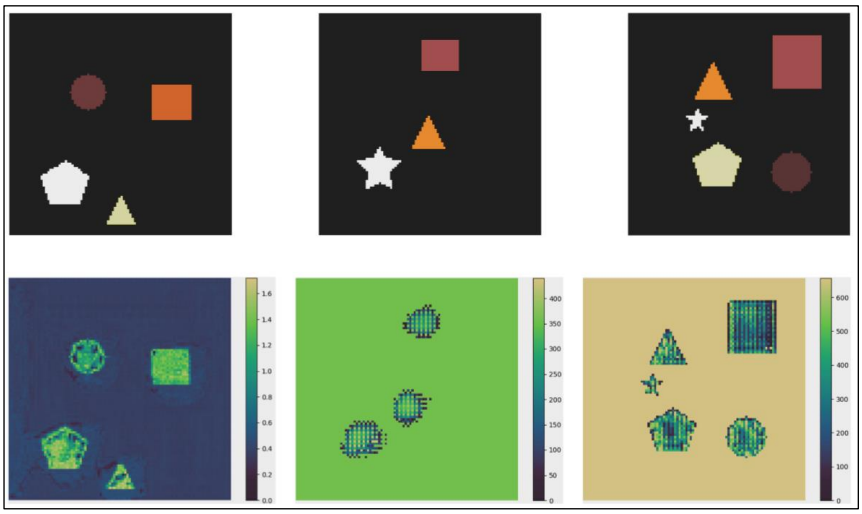

Рисунок 2.2 – Порівняння різних методів апроксимації: conv 1×1, Gumbel-Softmax та soft masks

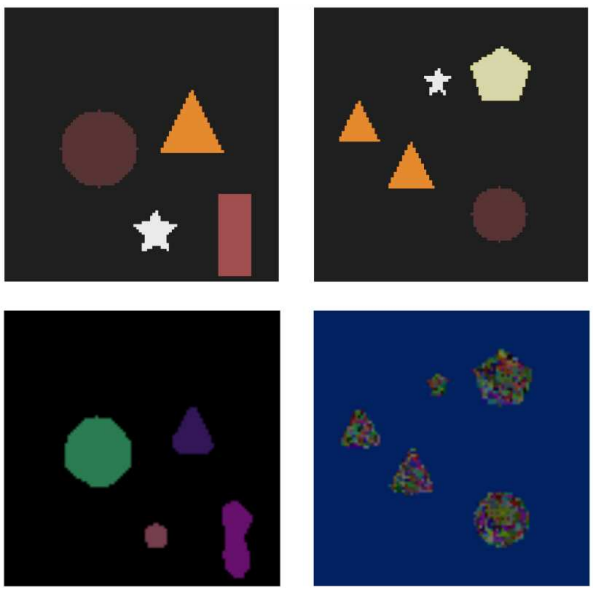

Рисунок 2.3 – Приклади різних виходів KNN з різними STE

Додатково перевірені градієнти перед початком навчання (рис. 2.4). Вхідне зображення передавалось через модуль KNN, і вихід направлявся до функції втрат для зворотного проходу. У процесі обчислення перетину між фактичним результатом та еталонною розміткою, градієнти примусово обнулялися, коли метрика дорівнювала нулю. Це зумовлено тим, що самоперетин призводив би до появи ненульових градієнтів, що негативно вплинуло б на процес навчання.

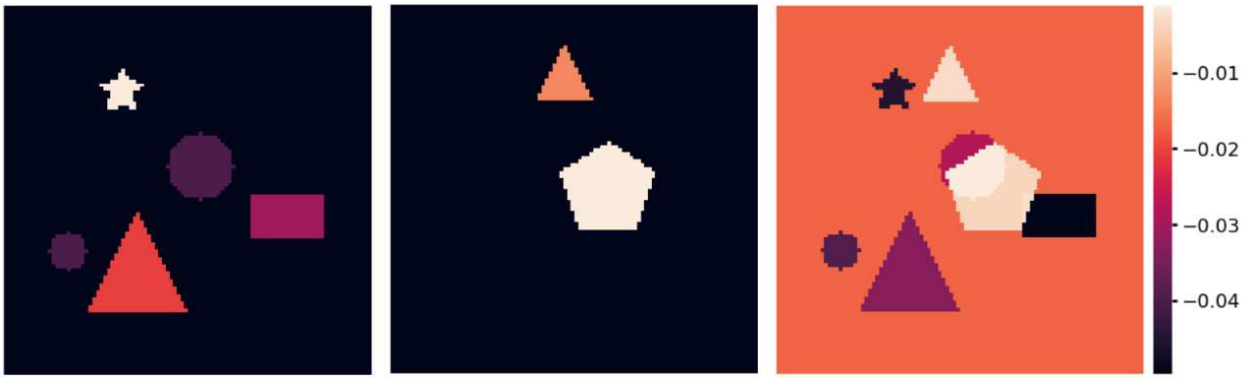

Рисунок 2.4 – Візуалізація обчислених градієнтів під час зворотного проходу

Запропонована зважена метрика для порівняння розбиттів зображень враховує як геометрію областей, так і їхні семантичні властивості за допомогою вагової функції φ. На відміну від поширених (Dice/Tversky, IoU, Lovász-Softmax) та орієнтованих на границі (Boundary/Hausdorff) функцій втрат, запропонований підхід принципово вирізняється тим, що визначає метрику в математичному сенсі на просторі розбиттів. Водночас згадані аналоги або не є метриками, або потребують спеціального згладжування та не задовольняють нерівність трикутника.

Властивість нерівності трикутника уможливлює ієрархічний пошук по простору розбиттів та більш обґрунтоване налаштування гіперпараметрів. На відміну від EMD та MiCROM, де точність досягається через розв’язання задач лінійного програмування, запропонована метрика зводиться до матричних

обчислень і є потенційно простішою з обчислювальної точки зору при помірній кількості областей.

Проте на початкових етапах навчання кількість областей зазвичай велика, що збільшує вимоги до пам’яті. Критичним обмеженням для використання метрики, як наскрізної (end-to-end) функції втрат, є необхідність диференційовного виділення сегментів, що не перетинаються. Застосування алгоритму KNN із перевіркою зв’язності та апроксимації STE дозволяє розв’язати цю проблему.

На синтетичному наборі даних модель, оптимізована за допомогою запропонованої метрики, змогла розділяти об’єкти з урахуванням зв’язності. Це робить метрику кандидатом для комбінованих функцій втрат, де вона підкреслюватиме правильну топологію розбиття, доповнюючи функції, що фокусуються на семантиці або належності до класу.

Вибір та нормалізація φ безпосередньо впливають на чутливість до розмірів об’єктів та контрасту. Порівняння ключових характеристик наведено у таблиці 2.5.

Таблиця 2.5 – Порівняння запропонованої метрики з існуючими підходами

Характеристика	**Dice/Tversky**	**IoU**	**EMD/MICROM**	**Запропонована**
Є метрикою?	Ні	Ні	Так (умовно)	Так
Нер. трикутника	Ні	Ні	Так	Так
Складність	Низька	Низька	Висока	Середня
Семантика	Ні	Ні	Так (ваги)	Так (ϕ)
Топологія	Ні	Ні	Частково	Так
Диф-сть	Так	Часткова	Ні	Так (STE)

2.5 Висновки за розділом

1) Проведено комплексний аналіз функцій втрат для задач сегментації зображень. Функції втрат систематизовано за категоріями: distribution-based (Cross-Entropy, Balanced CE, Weighted CE, Focal Loss), region-based (Dice, Tversky, IoU, Lovász-Softmax), boundary-based (Boundary Loss, Hausdorff Loss, Active Contour) та embedding-based (Triplet Loss, Center Loss, Contrastive Loss). Для кожної категорії визначено переваги, недоліки та області застосування. Аналіз виявив, що жодна з широко використовуваних функцій втрат не є повноцінною метрикою на просторі розбиттів та не задовольняє нерівності трикутника, що обмежує можливості ієрархічного пошуку та ефективного порівняння сегментацій.

2) Запропоновано зважену метрику для порівняння розбиттів зображень, яка, на відміну від наявних підходів, враховує не лише форму регіонів та їх взаємне просторове розташування, але й семантичні властивості (яскравість, колір, текстуру) через вагову функцію $\varphi([x]k,[y]l)$. Метрика визначається функціоналом $\rho(X,Y)=\sum_{k=1}^{m}\sum_{l=1}^{n}\varphi([x]_k,[y]_l)\,|[x]_k \triangle [y]_l|\,|[x]_k \cap [y]_l|$, що одночасно охоплює міри подібності та відмінності класів еквівалентності з простою обчислюваністю через матричні операції.

3) Математично доведено, що запропонований функціонал задовольняє всі аксіоми метрики: невід'ємність, рефлексивність, симетрію та нерівність трикутника. Особливу увагу приділено доведенню нерівності трикутника, що є ключовою властивістю для використання метрики в алгоритмах пошуку та порівняння. Встановлено, що метричний простір формується на певному обмеженні $P'_U \subset P_U$, де нерівність трикутника виконується при визначених умовах на вагові коефіцієнти.

4) Розроблено підхід до застосування запропонованої метрики як функції втрат для навчання нейронних мереж. Для подолання проблеми недиференційовності операції виділення сегментів запропоновано

використання кластеризації KNN з перевіркою зв’язності елементів та апроксимації Straight-Through Estimator (STE). Експериментально досліджено різні варіанти STE: згортка 1×1, Gumbel-Softmax та soft masks.

5) Проведено експериментальне дослідження на синтетичному наборі даних, що підтвердило працездатність запропонованого підходу. Результати показали, що метрика дозволяє знаходити об’єкти та підкреслювати розділення областей з урахуванням їх зв’язності, що робить її перспективним кандидатом для комбінованих функцій втрат, де вона доповнюватиме функції, що фокусуються на семантиці або приналежності до класу.

3 АРХІТЕКТУРА МУЛЬТИМОДАЛЬНОЇ МОДЕЛІ СЕГМЕНТАЦІЇ ЕКЗЕМПЛЯРІВ ЗА ТЕКСТОВИМ ЗАПИТОМ

3.1 Теоретичні основи та базові компоненти архітектури

Задача сегментації екземплярів за текстовим запитом вимагає поєднання двох фундаментальних завдань: розуміння природної мови для інтерпретації запиту користувача та візуального аналізу для локалізації та виділення відповідних об’єктів на зображенні. Традиційні методи сегментації екземплярів працюють з фіксованим набором класів, визначеним на етапі навчання, що обмежує їх практичне застосування. Для подолання цього обмеження необхідні мультимодальні архітектури, здатні оперувати у спільному семантичному просторі візуальних та текстових представлень.

Фундаментом для побудови таких архітектур стала модель CLIP [11], розроблена дослідниками з OpenAI. Вона була навчена на масштабному наборі з чотириста мільйонів пар зображення-текст. Процес навчання базується на контрастивній функції втрат, що максимізує подібність між векторним представленням відповідних пар зображення-текст та мінімізує подібність між невідповідними парами. В результаті модель формує спільний семантичний простір, де візуальні та текстові ознаки розташовуються поблизу, якщо вони семантично пов’язані.

Архітектура CLIP складається з двох окремих енкодерів: візуального енкодера для обробки зображень та текстового енкодера для обробки природної мови. Візуальний енкодер може бути реалізований на основі згорткової нейронної мережі (ResNet) [103] або трансформера (ViT) [104]. Обидва енкодери перетворюють свої входи у спільний простір векторних ознак фіксованої розмірності, де семантична близькість вимірюється за допомогою косинусної відстані.

Ключовою перевагою CLIP є здатність до zero-shot класифікації. Замість навчання на основі конкретних класів із розміченими прикладами, модель

може класифікувати зображення за довільними текстовими описами, перетворюючи назви класів на текстові запити. Ця властивість відкриває можливості для створення систем комп'ютерного зору з відкритим словником (open-vocabulary), де набір можливих категорій не обмежений заздалегідь.

На основі CLIP було розроблено модель CLIPSeg [12], призначену для семантичної сегментації за текстовим або візуальним запитом. CLIPSeg розширює можливості CLIP від класифікації цілого зображення до попіксельної класифікації, зберігаючи при цьому здатність працювати з довільними текстовими описами. Архітектура CLIPSeg додає до замороженого CLIP енкодера легкий декодер на основі трансформера, який генерує маску сегментації відповідно до запиту.

Декодер CLIPSeg використовує механізм FiLM [105] для умовної модуляції візуальних ознак. Візуальні ознаки з різних шарів CLIP енкодера комбінуються та поступово збільшуються у роздільній здатності за допомогою транспонованих згорток. Текстовий векторний опис запиту використовується для модуляції цих ознак через афінні перетворення, що дозволяє моделі фокусуватися на регіонах, семантично пов'язаних із запитом.

Однак CLIPSeg вирішує лише задачу семантичної сегментації, де результатом є бінарна маска всіх пікселів, що відповідають запиту, без розділення на окремі екземпляри. Для багатьох практичних застосувань, таких як підрахунок об'єктів, відстеження руху або взаємодія з конкретними екземплярами, необхідна саме сегментація екземплярів, де кожен окремий об'єкт отримує унікальний ідентифікатор.

Для вирішення задачі сегментації екземплярів у запропонованій архітектурі InstanceCLIPSeg використовуються концепції з моделі Panoptic-DeepLab [106]. Ця модель демонструє ефективність bottom-up підходу для паноптичної сегментації, де об'єднуються переваги семантичної сегментації та сегментації екземплярів. Ключова ідея Panoptic-DeepLab полягає у передбаченні двох типів інформації для кожного пікселя: належності до семантичного класу та характеристик, що дозволяють згрупувати пікселі в

окремі екземпляри. Модель використовує три модулі на виході: семантичний для класифікації пікселів, модуль центрів для прогнозування теплової карти центрів екземплярів та модуль зміщень для прогнозування вектора від кожного пікселя до центру його екземпляра. Під час роботи моделі центри екземплярів виявляються як локальні максимуми на тепловій карті, а пікселі групуються навколо найближчого центру на основі передбачених зміщень.

Такий підхід має суттєву перевагу: він не потребує попереднього виявлення регіонів інтересу (Region Proposals), як у двоетапних методах типу Mask R-CNN. Це забезпечує кращу масштабованість та здатність працювати з довільною кількістю екземплярів без обмежень, накладених механізмом Non-Maximum Suppression для обмежувальних рамок.

Проте оригінальний підхід Panoptic-DeepLab з прогнозуванням зміщень до центру виявився непридатним для використання в InstanceCLIPSeg. Причина полягає у характеристиках функції активації. Зміщення до центру може мати як додатні, так і від'ємні значення залежно від положення пікселя відносно центру об'єкта. При використанні функції активації ReLU, яка обнуляє всі від'ємні значення, мережа не здатна коректно передбачати від'ємні зміщення і буде більш стабільною.

Для вирішення цієї проблеми було застосовано альтернативний підхід з моделі PRN (Panoptic Refinement Network) [107]. Замість прогнозування зміщень до центру, PRN пропонує передбачати відстані від кожного пікселя до чотирьох границь обмежувальної рамки об'єкта: верхньої, нижньої, лівої та правої. Усі ці відстані за визначенням є невід'ємними для пікселів всередині об'єкта, що узгоджується з властивостями функції активації ReLU.

Такий підхід також має додаткову перевагу – він безпосередньо надає інформацію про обмежувальну рамку об'єкта, яка може бути корисною для подальшої обробки або візуалізації результатів. Знаючи відстані до всіх чотирьох границь, можна безпосередньо обчислити координати обмежувальної рамки без додаткових перетворень.

3.2 Загальна структура моделі InstanceCLIPSeg

Архітектура InstanceCLIPSeg побудована на основі CLIPSeg з суттєвими модифікаціями для підтримки сегментації екземплярів. Модель складається з чотирьох основних компонентів: енкодера зображення, енкодера запиту, декодера центрів та декодера відстаней.

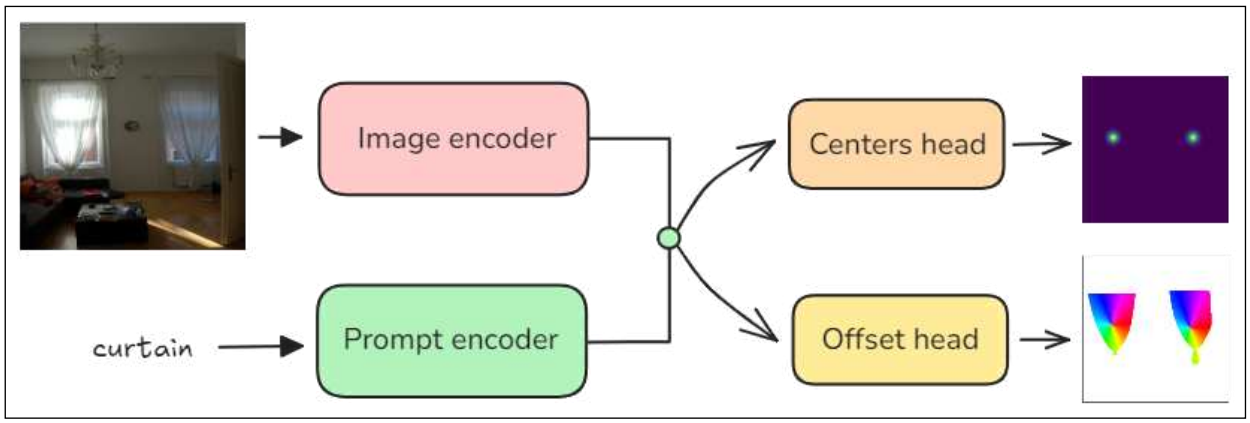

Рисунок 3.1 – Архітектура InstanceCLIPSeg

Енкодер зображення (image encoder) реалізований на основі Vision Transformer (ViT-B/16), адаптованого з оригінальної моделі CLIP. Vision Transformer розбиває вхідне зображення на непересічні фрагменти розміром 16×16 пікселів та обробляє їх як послідовність токенів за допомогою стандартної архітектури трансформера. Кожен фрагмент лінійно проектується у простір ембедингів, до якого додаються позиційні ембединги для збереження просторової інформації. Спеціальний токен [CLS] додається на початок послідовності для агрегації глобальної інформації про зображення.

Оригінальна модель CLIP використовує вхідну роздільну здатність 224×224 пікселів, що відповідає послідовності з 196 фрагментів (14×14). Для покращення здатності моделі працювати з об’єктами меншого розміру та підвищення деталізації сегментації, в InstanceCLIPSeg вхідну роздільну здатність збільшено до 352×352 пікселів, як і в CLIPSeg. Це відповідає

послідовності з 484 фрагментів (22×22) та забезпечує вдвічі вищу щільність просторового представлення.

Збільшення роздільної здатності потребує адаптації позиційних ембедингів, оскільки оригінальні ембединги навчені для сітки 14×14. Для цього застосовується білінійна інтерполяція позиційних ембедингів з розміру 14×14 до 22×22. Така інтерполяція зберігає відносні просторові відношення між позиціями та дозволяє використовувати знання, набуті моделлю під час попереднього навчання.

Енкодер зображення залишається нетренованим (frozen) під час навчання InstanceCLIPSeg. Це рішення зумовлене кількома факторами. По-перше, таке рішення зберігає візуальні представлення, набуті CLIP під час навчання на масштабному наборі даних. По-друге, це суттєво зменшує кількість параметрів, що навчаються, та обчислювальні витрати на навчання. По-третє, це запобігає катастрофічному забуванню (catastrophic forgetting), коли донавчання на специфічному наборі даних може погіршити загальні можливості моделі.

Енкодер запиту (prompt encoder) отримує на вхід текстовий опис об’єктів та перетворює його у векторне представлення. Текстовий енкодер CLIP базується на архітектурі трансформера з 12 шарами, прихованою розмірністю 512 та 8 модулями уваги. Вхідний текст токенізується за допомогою Byte Pair Encoding (BPE) з словником розміром 49152 токени.

Текстовий запит має описувати об’єкти, що належать до однієї семантичної категорії. Обмеження полягає у тому, що запит не повинен містити суперечливих ознак. Наприклад, запит «червоний та синій автомобіль» є коректним, оскільки описує автомобілі з різними атрибутами однієї категорії. Проте, запит «автомобіль та пішохід» потребує розділення на два окремі запити, оскільки описує об’єкти різних категорій.

Для формування текстового запиту використовується шаблон «a photo of a {object}», де {object} замінюється на опис шуканого об’єкта. Такий шаблон відповідає формату, використаному під час навчання CLIP, та забезпечує

кращу відповідність між текстовими та візуальними векторними представленнями. Дослідження показують, що правильний вибір шаблону запиту може суттєво впливати на якість zero-shot класифікації.

Вихід текстового енкодера – це вектор фіксованої розмірності 512, що представляє семантичний зміст запиту у спільному з візуальними векторними представленнями у просторі. Цей вектор використовується для умовної модуляції візуальних ознак у декодерах, спрямовуючи їх увагу на регіони, семантично відповідні запиту.

Декодер центрів (centers head) є першим з двох спеціалізованих декодерів InstanceCLIPSeg. Його задача – прогнозувати теплову карту ймовірних центрів мас об'єктів, що відповідають текстовому запиту. Кожен піксель вихідної карти представляє ймовірність того, що він є центром певного об'єкта шуканої категорії.

Центри об'єктів під час навчання кодуються за допомогою двовимірного розподілу Гауса. Для кожного об'єкта визначається його центр мас на основі маски сегментації, а потім навколо цього центру формується гаусівський пік з певним стандартним відхиленням. Функція кодування центру об'єкта з координатами (x_c, y_c) має вигляд:

$$G(x, y) = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2} \right),$$

де σ – стандартне відхилення, що визначає ширину піку.

Такий підхід, на відміну від бінарних міток, створює плавний градієнт ймовірності, що полегшує навчання мережі. Модель отримує позитивний зворотний зв'язок не лише за точне влучання в центр, а й за прогнозування у його околі.

Декодер відстаней (offset head) є другим спеціалізованим декодером. Його задача – прогнозувати для кожного пікселя, що належить об'єкту, відстані до чотирьох границь обмежувальної рамки цього об'єкта. Вихід

декодера – це чотириканальна карта, де кожен канал відповідає одній границі: верхній, нижній, лівій та правій.

Для пікселя з координатами (x, y), що належить об'єкту з обмежувальною рамкою $[x_{min}, y_{min}, x_{max}, y_{max}]$, цільові значення чотирьох каналів обчислюються як:

$$d_{top}(x,y) = y - y_{min} \quad d_{bottom}(x,y) = y_{max} - y \quad d_{left}(x,y) = x - x_{min} \quad d_{right}(x,y) = x_{max} - x .$$

Усі ці значення є невід'ємними для пікселів всередині обмежувальної рамки, що узгоджується з використанням функції активації ReLU. Для пікселів фону (що не належать жодному об'єкту) цільові значення встановлюються рівними нулю.

Важливою особливістю архітектури є відмова від додаткового прогнозування фону, як окремого класу. У традиційних підходах до сегментації, фон часто моделюється як окрема категорія, що потребує окремого модуля або додаткового каналу виходу. Замість виділення окремого каналу для класу фону, модель формує його автоматично шляхом інверсії сукупної маски знайдених об'єктів. Це рішення оптимізує структуру декодерів та зменшує кількість параметрів, що навчаються, зберігаючи при цьому повноту сцени.

Таке рішення обґрунтоване специфікою задачі. При пошуку об'єктів за текстовим запитом користувача цікавлять саме знайдені об'єкти, а не фон. Якщо необхідна явна сегментація фону, вона може бути легко отримана шляхом інвертування результуючої маски.

3.3 Архітектура декодерів та механізми відновлення роздільної здатності

Критичним аспектом архітектури сегментації є процес відновлення просторової роздільної здатності ознак після їх стиснення у енкодері. Vision Transformer зменшує просторову роздільну здатність вхідного зображення у 16 разів (розмір фрагмента), тому для отримання попіксельних передбачень необхідно відновити оригінальну роздільну здатність.

В оригінальній моделі CLIPSeg для збільшення розмірності використовується послідовність шарів TransposedConvolution (транспонована згортка, також відома як деконволюція). Транспонована згортка є оберненою операцією до звичайної згортки з кроком – замість зменшення роздільної здатності вона її збільшує шляхом вставки нулів між елементами та застосування згортки.

Проте цей підхід має відомий недолік – створення характерних артефактів типу «шахової дошки» (checkerboard artifacts) [108]. Ці артефакти виникають через нерівномірне перекриття при застосуванні транспонованої згортки. Коли розмір ядра не ділиться націло на крок, різні позиції вихідного зображення отримують внесок від різної кількості елементів ядра.

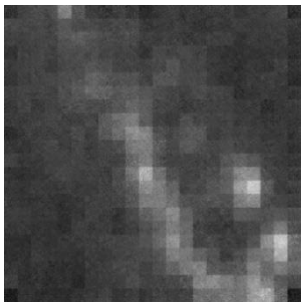

Рисунок 3.2 – Артефакти після TransposedConvolution

Математично, якщо використовується транспонована згортка з ядром розміру k та кроком s, то для вихідної позиції i кількість елементів ядра, що роблять внесок, визначається як кількість цілих j, для яких виконується

$0 \le i - j \cdot s < k$. Коли k не ділиться на s, ця кількість варіюється залежно від позиції i, створюючи періодичні коливання інтенсивності.

Для задачі семантичної сегментації, де важлива лише бінарна належність пікселя до певного класу, такі артефакти можуть бути частково компенсовані пороговою фільтрацією. Однак для задач InstanceCLIPSeg, де необхідні точні числові значення (центри об'єктів та відстані до границь), артефакти суттєво погіршують якість передбачень. Узгоджені, плавні значення у межах кожного об'єкта є критичними для коректної роботи алгоритму постобробки. Під час постобробки, коли пікселі групуються на основі схожості передбачених відстаней, ці коливання можуть призводити до неправильного розділення об'єктів або об'єднання різних об'єктів.

Для пом'якшення проблеми артефактів застосовується спеціальна ініціалізація ваг транспонованої згортки – білінійна ініціалізація. Ваги ініціалізуються таким чином, щоб операція на початку навчання відповідала білінійній інтерполяції. Білінійний фільтр обчислюється за формулою:

$$w[i, j] = \left(1 - \frac{|i - c|}{f}\right) \cdot \left(1 - \frac{|j - c|}{f}\right),$$

де $f = [k / 2]$ – коефіцієнт масштабування, $c = (2f - 1 - f \bmod 2) / (2f)$ – центр фільтра. Обчислений фільтр копіюється у всі зрізи тензора ваг, забезпечуючи плавний початок навчання.

Для декодера центрів (centers head) було розроблено архітектуру з чотирьох послідовних блоків, що відновлюють просторову роздільну здатність від 22×22 (вихід ViT для роздільної здатності 352×352) до 352×352 пікселів. Проміжні роздільні здатності становлять 44×44, 88×88 та 176×176 пікселів, що відповідає послідовному подвоєнню на кожному етапі.

Кожен блок декодера центрів складається з двох функціонально різних підблоків: підблоку збільшення розмірності та підблоку уточнення.

Підблок збільшення розмірності реалізує власне збільшення просторової роздільної здатності та містить:

1) транспоновану згортку (transposed convolution) з розміром ядра 4×4, кроком 2 та відступом (padding) 1, що забезпечує подвоєння роздільної здатності;

2) пакетну нормалізацію (batch normalization) для стабілізації розподілу активацій;

3) функцію активації ReLU для введення нелінійності.

Вибір параметрів транспонованої згортки (kernel_size=4, stride=2, padding=1) забезпечує подвоєння роздільної здатності за формулою:

$$H_{out} = (H_{in} - 1) \times stride - 2 \times padding + kernel\_size = (H_{in} - 1) \times 2 - 2 + 4 = 2 \times H_{in}.$$

Підблок уточнення призначений для згладжування артефактів, що виникають після транспонованої згортки, та містить:

1) звичайну згортку (convolution) з розміром ядра 3×3, кроком 1 та відступом 1, що зберігає роздільну здатність;

2) пакетну нормалізацію (batch normalization);

3) функцію активації ReLU.

Комбінація транспонованої згортки з наступною звичайною згорткою є ефективним способом зменшення артефактів. Звичайна згортка діє як локальний фільтр, що усереднює та вирівнює значення сусідніх пікселів. Це особливо важливо для теплової карти центрів, де необхідні плавні гаусівські піки без високочастотних коливань.

Загальна формула для обчислення кількості параметрів одного блоку декодера центрів:

1) TransposedConv: $C_{in} \times C_{out} \times 4 \times 4 + C_{out}$ (bias);

2) BatchNorm (1): $2 \times C_{out}$ (gamma та beta);

3) Conv: $C_{out} \times C_{out} \times 3 \times 3 + C_{out}$ (bias);

4) BatchNorm (2): $2 \times C_{out}$.

Для декодера відстаней вимоги до якості результату ще вищі, оскільки він потребує більшого просторового контексту та плавніший розподіл значень. Карти відстаней повинні демонструвати монотонні зміни всередині кожного об'єкта – значення поступово зростають від однієї границі до протилежної. Будь-які коливання порушують цю монотонність та ускладнюють постобробку.

Ще одним методом збільшення розмірносте є білінійна інтерполяція. Білінійна інтерполяція є детермінованим методом збільшення роздільної здатності, що не має параметрів для навчання. Для кожного пікселя у збільшеному зображенні значення обчислюється як зважена сума чотирьох найближчих сусідів з оригінальної карти, де ваги визначаються відстанями до цих сусідів. Математично, для пікселя з координатами (x, y) у збільшеному зображенні значення обчислюється за формулою:

$$f(x,y) = (1-\alpha)(1-\beta) f_{00} + \alpha(1-\beta) f_{10} + (1-\alpha)\beta f_{01} + \alpha\beta f_{11},$$

де f_{ij} – значення у чотирьох сусідніх пікселях оригінального зображення, а α та β – дробові частини координат (x, y) після масштабування до системи координат оригінального зображення.

Білінійна інтерполяція гарантовано не створює артефактів і забезпечує плавні переходи між пікселями. Кожен вихідний піксель формується однаковим чином, як зважена комбінація чотирьох сусідів, що усуває періодичні нерівномірності, характерні для транспонованої згортки. Однак білінійна інтерполяція має суттєве обмеження – вона не може адаптуватися до специфіки даних. Незалежно від контенту зображення, операція завжди виконує однакове згладжування. Це може бути недостатнім для задач, де необхідно відновлювати складні шаблони або різкі переходи.

Важливим параметром білінійної інтерполяції є режим вирівнювання кутів (align corners). При значенні true кутові пікселі вхідного та вихідного тензорів мають однакові значення, що є важливим для збереження границь об'єктів при сегментації. Координатна сітка вхідного та вихідного зображень вирівнюється таким чином, щоб центри кутових пікселів збігалися.

Для декодера відстаней у підблоці збільшення розмірності замість транспонованої згортки ми використовуємо альтернативний механізм на основі шару PixelShuffle (також відомий як sub-pixel convolution).

Розглянемо метод PixelShuffle [24]. PixelShuffle є операцією перестановки пікселів, що перетворює тензор з форми $(batch, C \times r^2, H, W)$ у тензор форми $(batch, C, H \times r, W \times r)$, де r – коефіцієнт збільшення (upscale factor). Операція працює наступним чином: спочатку кількість каналів збільшується за допомогою звичайної згортки у r^2 разів, а потім ці додаткові канали «розгортаються» у просторові координати.

Математично, для вхідного тензору X розміру $(B, C \times r^2, H, W)$ вихідний тензор Y розміру $(B, C, H \times r, W \times r)$ формується за правилом:

$$Y[b, c, h \cdot r + i, w \cdot r + j] = X[b, c \cdot r^2 + i \cdot r + j, h, w],$$

де $i, j \in 0,1,\ldots,r-1$.

Інтуїтивно, кожен «суперпіксель» розміром 1×1 з r^2 каналами у вхідному тензорі розгортається у блок $r \times r$ пікселів з одним каналом у вихідному. Для $r = 2$, що використовується в архітектурі InstanceCLIPSeg, чотири послідовних канали перетворюються у квадрат 2×2 пікселів.

На відміну від транспонованої згортки, яка генерує нові значення через інтерполяцію, що навчається, PixelShuffle лише переставляє існуючі значення, а генерація цих значень делегується попередній згортці.

PixelShuffle має кілька ключових переваг порівняно з транспонованою згорткою. По-перше, операція є детермінованою та не залежить від

параметрів, що навчаються. Це усуває можливість генерації артефактів операцією збільшення роздільної здатності. Усі артефакти, якщо вони виникають, є результатом роботи попередньої згортки, яку легше контролювати та ініціалізувати. По-друге, PixelShuffle забезпечує ефективніше використання параметрів. Транспонована згортка 4×4 з C вхідними та C вихідними каналами має $C^2\times16$ параметрів. Згортка 3×3 перед PixelShuffle з C вхідними та 4C вихідними каналами має $C\times4C\times9=36C^2$ параметрів, але вона також виконує нелінійне перетворення ознак, а не лише збільшення роздільної здатності. По-третє, PixelShuffle краще зберігає високочастотні деталі. Транспонована згортка є формою інтерполяції, яка природно згладжує вихід. PixelShuffle передає значення без модифікації, дозволяючи попередній згортці генерувати деталі, які будуть збережені в збільшеному виході.

При використанні PixelShuffle зі стандартною ініціалізацією ваг виникає специфічна проблема блочних артефактів. Методи ініціалізації Xavier [109] або Kaiming [110] генерують випадкові ваги, які не враховують структуру операції PixelShuffle. Канали, що відповідають різним позиціям у вихідному блоці 2×2, отримують незалежні випадкові ваги. Це призводить до блочних артефактів на початку навчання. Вихідне зображення має помітну структуру 2×2 блоків, де кожен підпіксель (позиція в блоці) має систематично відмінні характеристики від сусідів. Хоча мережа може навчитися компенсувати цю неузгодженість у процесі оптимізації, це уповільнює збіжність та може призводити до застрявання в локальних мінімумах.

Метод ICNR (Initialized Sub-Pixel Convolution) [111] був запропонований для вирішення проблеми блочних артефактів. Ключова ідея полягає у забезпеченні однакових початкових значень для всіх підпікселів, що усуває систематичні відмінності між ними на початку навчання. Алгоритм ICNR складається з трьох кроків. На першому кроці створюється «стиснене» ядро

(subkernel) з розмірністю (C_{out}, C_{in}, k, k). Це ядро ініціалізується стандартним методом, наприклад ініціалізацією Kaiming:

$$\sigma = \sqrt{\frac{2}{n_{out}}},$$

де $n_{out} = C_{out} \times k \times k$ – кількість вихідних з'єднань.

На другому кроці ініціалізоване ядро повторюється r^2 разів вздовж виміру вихідних каналів. Для $r = 2$ кожен зріз ядра копіюється 4 рази, створюючи ядро з розмірністю $(C_{out} \times 4, C_{in}, k, k)$. На третьому кроці результуюче ядро присвоюється вагам згорткового шару, який передує PixelShuffle. Оскільки канали, що відповідають одному вихідному пікселю (одній позиції в блоці 2×2), мають однакові початкові ваги, вихід PixelShuffle буде однорідним без блочних артефактів. Математично, якщо W – ваги згортки перед PixelShuffle розміром $(C \cdot r^2, C_{in}, k, k)$, то після ICNR ініціалізації:

$$W[c \cdot r^2 + i, :, :, :] = W_{sub}[c, :, :, :] \quad \forall i \in \{0, 1, ..., r^2 - 1\},$$

де W_{sub} – стиснене ядро розміром (C, C_{in}, k, k).

Варіації архітектури декодеру відстаней будуть детально розглянуті далі. Блоки уточнення (refinement blocks) додаються після кожної операції збільшення роздільної здатності для корекції раніше згаданих недоліків. Вони також згладжують артефакти від попереднього етапу, виконують адаптацію до збільшеної роздільної здатності через нелінійне перетворення, та відновлюють деталі, втрачені при збільшенні роздільної здатності. Базовий блок уточнення складається зі згортки з ядром 3×3, пакетної нормалізації та функції активації ReLU. Ядро 3×3 з відступом 1 зберігає просторові розміри, дозволяючи блоку фокусуватися на уточненні значень без зміни роздільної здатності. Рецептивне поле 3×3 достатнє для згладжування локальних нерівномірностей.

Розширений блок уточнення додає другу згортку 3×3 з власною нормалізацією та активацією. Послідовність двох згорток 3×3 еквівалентна одній згортці 5×5 з точки зору рецептивного поля, але має меншу кількість параметрів та додаткову нелінійність між шарами. Більше рецептивне поле дозволяє враховувати ширший контекст при уточненні значень.

Кількість блоків у декодері відстаней також становить чотири, що забезпечує послідовне збільшення роздільної здатності від 22×22 до 352×352 пікселів через проміжні розміри.

Вибір функції активації ReLU для обох декодерів обумовлений кількома факторами:

По-перше, ReLU є обчислювально ефективною функцією, що не потребує складних операцій обчислення експоненти чи ділення. Операція $\max(0,x)$ виконується за сталий час та легко розпаралелюється на GPU.

По-друге, ReLU забезпечує розріджену активацію (sparse activation), коли значна частина нейронів повертає нульове значення. Це сприяє кращому узагальненню моделі та зменшує ризик перенавчання.

По-третє, ReLU обмежує вихід невід'ємними значеннями, що відповідає фізичному змісту прогнозування величин. Теплова карта центрів відображає ймовірності, які за значенням є невід'ємними. Відстані до меж обмежувальної рамки також є невід'ємними для пікселів всередині об'єкта.

Пакетна нормалізація (Batch Normalization) [112] відіграє критичну роль у стабілізації процесу навчання глибоких мереж. Вона нормалізує активації всередині міні-пакету, віднімаючи середнє та ділячи на стандартне відхилення:

$$\hat{y} = \frac{y - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}},$$

де μ_B та σ_B^2 – середнє та дисперсія по міні-пакету, ε – мала константа для числової стабільності.

Після нормалізації застосовується афінне перетворення з параметрами γ та β, що навчаються:

$$z = \gamma \hat{y} + \beta .$$

Пакетна нормалізація вирішує проблему внутрішнього коваріантного зсуву (internal covariate shift), коли розподіл активацій змінюється протягом навчання через оновлення параметрів попередніх шарів. Це дозволяє використовувати вищі швидкості навчання та робить мережу менш чутливою до ініціалізації ваг.

У контексті декодерів для сегментації пакетна нормалізація також сприяє узгодженню масштабів активацій між різними шарами. При послідовному збільшенні роздільної здатності без нормалізації, активації можуть неконтрольовано зростати або зменшуватися, що ускладнює навчання.

Вихідні шари обох декодерів відрізняються від внутрішніх блоків. Для декодера центрів вихідний шар – це згортка 1×1, що зменшує кількість каналів до одного, з наступною сигмоїдною функцією активації для отримання значень у діапазоні [0, 1], що інтерпретуються як ймовірності.

Для декодера відстаней вихідний шар – це згортка 3×3, що формує чотири канали (по одному для кожної границі).

3.4 Механізм умовної модуляції на основі текстового запиту

Центральним механізмом, що забезпечує здатність моделі знаходити об’єкти за текстовим описом, є умовна модуляція візуальних ознак на основі текстового запиту. Цей механізм успадкований від CLIPSeg та адаптований для використання у двох декодерах InstanceCLIPSeg.

Умовна модуляція реалізована за допомогою механізму FiLM (Feature-wise Linear Modulation). FiLM виконує поелементне афінне перетворення ознак на основі умовного вектора:

$$\text{FiLM}(F, c) = \gamma(c) \odot F + \beta(c),$$

де F – тензор візуальних ознак, c – умовний вектор (текстовий ембединг), $\gamma(c)$ та $\beta(c)$ – функції, що генерують параметри масштабування та зсуву на основі умовного вектора, $\odot$ – поелементне множення.

У контексті InstanceCLIPSeg функції γ та β реалізовані як лінійні проекції текстового ембедингу:

$$\gamma(c) = W_\gamma \cdot c + b_\gamma \quad \beta(c) = W_\beta \cdot c + b_\beta,$$

де W_γ, W_β – матриці ваг, b_γ, b_β – вектори зсуву, що навчаються під час тренування.

FiLM модуляція застосовується після кожного блоку TransformerEncoderLayer у CLIP енкодері та перед подачею ознак до декодерів. Це дозволяє текстовому запиту впливати на візуальні представлення на всіх рівнях ієрархії, від низькорівневих просторових ознак до високорівневих семантичних.

Механізм модуляції можна інтерпретувати як м’який механізм уваги (soft attention), де текстовий запит визначає, які візуальні ознаки підсилюються ($\gamma > 1$ або $\beta > 0$), а які придушуються ($\gamma < 1$ або $\beta < 0$). Регіони зображення, семантично пов’язані із запитом, отримують вищі значення активацій, що полегшує їх виявлення декодерами.

Важливою особливістю є те, що обидва декодери (центрів та відстаней) отримують однаковим чином модульовані візуальні ознаки. Це забезпечує узгодженість між тепловою картою центрів та картами відстаней – центри та

відстані передбачаються для одних і тих самих об’єктів, виявлених на основі текстового запиту.

Спільний семантичний простір CLIP, де розташовані як візуальні, так і текстові ембединги, є фундаментом для ефективної роботи механізму модуляції. Завдяки контрастивному навчанню на масштабному наборі пар зображення-текст, семантично пов’язані концепції розташовуються поблизу у цьому просторі. Тому текстовий ембединг запиту «a photo of a car» буде близьким до візуальних ембедингів областей зображення, що містять автомобілі.

Ця близькість транслюється у схожі шаблони модуляції: параметри γ та β, обчислені для текстового ембедингу автомобіля, будуть підсилювати візуальні ознаки, характерні для регіонів з автомобілями. Модель навчається встановлювати цю відповідність під час дотренування на наборах даних для сегментації.

3.5 Алгоритм постобробки результатів сегментації

Після отримання прогнозів від обох декодерів – теплової карти центрів та чотириканальної карти відстаней – виконується етап постобробки для формування фінальних масок окремих екземплярів об’єктів. Постобробка є необхідним компонентом архітектури, оскільки вихід нейронної мережі представляє собою неперервні карти значень, а не дискретні маски з унікальними ідентифікаторами екземплярів.

Алгоритм постобробки складається з п’яти послідовних етапів, кожен з яких виконує специфічну функцію у формуванні кінцевого результату.

На першому етапі виконується порогова фільтрація теплової карти центрів. Застосовується поріг θ_{center} для відсіювання пікселів з низькою ймовірністю бути центром об’єкта:

$$M_{threshold}(x,y)=\begin{cases}1, & \text{якщо } H_{centers}(x,y)>\theta_{center}\\ 0, & \textit{інакше}\end{cases},$$

де $H_{centers}$ – теплова карта центрів, $M_{threshold}$ – бінарна маска потенційних центрів. Значення порогу θ_{center} є гіперпараметром, що балансує між чутливістю (здатністю виявляти всі об’єкти) та специфічністю (уникнення хибних спрацювань).

На другому етапі до відфільтрованої карти центрів застосовується немаксимальне придушення (Non-Maximum Suppression, NMS). NMS виділяє локальні максимуми, усуваючи близько розташовані піки та залишаючи лише найбільш ймовірні центри. Для кожного пікселя (x,y) перевіряється, чи є його значення максимальним у локальному вікні розміром $w_{nms}\times w_{nms}$:

$$\mathrm{NMS}(x,y)=\begin{cases}1, & \text{якщо } H_{centers}(x,y)=\max\limits_{|x'-x|\le\frac{w_{nms}}{2},|y'-y|\le\frac{w_{nms}}{2}} H_{centers}(x',y')\\ 0, & \textit{інакше}\end{cases}.$$

Результатом є множина координат виявлених центрів $C=(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)$, де n – кількість знайдених об’єктів.

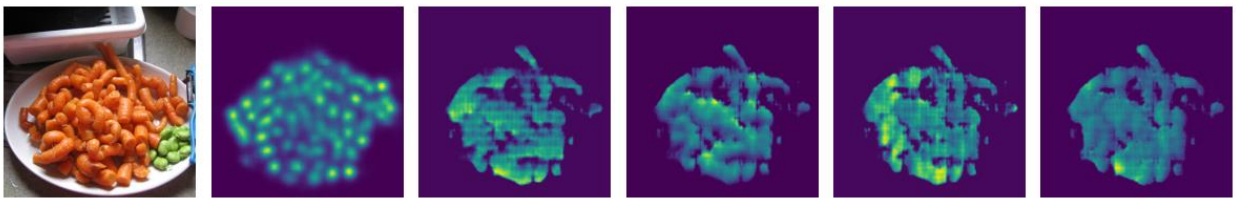

Рисунок 3.3 – Результат роботи мережі: вхідне зображення, знайдені центри та чотири карти відстаней

На третьому етапі обчислюються обмежувальні рамки для кожного виявленого центру на основі передбачених відстаней. Для центру (x_c,y_c) координати обмежувальної рамки визначаються як:

$$y_{min} = y_c - D_{top}(x_c, y_c) \quad y_{max} = y_c + D_{bottom}(x_c, y_c) \quad x_{min} = x_c - D_{left}(x_c, y_c) \quad x_{max} = x_c + D_{right}(x_c, y_c),$$

де $D_{top}, D_{bottom}, D_{left}, D_{right}$ – чотири канали карти відстаней.

Через можливий шум у результатах роботи мережі використовуються не лише значення безпосередньо в центрі, а й усереднені значення в околі центру радіусом r_{avg}:

$$\bar{D}k(x_c, y_c) = \frac{1}{|N|} \sum (x', y') \in N D_k(x', y'),$$

де $N = (x', y') : |x' - x_c| \le r_{avg} \wedge |y' - y_c| \le r_{avg}$ – множина пікселів в околі центру, $k \in top, bottom, left, right$.

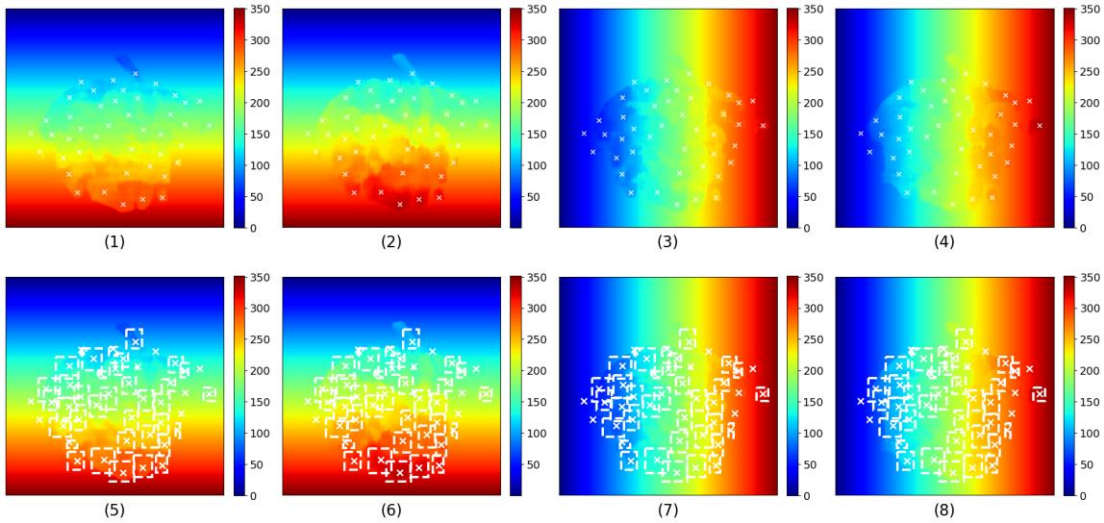

Рисунок 3.4 – Виявлення об’єктів за передбаченими відстанями до їхніх країв та обчислення обмежувальних рамок

На четвертому етапі будуються координатні карти для перетворення відстаней до границь у просторові координати. Створюються дві базові сітки: горизонтальна G_x розміром $H \times W$, де $G_x(i, j) = j$, та вертикальна G_y розміром $H \times W$, де $G_y(i, j) = i$.

Для кожного каналу відстаней обчислюється карта абсолютних координат границь:

$$Map_{top} = G_y - D_{top}\ Map_{bottom} = G_y + D_{bottom}\ Map_{left} = G_x - D_{left}\ Map_{right} = G_x + D_{right}\ .$$

Після цього перетворення пікселі, що належать одному об’єкту, матимуть однакові (з точністю до похибки прогнозування) значення на відповідних картах координат границь. Це дозволяє групувати пікселі за належністю до екземплярів.

На п’ятому етапі виконується формування масок екземплярів шляхом кластеризації пікселів навколо виявлених центрів. Для кожного центру (x_c, y_c) з обчисленою обмежувальною рамкою $[x_{min}, y_{min}, x_{max}, y_{max}]$ виконується наступний алгоритм:

Ініціалізація – маска екземпляра $M_{instance}$ встановлюється рівною 1 у позиції центру та 0 в інших позиціях. Формується черга Q для обходу в ширину (BFS), що спочатку містить лише координати центру.

Розширення регіону – поки черга Q не порожня, витягується поточний піксель (x, y). Для кожного сусіднього пікселя (x', y') з 4-зв’язного околу перевіряються умови приєднання до маски:

1) піксель ще не належить масці: $M_{instance}(x', y') = 0$;

2) піксель знаходиться в межах обмежувальної рамки: $x_{min} \le x' \le x_{max}$ та $y_{min} \le y' \le y_{max}$;

3) значення на картах координат границь близькі до очікуваних:

$$|Map_{top}(x', y') - y_{min}| < \delta \quad |Map_{bottom}(x', y') - y_{max}| < \delta \quad |Map_{left}(x', y') - x_{min}| < \delta,$$
$$|Map_{right}(x', y') - x_{max}| < \delta$$

де δ – допустиме відхилення, що є гіперпараметром алгоритму.

Якщо всі умови виконані, піксель додається до маски ($M_{instance}(x', y') = 1$) та до черги Q.

Врахування зв'язності пікселів гарантує, що до одного екземпляра належать лише просторово суміжні пікселі. Навіть якщо два просторово розділені регіони мають подібні значення на картах відстаней, вони не будуть об'єднані в один екземпляр.

Рисунок 3.5 – Результат розбиття знайдених об'єктів на окремі екземпляри

Параметр допустимого відхилення δ контролює чутливість алгоритму до шуму у прогнозуваннях. Менше значення δ призводить до суворішого критерію приєднання пікселів, що може викликати недосегментацію (under-segmentation) об'єктів з неточними значеннями. Більше значення δ робить критерій м'якішим, що може викликати пересегментацію (over-segmentation) або злиття близько розташованих об'єктів.

Алгоритм постобробки має кілька важливих властивостей. Він здатний виділяти об'єкти, що перекриваються, якщо перекриття коректно відображене

у картах відстаней. Кожен центр обробляється незалежно, тому пікселі у зоні перекриття можуть бути призначені різним екземплярам на основі їх значень на картах відстаней.

Обчислювальна складність алгоритму є лінійною відносно кількості пікселів зображення та кількості виявлених центрів: $O(n \times H \times W)$, де n – кількість центрів, $H \times W$ – роздільна здатність зображення. Кожен піксель обробляється не більше одного разу для кожного екземпляра під час обходу в ширину.

3.6 Стратегія навчання та функції втрат

Процес навчання моделі InstanceCLIPSeg базується на стратегії дотренування (fine-tuning) попередньо навченої моделі CLIPSeg. Ваги оригінальної CLIPSeg завантажуються у відповідні компоненти архітектури: CLIP енкодер зображення, CLIP енкодер тексту та шари умовної модуляції. Нові компоненти – декодери центрів та відстаней – ініціалізуються випадковими вагами або за обраними алгоритмами [113].

Ваги CLIP енкодерів залишаються без навчання упродовж усього процесу. Це рішення зумовлене необхідністю збереження мультимодальних представлень, набутих під час масштабного попереднього навчання CLIP. Дотренування енкодерів на відносно невеликих наборах даних для сегментації могло б призвести до катастрофічного забування загальних візуально-текстових відповідностей або до перенавчання.

Навчаються такі компоненти моделі:

1) шари FiLM модуляції (параметри γ та β);
2) усі шари декодера центрів;
3) усі шари декодера відстаней.

Для оптимізації використовується алгоритм AdamW. Розмір міні-пакету встановлено рівним 64. Такий розмір пакету забезпечує стабільні оцінки градієнтів та ефективне використання паралелізму GPU.

Для декодера центрів використовується зважена середньоквадратична похибка (Weighted Mean Square Error) як функція втрат:

$$\mathcal{L}_{centers} = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot (\hat{y}_i - y_i)^2,$$

де $\hat{y}_i$ – передбачене значення, y_i – цільове значення, w_i – вага для пікселя i, N – загальна кількість пікселів.

Зважування необхідне через суттєвий дисбаланс класів: пікселі центрів об'єктів (кодовані гаусівськими піками) займають незначну частку зображення порівняно з фоном. Без зважування модель могла б мінімізувати втрати, просто передбачаючи нульові значення для всіх пікселів.

Вага для пікселя визначається на основі його цільового значення:

$$w_i = \begin{cases} w_{fg}, & якщо y_i > 0 \text{ (піксель належить центру)} \\ w_{bg}, & \text{інакше (фон)} \end{cases}.$$

У реалізації використовуються значення $w_{fg} = 10$ та $w_{bg} = 1$. Це означає, що помилка у передбаченні центру штрафується у 10 разів сильніше, ніж помилка для фонового пікселя.

Для декодера відстаней використовується зважена L1 функція втрат (Weighted L1 Loss):

$$\mathcal{L}_{offsets} = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot |\hat{y}_i - y_i|.$$

L1 (Mean Absolute Error) обрана замість L2 (Mean Square Error) через її кращу стійкість до викидів. Функція L2 штрафує пропорційно квадрату помилки, що надає надмірну вагу великим відхиленням. У картах відстаней можуть виникати значні локальні відхилення через складну геометрію об’єктів або неточності у розмітці, і L1 краще справляється з такими випадками.

Зважування для декодера відстаней також спрямоване на боротьбу з дисбалансом класів, але з іншою семантикою:

$$w_i = \begin{cases} w_{obj}, & \text{якщо піксель } i \text{ належить об'єкту} \\ 1, & \text{інакше} \end{cases}.$$

Не нульова вага для фонових пікселів означає, що модель також штрафується за прогнозування на фоні.

Загальна функція втрат є сумою втрат обох декодерів:

$$\mathcal{L}_{total} = \mathcal{L}_{centers} + \lambda \cdot \mathcal{L}_{offsets},$$

де λ – коефіцієнт балансування, що контролює відносну важливість двох компонентів втрат.

Навчання проводиться протягом 20 епох. Для управління швидкістю навчання використовується планувальник SequentialLR, що дозволяє послідовно застосовувати різні стратегії:

Фаза розігріву (warmup) – перші кілька епох використовується LinearLR, що лінійно збільшує швидкість навчання від малого початкового значення до цільового:

$$lr(t) = lr_{initial} + (lr_{target} - lr_{initial}) \cdot \frac{t}{T_{warmup}},$$

де t – поточна ітерація, T_{warmup} – тривалість фази розігріву.

Фаза розігріву запобігає різким змінам ваг на початку навчання. Випадково ініціалізовані ваги декодерів можуть генерувати великі градієнти, і високе початкове значення швидкості навчання могло б дестабілізувати оптимізацію.

Основна фаза – після розігріву застосовується ExponentialLR зі зменшенням швидкості навчання:

$$lr(t) = lr_{target} \cdot \gamma^t,$$

де $\gamma < 1$ – коефіцієнт згасання. У реалізації швидкість навчання зменшується від 1×10^{-3} до 1×10^{-4}.

Експоненціальне зменшення дозволяє моделі робити більші кроки на початку для швидкого наближення до оптимуму, а потім менші кроки для точного налаштування у його околі.

Аугментація даних є важливим компонентом стратегії навчання. Застосовуються нижченаведені трансформації.

Обертання (Rotate) – зображення разом з масками та обмежувальними рамками обертається на випадковий кут. Це робить модель інваріантною до орієнтації об’єктів.

Відображення (Flip) – горизонтальне та вертикальне відображення з ймовірністю 0,5. Збільшує варіативність даних без зміни семантичного змісту.

Випадкове масштабування (RandomScale) – зображення масштабується з випадковим коефіцієнтом у діапазоні [1, 2]. Масштабування з коефіцієнтом більше 1 особливо важливо, оскільки дозволяє моделі навчитися працювати з об’єктами різних розмірів відносно роздільної здатності зображення.

Усі геометричні перетворення застосовуються синхронно до зображення, масок сегментації та координат обмежувальних рамок для збереження відповідності між входами та цільовими значеннями.

Для навчання використовуються два набори даних: LVIS (Large Vocabulary Instance Segmentation) [114] та PhraseCut [115].

Набір LVIS містить понад 164000 зображень з детальною попіксельною розміткою для більш ніж 1200 категорій об’єктів. Категорії організовані ієрархічно та охоплюють широкий спектр побутових предметів, тварин, транспорту тощо.

Набір PhraseCut містить близько 77000 зображень з текстовими описами для посилання на об’єкти. На відміну від LVIS, де кожен об’єкт має категоріальну мітку, у PhraseCut об’єкти описуються фразами природної мови, що можуть включати атрибути («великий червоний автомобіль»), просторові відношення («людина зліва від дерева») та контекстуальні описи («жінка, що тримає каву»).

Комбінування цих наборів забезпечує баланс між двома аспектами навчання: LVIS надає великий обсяг різноманітних екземплярів для навчання точної сегментації, тоді як PhraseCut навчає модель розуміти складні текстові описи з контекстуальними залежностями.

На основі аналізу розподілу розмірів об’єктів у наборах даних встановлено обмеження: модель навчається лише на об’єктах, площа яких після масштабування до роздільної здатності 352×352 становить 100 квадратних пікселів або більше. Об’єкти меншого розміру виключаються з навчання через обмежену здатність архітектури їх знаходити та сегментувати.

3.7 Висновки за розділом

У третьому розділі представлено розробку архітектури мультимодальної моделі InstanceCLIPSeg та методів її навчання. За результатами розділу сформульовані такі висновки:

1) Запропоновано архітектуру моделі InstanceCLIPSeg, яка базується на модифікації CLIPSeg та інтеграції концепцій Panoptic-DeepLab і PRN.

Встановлено, що використання чотирьохкомпонентної структури (енкодери зображення та тексту, декодери центрів та відстаней) дозволяє реалізувати сегментацію екземплярів в умовах open-vocabulary без обмеження фіксованим набором класів.

2) Розроблено спеціалізовані архітектури декодерів для відновлення просторової роздільної здатності. Показано, що використання механізму PixelShuffle у декодері відстаней та комбінації згорток у декодері центрів дозволяє уникнути артефактів типу «шахової дошки» та забезпечити точне відновлення роздільної здатності від 22×22 до 352×352 пікселів.

3) Розроблено алгоритм постобробки результатів висхідної (bottom-up) сегментації. Визначено, що поєднання порогової фільтрації, NMS та кластеризації з урахуванням просторової зв'язності дозволяє ефективно формувати фінальні маски екземплярів, забезпечуючи гнучкий баланс між чутливістю та специфічністю виявлення.

4) Обґрунтовано підхід до неявного знаходження фону, як інверсії об'єднаної маски виявлених об'єктів. Встановлено, що така стратегія дозволяє спростити архітектуру декодерів та знизити обчислювальні витрати, покладаючись на семантичну вибірковість CLIP-енкодера.

5) Сформовано стратегію навчання моделі, що передбачає використання компонентів CLIP та розроблених зважених функцій втрат. Показано, що комбінування наборів даних LVIS та PhraseCut забезпечує здатність моделі як до точної сегментації великого словника категорій, так і до розуміння складних текстових запитів.

6) Реалізовано одноетапний підхід до сегментації екземплярів, який, на відміну від двоетапних методів, виконує виявлення та сегментацію за один прохід мережі, що забезпечує спрощення архітектури системи та потенційні переваги у швидкості обробки.

4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

4.1 Механізми відновлення просторової роздільної здатності

Декодер відстаней є ключовим компонентом архітектури InstanceCLIPSeg, що відповідає за генерацію просторових карт, які кодують геометричну інформацію про розташування кожного пікселя відносно границь об’єкта, якому він належить. На відміну від декодера центрів, який генерує карту з локалізованими піками в центрах мас об’єктів, декодер відстаней повинен передбачати неперервні числові значення для кожного пікселя зображення, що створює суттєво складнішу задачу регресії. Важливою особливістю цільового представлення є те, що значення відстаней визначені лише для пікселів, що належать об’єктам. Для пікселів фону цільові значення формально дорівнюють нулю, хоча семантично вони не мають змісту. Це створює додаткове навантаження на модель, яка повинна одночасно навчитися розрізняти об’єкти від фону та точно передбачати відстані для пікселів об’єктів.

У цьому розділі представлено систематичне дослідження різних архітектурних рішень для декодера відстаней та їх вплив на якість передбачень. Досліджено механізми відновлення просторової роздільної здатності, роль позиційного кодування, стратегії багаторівневого злиття ознак та їх комбінації.

У попередньому розділі були проаналізовано методи збільшення просторової роздільної здатності для декодеру. Зазначені їх переваги та недоліки. Додатково слід зазначити, що крім звичайної згортки, є і координатна (CoordConv) [116], яка може враховувати положення об’єкта при прогнозуванні.

Вибір між координатною та звичайною згорткою в блоці уточнення залежить від етапу декодування. Стандартні згорткові шари мають властивість трансляційної еквіваріантності: вони застосовують однакові ваги до всіх

позицій вхідної карти ознак, і результат не залежить від абсолютного положення в зображенні. Ця властивість є корисною для багатьох задач розпізнавання, де об’єкт має ідентифікуватися незалежно від його розташування у кадрі. Однак для задачі прогнозування відстаней до границь трансляційна еквіваріантність є суттєвим обмеженням. Відстань від пікселя до границі обмежувальної рамки залежить не лише від локального контексту (текстури, границі об’єкта), але й від абсолютного положення пікселя в всередині об’єкта. Наприклад, для пікселя в лівому верхньому куті об’єкта відстань до верхньої та лівої границь буде невеликою незалежно від того, як виглядає його локальний контекст. Стандартна згортка не має доступу до цієї позиційної інформації.

Координатні згортки (CoordConv) вирішують проблему відсутності позиційної інформації шляхом явного додавання просторових координат до вхідних ознак. Перед застосуванням згорткової операції до вхідного тензора додаються два додаткових канали, що містять нормалізовані координати кожного пікселя. Модуль генерації координат створює координатні канали таким чином. Для карти ознак розміром $H \times W$ створюються дві координатні матриці. Матриця y-координат G_y має однакові значення вздовж кожного рядка, які лінійно змінюються від -1 (верхній край) до $+1$ (нижній край):

$$G_y[i, j] = \frac{2i}{H-1} - 1, \quad i \in \{0,1,...,H-1\}.$$

Матриця x-координат G_x має однакові значення вздовж кожного стовпця, які змінюються від -1 (лівий край) до $+1$ (правий край):

$$G_x[i, j] = \frac{2j}{W-1} - 1, \quad j \in \{0,1,...,W-1\}.$$

Нормалізація до діапазону [-1, 1] забезпечує стабільність числових значень незалежно від абсолютного розміру карти ознак. Це є важливим при роботі з зображеннями різної роздільної здатності або при зміні роздільної здатності під час декодування. Координатні матриці розширюються до розмірності пакету (batch) шляхом повторення та об'єднуються з вхідним тензором вздовж канального виміру. Якщо вхідний тензор мав форму $B \times C \times H \times W$, результат має форму $B \times (C+2) \times H \times W$. Наступна згортка враховує ці додаткові канали, отримуючи можливість формувати вихід, що залежить від позиції.

Розглянемо формальне обґрунтування ефективності координатних згорток для задачі прогнозування відстаней. Нехай $f : \mathbb{R}^2 \to \mathbb{R}^4$ – функція, що відображає координати пікселя (x, y) у вектор відстаней до чотирьох границь $(d_{top}, d_{bottom}, d_{left}, d_{right})$. Для пікселя всередині об'єкта з обмежувальною рамкою $[x_{min}, y_{min}, x_{max}, y_{max}]$ цільові відстані визначаються як:

$$d_{top}(x, y) = y - y_{min} \quad d_{bottom}(x, y) = y_{max} - y \quad d_{left}(x, y) = x - x_{min} \quad d_{right}(x, y) = x_{max} - x .$$

Кожна з цих функцій є лінійною відносно координат пікселя. Стандартна згортка, що оперує лише значеннями ознак без доступу до координат, не може безпосередньо моделювати цю лінійну залежність. Вона може лише апроксимувати її через складні нелінійні комбінації локальних ознак.

Координатна згортка, навпаки, має прямий доступ до координат (x, y). Ваги, що відповідають координатним каналам, можуть безпосередньо моделювати лінійну залежність:

$$\hat{d}_{top}(x, y) = w_y \cdot y + w_0 + g(F(x, y)) ,$$

де w_y та w_0 – ваги для координатного каналу та зміщення, $g(F(x,y))$ – внесок від локальних ознак $F(x,y)$.

Таким чином, координатна згортка може розділити задачу на дві частини: моделювання позиційної залежності (через координатні канали) та моделювання залежності від локального контексту (через канали ознак). Це спрощує оптимізацію та покращує генералізацію.

Вибір етапів декодера для застосування координатних згорток є важливим архітектурним рішенням, що впливає на баланс між якістю передбачень та обчислювальними витратами. На ранніх етапах декодування, коли роздільна здатність низька (22×22, 44×44), координатна інформація є критичною для встановлення зв'язку між позицією та очікуваними відстанями. На цих етапах кожен піксель карти ознак відповідає значній області вхідного зображення (16×16 або 8×8 пікселів) і локальний контекст не надає достатньої інформації про абсолютне положення. На пізніх етапах, коли роздільна здатність наближається до цільової (176×176, 352×352), локальний контекст стає більш інформативним. Кожен піксель карти ознак відповідає меншій області вхідного зображення і локальні ознаки (границі об'єктів, текстурні переходи) можуть надавати достатню інформацію для уточнення відстаней. На цих етапах координатні згортки можуть бути замінені звичайними для зменшення обчислювальних витрат.

Вони збільшують кількість параметрів та обчислювальні витрати порівняно зі звичайними згортками. Для згортки з ядром $k\times k$, C_{in} вхідними та C_{out} вихідними каналами кількість параметрів становить:

1) звичайна згортка: $C_{in}\times C_{out}\times k^2 + C_{out}$;
2) координатна згортка: $(C_{in}+2)\times C_{out}\times k^2 + C_{out}$.

Збільшення кількості параметрів становить $2\times C_{out}\times k^2$, що є відносно невеликим для типових конфігурацій. Наприклад, для згортки 3×3 з 64 вихідними каналами додається лише $2\times64\times9=1152$ параметри.

Обчислювальна складність також збільшується пропорційно до збільшення кількості вхідних каналів. Для карти ознак розміром $H \times W$ кількість операцій множення-додавання становить:

1) звичайна згортка: $C_{in} \times C_{out} \times k^2 \times H \times W$;

2) координатна згортка: $(C_{in} + 2) \times C_{out} \times k^2 \times H \times W$.

Відносне збільшення обчислювальних витрат дорівнює $(C_{in} + 2) / C_{in}$, що є значним для малих значень C_{in} (наприклад, 3% для $C_{in} = 64$) і незначним для великих.

4.2 Багаторівневе злиття ознак та допоміжні з'єднання

Базова архітектура декодера відстаней використовує лише фінальний вихід базового декодера, який інтегрує інформацію з різних шарів енкодера в єдине представлення розміром $B \times 64 \times 22 \times 22$. Хоча це представлення є семантично достатнім, процес злиття може втрачати деякі деталі, особливо низькорівневі просторові ознаки з ранніх шарів енкодера.

Архітектури з багаторівневим злиттям ознак вирішують цю проблему шляхом збереження та окремого використання активацій з різних шарів енкодера. Це дозволяє декодеру відстаней отримати доступ як до високорівневих семантичних ознак (з глибоких шарів), так і до низькорівневих просторових деталей (з початкових шарів). Vision Transformer, що використовується як енкодер зображення в архітектурі CLIP, складається з послідовності трансформерних блоків. На відміну від згорткових мереж, де роздільна здатність ознак поступово зменшується, Vision Transformer зберігає однакову роздільну здатність токенів на всіх рівнях (22×22 для вхідного зображення 352×352). Проте семантичний зміст ознак суттєво змінюється вздовж глибини мережі. Ранні шари (наприклад, 3-й блок) зберігають більше низькорівневої інформації: границі, текстури, локальні шаблони. Глибокі шари (наприклад, 7-й та 12-й блоки) містять більш абстрактні семантичні

ознаки: категоріальну інформацію, контекстні залежності, глобальні властивості об’єктів. Для задачі прогнозування відстаней до границь корисними є обидва типи інформації. Низькорівневі ознаки допомагають точно локалізувати границі об’єктів. Високорівневі ознаки допомагають визначити, які пікселі належать одному об’єкту, та забезпечують семантичну узгодженість результатів.

Перша стратегія багаторівневого злиття передбачає інтеграцію всіх джерел ознак на початку декодування. Цей підхід концептуально подібний до архітектури SegFormer, де ознаки з різних рівнів ієрархічного енкодера об’єднуються перед фінальним декодуванням.

Декодер з одноразовим злиттям отримує три джерела інформації:

1) основну семантичну карту з виходу базового декодера $B \times 64 \times 22 \times 22$;

2) активації з 7-го шару енкодера (проміжний рівень);

3) активації з 3-го шару енкодера (ранній рівень).

Кожне джерело спочатку проєктується в спільний простір розмірністю d_{embed} каналів через окремі проєкційні шари. Проєкція основної карти використовує згортку 3×3, що надає додаткове згладжування та враховує локальний контекст. Проєкції допоміжних з’єднань використовують згортки 1×1, які виконують лише канальне перетворення без зміни просторової структури:

$$F_{main} = \mathrm{BN}(\mathrm{Conv}_{3\times3}(X_{semantic})) \quad F_{L7} = \mathrm{BN}(\mathrm{Conv}_{1\times1}(X_{L7})) \quad F_{L3} = \mathrm{BN}(\mathrm{Conv}_{1\times1}(X_{L3})).$$

Використання окремих шарів пакетної нормалізації для кожної гілки дозволяє адаптуватися до різних статистичних характеристик кожного джерела ознак. Активації з різних шарів енкодера можуть мати суттєво різні розподіли значень, і спільна нормалізація могла б порушити цей баланс. Три

нормалізовані тензори об'єднуються вздовж канального виміру $F_{concat} = [F_{main}; F_{L7}; F_{L3}]$, утворюючи тензор розміром $B \times 3d_{embed} \times 22 \times 22$.

Блок злиття обробляє об'єднані ознаки для інтеграції інформації з різних джерел. Структура блоку містить координатну згортку для врахування позиційної інформації, пакетну нормалізацію, активацію ReLU, звичайну згортку для подальшого зменшення каналів, та ще одну пару нормалізація-активація:

$$F_{fused} = \text{ReLU}(\text{BN}[\text{Conv}_{3\times3}\{\text{ReLU}(\text{BN}[\text{CoordConv}_{3\times3}(F_{concat})])\}]).$$

Координатна згортка на вході блоку злиття є ключовим елементом: вона дозволяє мережі враховувати позицію при визначенні, яким чином комбінувати ознаки з різних джерел. Наприклад, для пікселів біля границі об'єкта може бути важливіша інформація з ранніх шарів (де границі чіткіші), тоді як для центральних пікселів – семантична інформація з глибоких шарів. Після злиття виконується послідовність етапів збільшення роздільної здатності через чотири блоки, кожен з яких подвоює просторові розміри та потенційно зменшує кількість каналів. Конфігурація каналів типово становить: $128 \to 64 \to 32 \to 16 \to 16$.

Друга стратегія – ієрархічні допоміжні з'єднання. Концептуально базується на архітектурі U-Net, адаптованій для специфіки CLIP енкодера. На відміну від гібридного підходу, де всі джерела ознак інтегруються одноразово на початку, ієрархічна архітектура підтримує постійний зв'язок між енкодером та декодером на різних етапах збільшення роздільної здатності. Класична архітектура U-Net передбачає з'єднання між рівнями енкодера та декодера з однаковою просторовою роздільною здатністю. Однак CLIP ViT зберігає однакову роздільну здатність токенів (22×22) на всіх рівнях, що унеможливлює пряме застосування класичної схеми. Тому допоміжні

з’єднання інтегруються на різних етапах декодування з відповідним масштабуванням.

Важливим архітектурним рішенням є вибір між використанням зменшених активацій (64 канали) та «сирих» активацій з енкодера (768 каналів). Активації зниженої розмірності вже пройшли попередню обробку в базовому декодері, що зменшує обчислювальні витрати та забезпечує узгодженість з основним потоком даних. «Сирі» активації зберігають більше інформації, особливо для низькорівневих ознак, але потребують додаткових проєкційних шарів для зменшення розмірності.

При використанні «сирих» активацій проєкційні шари мають двоетапну структуру для забезпечення нелінійного перетворення:

$$F_{proj} = \mathrm{BN}(\mathrm{Conv}_{1\times1}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{1\times1}(X_{raw}))))), \tag{4.1}$$

де перша згортка зменшує розмірність з 768 до 128 каналів, а друга – з 128 до 64. Двоетапна проєкція $768 \to 128 \to 64$ замість прямої $768 \to 64$ краще зберігає інформацію через додаткову нелінійність.

Структура декодера з ієрархічними з’єднаннями складається з чотирьох етапів збільшення роздільної здатності з різною конфігурацією допоміжних з’єднань.

Перший етап ($22\times22 \to 44\times44$). Основна гілка містить семантичну карту. Допоміжне з’єднання береться з 7-го шару енкодера, проєктується та додається до основної гілки перед блоком збільшення роздільної здатності. Роздільна здатність допоміжного з’єднання (22×22) відповідає роздільній здатності основної гілки, тому масштабування не потрібне.

Другий етап ($44\times44 \to 88\times88$). Допоміжне з’єднання береться з 3-го шару енкодера. Оскільки роздільна здатність допоміжного з’єднання (22×22) не відповідає роздільній здатності основної гілки (44×44), виконується білінійна інтерполяція для узгодження розмірів перед об’єднанням.

Третій та четвертий етапи. ($88\times88 \rightarrow 176\times176 \rightarrow 352\times352$). Допоміжні з'єднання не використовуються. На цих етапах інформація з енкодера вже інтегрована на попередніх етапах, і основна роль декодера полягає в уточненні деталей на основі локального контексту.

Логіка вибору рівнів для допоміжних з'єднань зумовлена кількома факторами. По-перше, активації CLIP енкодера мають фіксовану роздільну здатність 22×22. На перших двох етапах ця роздільна здатність близька до роздільної здатності основного потоку (22×22 та 44×44), і допоміжні з'єднання можуть бути ефективно інтегровані з мінімальною інтерполяцією. На пізніших етапах (88×88, 176×176) потрібна суттєва інтерполяція, яка може вносити артефакти та розмивати корисну інформацію.

По-друге, інформація з енкодера є найціннішою на ранніх етапах декодування, коли мережа формує базову просторову структуру передбачень. Глибокі шари енкодера (7-й) містять семантичну інформацію про належність пікселів до об'єктів, що критично для початкового етапу. Ранні шари (3-й) містять інформацію про границі та текстури, що важливо для уточнення на другому етапі. На пізніх етапах основна структура вже встановлена, і роль декодера зводиться до уточнення деталей на основі локального контексту.

По-третє, обмеження кількості допоміжних з'єднань зменшує складність моделі та ризик перенавчання. Кожне допоміжне з'єднання додає параметри в проєкційних шарах та збільшує кількість каналів у блоках збільшення роздільної здатності. При використанні допоміжних з'єднань блок збільшення роздільної здатності модифікується для прийому додаткового входу. Об'єднання виконується після операції збільшення роздільної здатності основної гілки та перед першою згорткою уточнення.

Кількість вхідних каналів першої згортки в блоці уточнення збільшується на кількість каналів допоміжного з'єднання. Наприклад, якщо основна гілка має 64 канали, а допоміжне з'єднання – також 64 канали, перша згортка отримує 64 + 64 = 128 каналів.

Масштабування допоміжного з'єднання, коли його роздільна здатність не відповідає роздільній здатності основної гілки, виконується за допомогою білінійної інтерполяції.

Дві стратегії багаторівневого злиття: одноразове злиття та ієрархічні з'єднання, – мають різні характеристики та області застосування. Одноразове злиття є концептуально простішим та обчислювально ефективнішим. Усі джерела ознак обробляються на початку декодування, і подальші етапи працюють лише з об'єднаним представленням. Це забезпечує чітке розділення між фазою злиття та фазою декодування. Однак одноразове злиття може втрачати деталі при обробці інформації з різних рівнів через єдиний блок, особливо якщо характеристики ознак суттєво відрізняються.

Ієрархічні з'єднання забезпечують гнучкішу інтеграцію інформації. Допоміжні з'єднання додаються на тих рівнях, де їхня інформація є найдоречнішою: семантичні ознаки з глибоких шарів – на ранньому етапі для формування структури, ознаки границь з початкових шарів – на наступному етапі для уточнення. Це дає змогу декодеру гнучко використовувати різні типи інформації на різних етапах обробки.

4.3 Конфігурації для досліджень

На основі узагальненої архітектури було сформовано дев'ять конкретних конфігурацій для експериментального дослідження.

Конфігурація 1 (рис. 4.1). Базовий декодер з білінійною інтерполяцією. Механізм збільшення роздільної здатності – білінійна інтерполяція з коефіцієнтом 2. Допоміжні з'єднання – відсутні. Координатні згортки: на всіх етапах підвищення роздільної здатності. Блок уточнення – згортка 3×3 → BatchNorm → ReLU. Архітектура енкодера – одна гілка (вихід останнього шару трансформера).

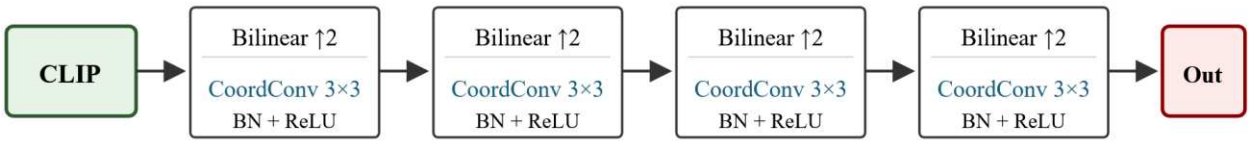

Рисунок 4.1 – Архітектура базового декодера з білінійною інтерполяцією

Конфігурація 2 (рис. 4.2). Декодер з PixelShuffle. Механізм збільшення роздільної здатності – PixelShuffle з коефіцієнтом 2. Допоміжні з'єднання – відсутні. Координатні згортки – відсутні. Блок уточнення – згортка 3×3 → BatchNorm → ReLU. Ініціалізація: стандартна (Kaiming).

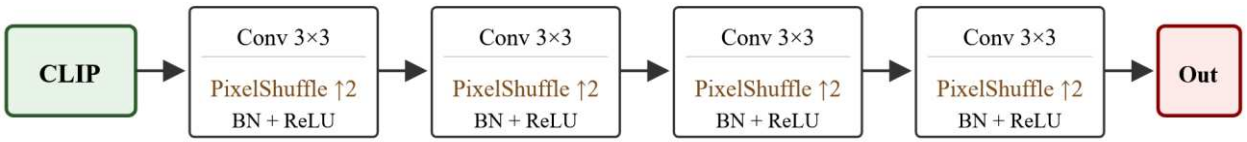

Рисунок 4.2 – Архітектура декодера з PixelShuffle

Конфігурація 3 (рис. 4.3). Декодер з PixelShuffle та уточненням. Механізм збільшення роздільної здатності – PixelShuffle з коефіцієнтом 2 та ICNR-ініціалізацією. Допоміжні з'єднання – відсутні. Координатні згортки – на перших двох етапах підвищення роздільної здатності. Блок уточнення – розширений (CoordConv 3×3 → BatchNorm → ReLU → згортка 3×3 → BatchNorm → ReLU). Ініціалізація – ICNR для усунення артефактів «шахової дошки».

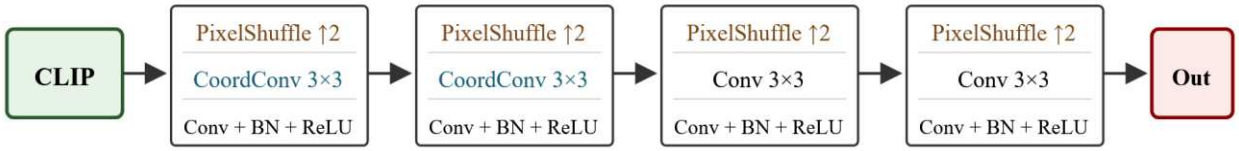

Рисунок 4.3 – Архітектура декодера з PixelShuffle та CoordConv

Конфігурація 4 (рис. 4.4). Гібридний декодер з білінійною інтерполяцією. Механізм збільшення роздільної здатності – білінійна

інтерполяція з коефіцієнтом 2. Допоміжні з'єднання – злиття ознак із трьох шарів трансформера (L3, L7, L9). Координатні згортки – у блоці злиття та на перших двох етапах декодера. Блок злиття – проекція кожної гілки до 128 каналів → конкатенація → CoordConv → згортка 3×3. Блок уточнення – згортка 3×3 → BatchNorm → ReLU → згортка 3×3 → BatchNorm → ReLU

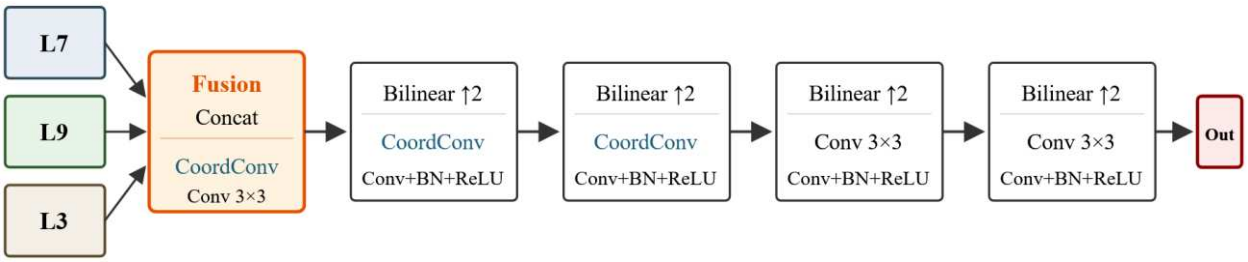

Рисунок 4.4 – Архітектура гібридного декодера з білінійною інтерполяцією

Конфігурація 5. Гібридний декодер з PixelShuffle. Механізм збільшення роздільної здатності – PixelShuffle з коефіцієнтом 2 та ICNR-ініціалізацією. Допоміжні з'єднання – злиття ознак із трьох шарів трансформера (L3, L7, L9). Координатні згортки – у блоці злиття та в режимі уточнення. Блок злиття – аналогічний конфігурації 4. Режими роботи – простий (рис. 4.5) PixelShuffle → BatchNorm, або з уточненням (рис. 4.6) – PixelShuffle → CoordConv → згортка 3×3.

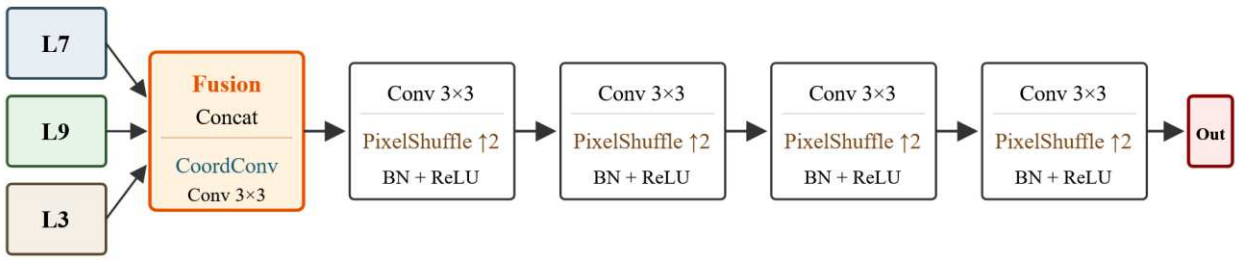

Рисунок 4.5 – Архітектура гібридного декодера з PixelShuffle

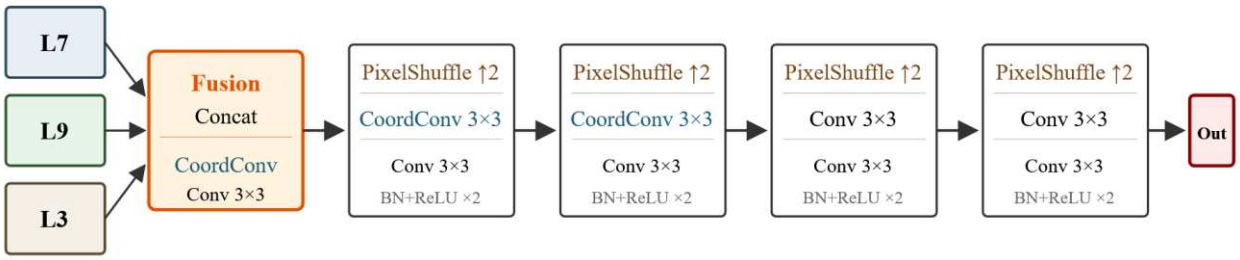

Рисунок 4.6 – Архітектура гібридного декодера з PixelShuffle та CoordConv

Конфігурація 6 (рис. 4.7). UNet-подібний декодер з білінійною інтерполяцією. Механізм збільшення роздільної здатності – білінійна інтерполяція з коефіцієнтом 2. Допоміжні з'єднання – skip-з'єднання з шарів L3 та L7 через проекційні блоки. Координатні згортки – на перших двох етапах декодера. Проекція skip-з'єднань – згортка 1×1 (768 → 128 → 64) → BatchNorm → ReLU. Блок уточнення: CoordConv 3×3 → BatchNorm → ReLU → згортка 3×3 → BatchNorm → ReLU. Інтеграція – конкатенація після підвищення роздільної здатності основної гілки.

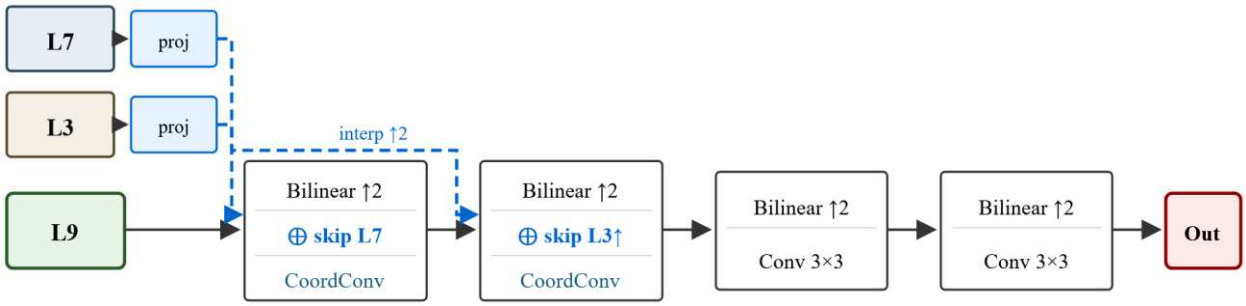

Рисунок 4.7 – Архітектура декодера з ієрархічними допоміжними з'єднаннями та білінійною інтерполяцією

Конфігурація 7. U-Net подібний декодер з PixelShuffle. Механізм збільшення роздільної здатності – PixelShuffle з коефіцієнтом 2 та ICNR-ініціалізацією. Допоміжні з'єднання – skip-з'єднання з шарів L3 та L7 (спільна проекція 64 → 64). Координатні згортки – в режимі уточнення. Проекція skip-з'єднань – згортка 1×1 → BatchNorm → ReLU (спільні ваги для L3 та L7). Інтеграція – конкатенація перед PixelShuffle-блоком із білінійною

інтерполяцією skip-ознак до потрібного розміру. Режими роботи – простий (рис. 4.8) або з уточненням, аналогічно конфігурації 5 (рис. 4.9).

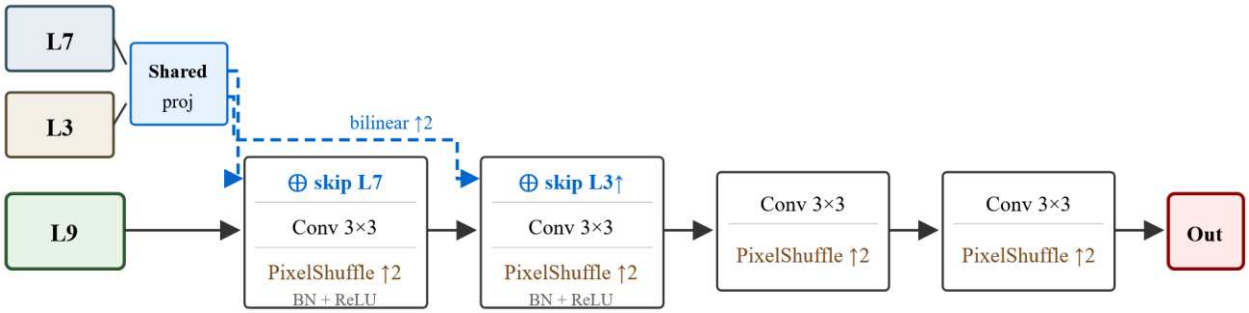

Рисунок 4.8 – Архітектура декодера з ієрархічними допоміжними з'єднаннями та PixelShuffle

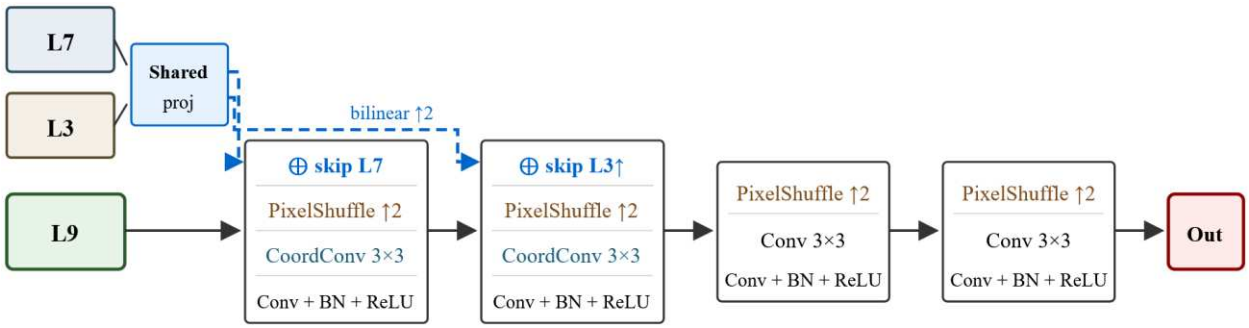

Рисунок 4.9 – Архітектура декодера з ієрархічними допоміжними з'єднаннями, PixelShuffle та CoordConv

4.4 Результати експериментального дослідження

Для порівняння архітектурних конфігурацій декодера відстаней було проведено серію експериментів на наборах даних LVIS, для навчання, та PhraseCut для перевірки якості за mean Dice score і нашою метрикою. Усі конфігурації навчалися протягом трьох епох з ідентичними гіперпараметрами. Для тесту використовувалося одне й те саме зображення з вхідним текстовим запитом – «a photo of a bird». Для постобробки використовувалися два алгоритми. Перший – групування від знайдених центрів об'єктів з врахуванням зв'язності. Другий – hough voting та врахування зв'язності без

використання передбачень центрів об’єктів. Перша архітектура для порівняння була з першої конфігурації. На рисунку 4.10 зліва направо наведено приклад вхідного зображення, знайдених центрів і чотирьох карт відстаней до обмежувальної рамки. На рисунку 4.11 показані результати пошуку об’єктів на кожному з чотирьох каналів карт відстаней. Результати постобробки показані на рисунках 4.12 (з урахуванням центрів) та 4.13 (без врахування центрів). Результати для всіх інших варіантів архітектур показані на рисунках 4.13-4.45.

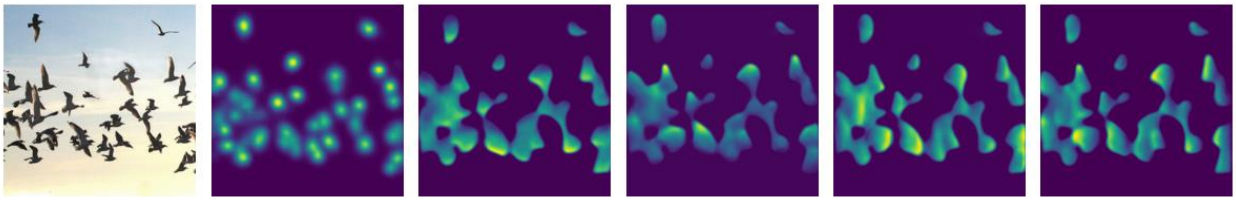

Рисунок 4.10 – Результат роботи мережі для архітектури з білінійною інтерполяцією

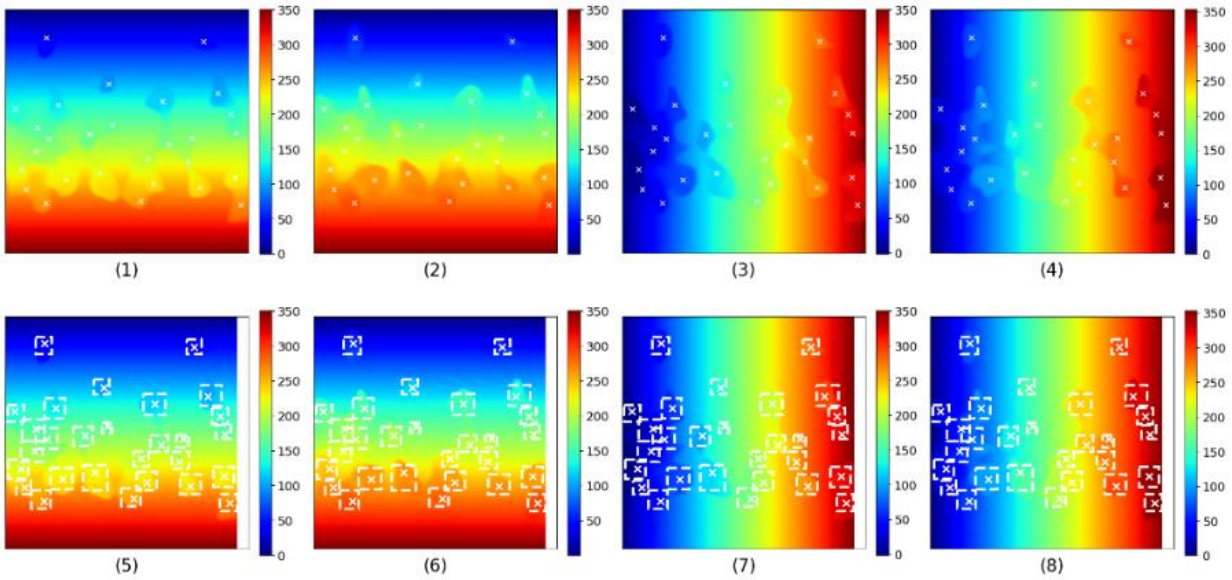

Рисунок 4.11 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для архітектури з білінійною інтерполяцією

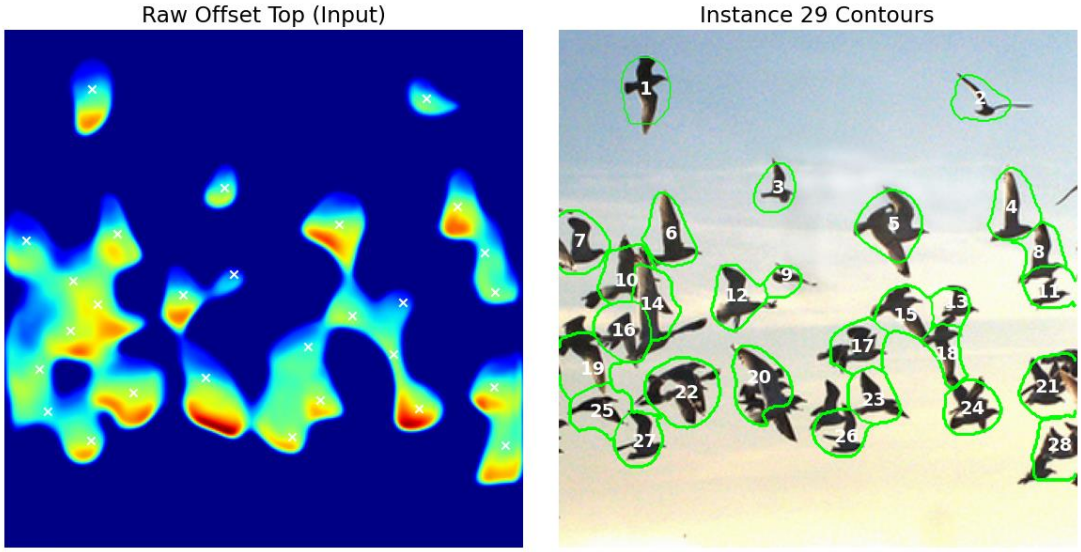

Рисунок 4.12 – Результат постобробки з врахуванням центрів для архітектури з білінійною інтерполяцією

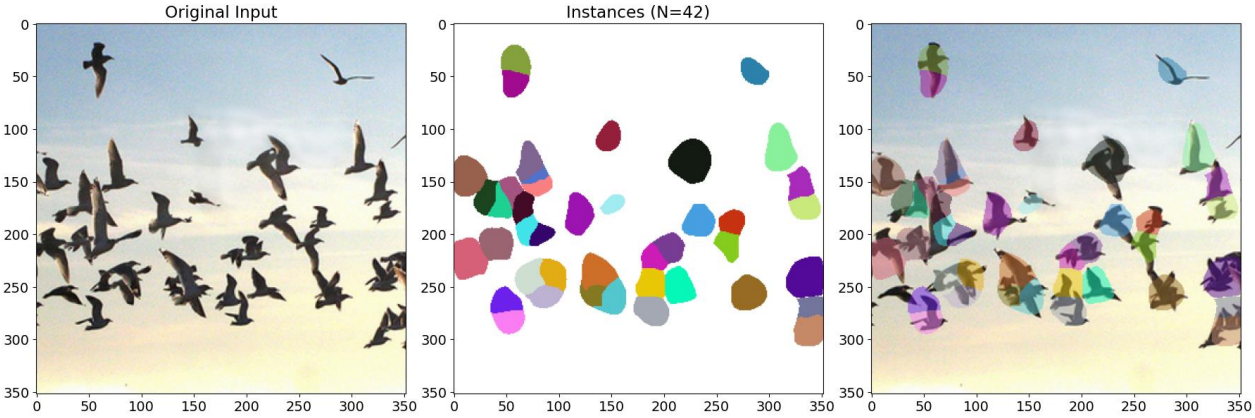

Рисунок 4.13 – Результат постобробки без врахування центрів для архітектури з білінійною інтерполяцією

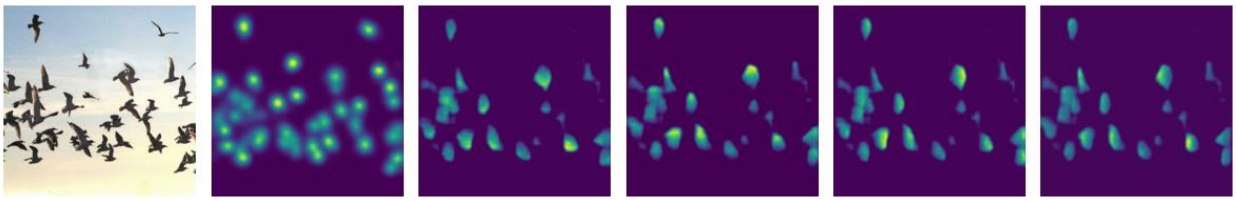

Рисунок 4.14 – Результат роботи мережі для PixelShuffle архітектури

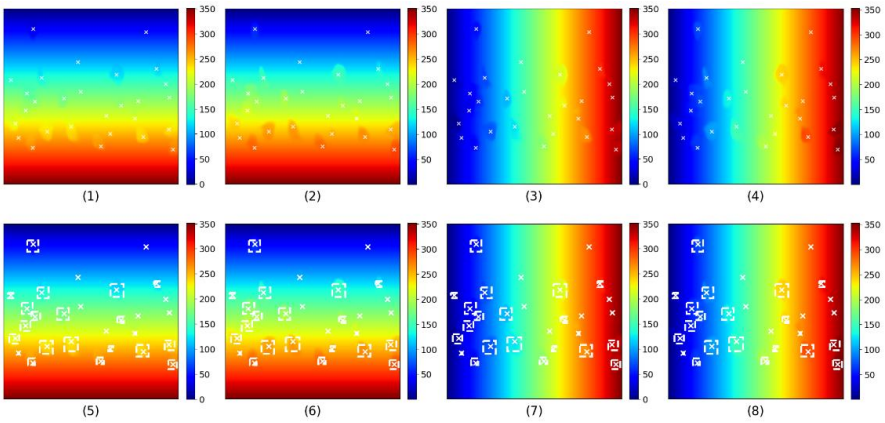

Рисунок 4.15 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для архітектури PixelShuffle

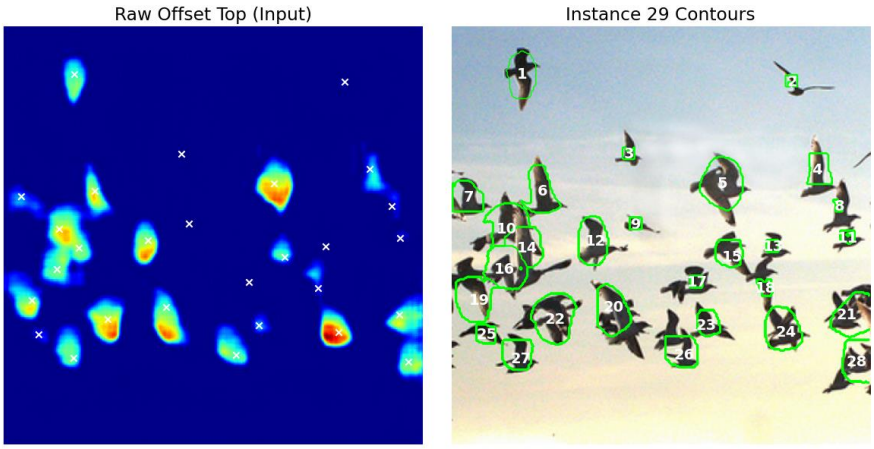

Рисунок 4.16 – Результат постобробки з врахуванням центрів для PixelShuffle архітектури

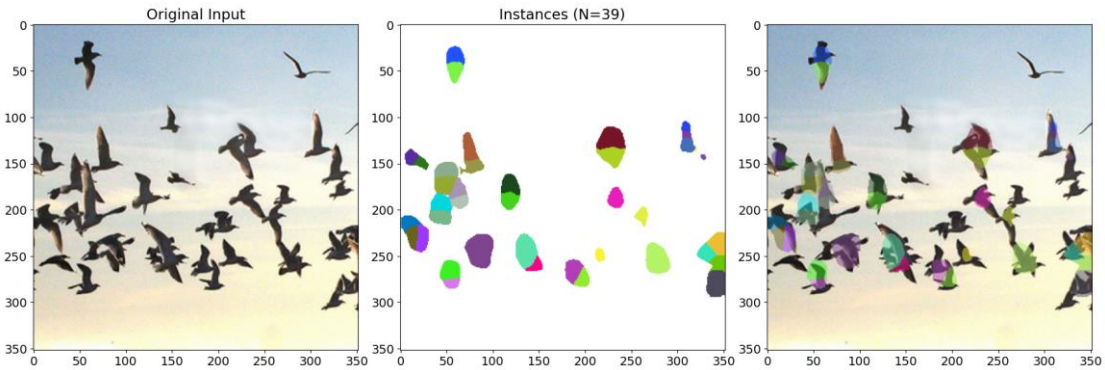

Рисунок 4.17 – Результат постобробки без врахування центрів для PixelShuffle архітектури

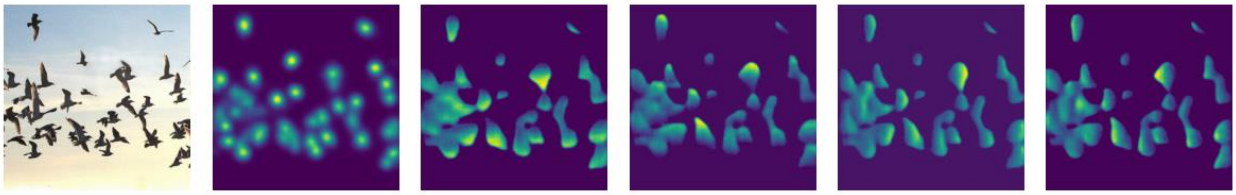

Рисунок 4.18 – Результат роботи мережі для архітектури PixelShuffle з CoordConv

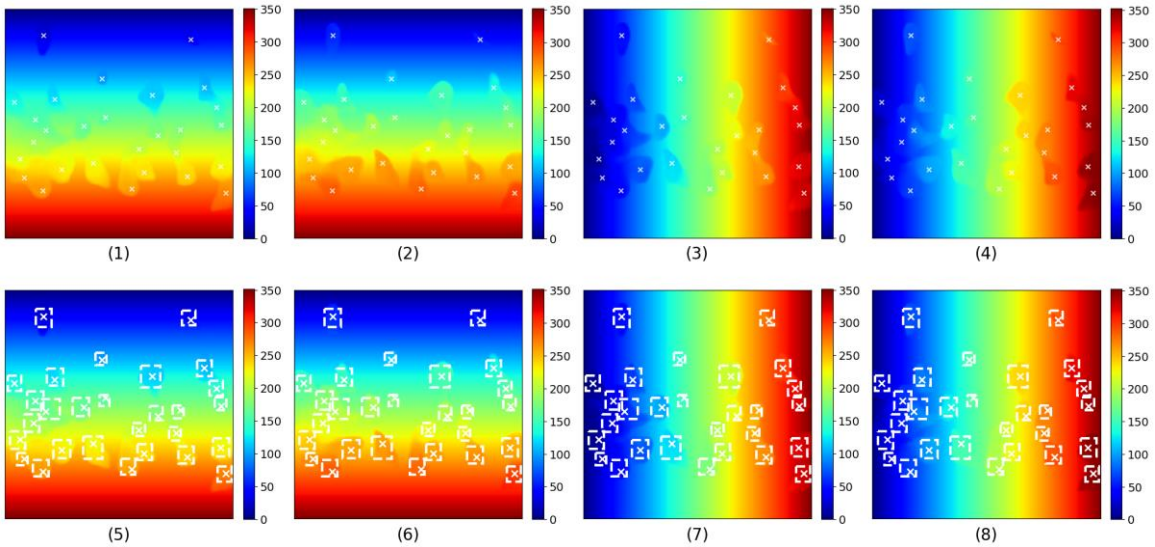

Рисунок 4.19 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для архітектури PixelShuffle з CoordConv

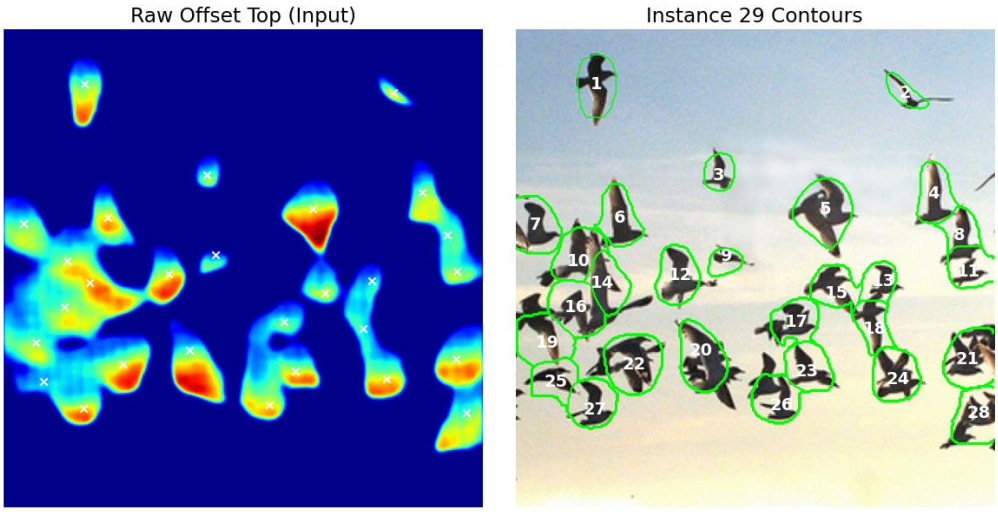

Рисунок 4.20 – Результат постобробки з врахуванням центрів для архітектури PixelShuffle з CoordConv

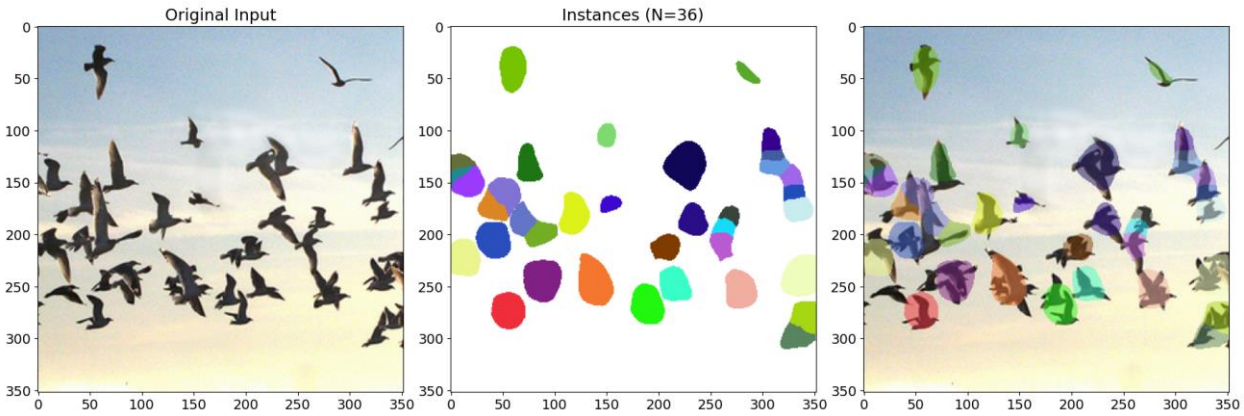

Рисунок 4.21 – Результат постобробки без врахування центрів для архітектури PixelShuffle з CoordConv

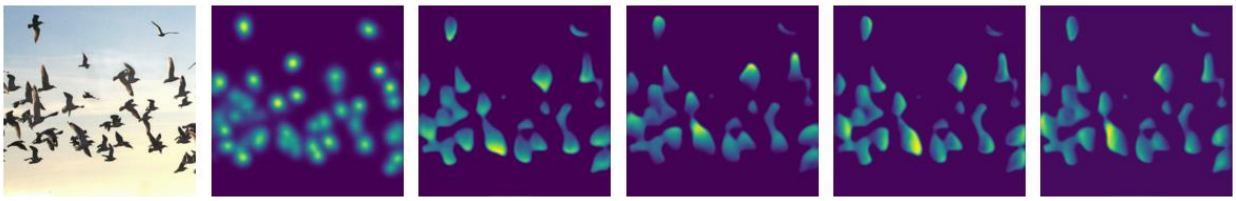

Рисунок 4.22 – Результат роботи мережі для архітектури з гібридним декодером та білінійною інтерполяцією

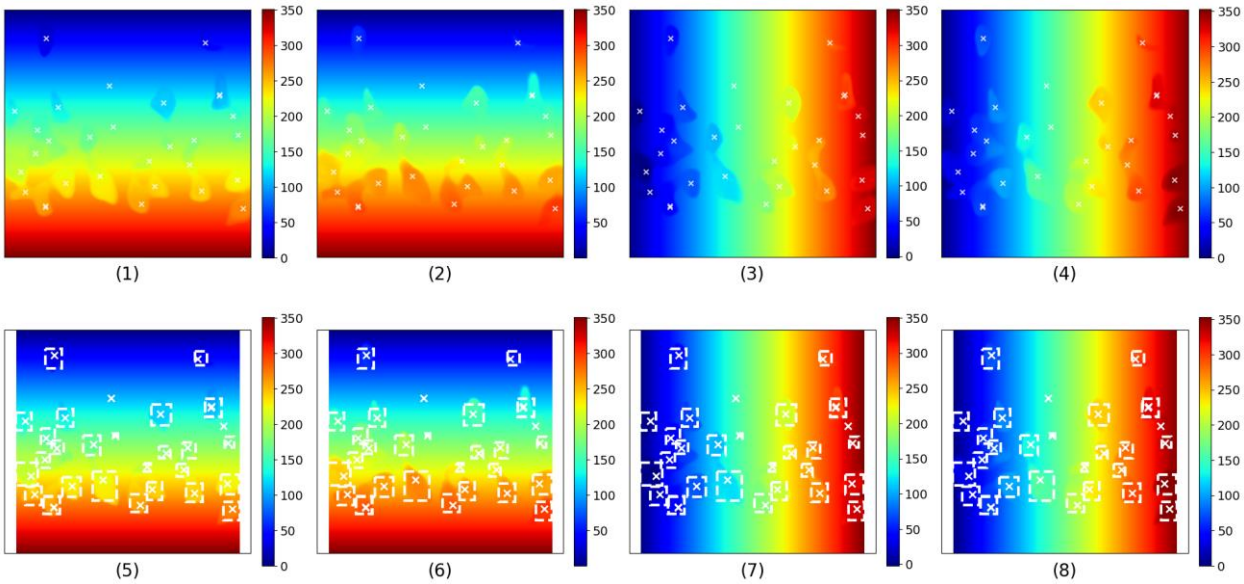

Рисунок 4.23 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для архітектури гібридного декодеру з білінійною інтерполяцією

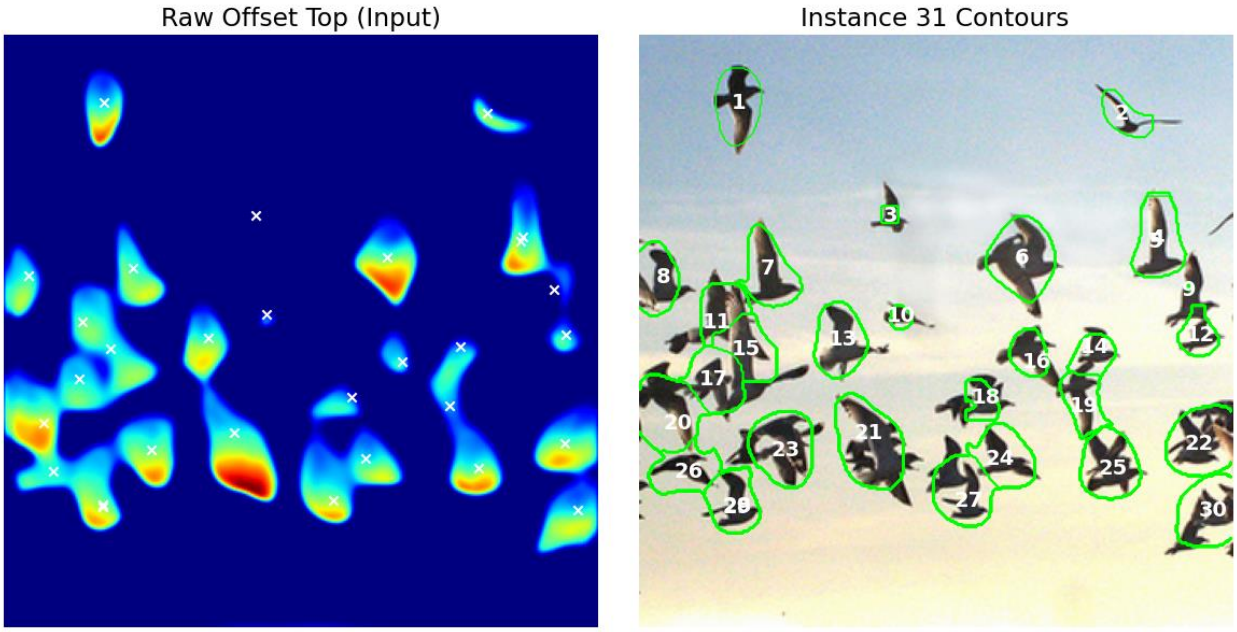

Рисунок 4.24 – Результат постобробки з врахуванням центрів для архітектури гібридного декодеру з білінійною інтерполяцією

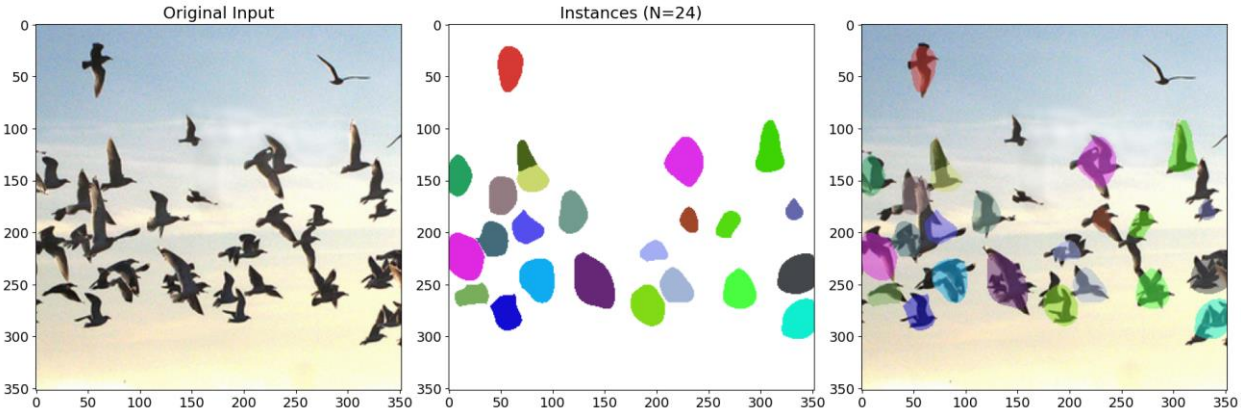

Рисунок 4.25 – Результат постобробки без врахування центрів для архітектури гібридного декодеру з білінійною інтерполяцією

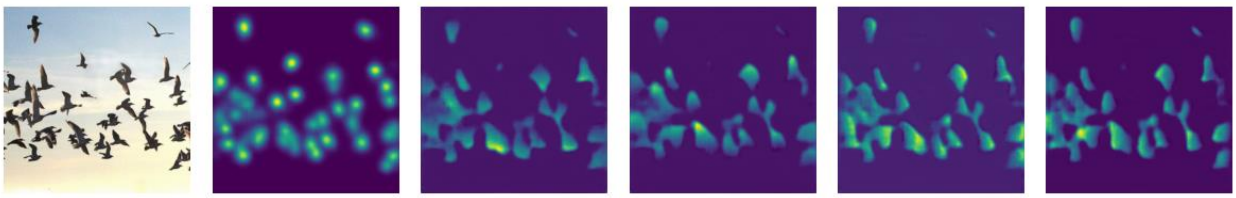

Рисунок 4.26 – Результат роботи мережі для гібридного декодеру з PixelShuffle

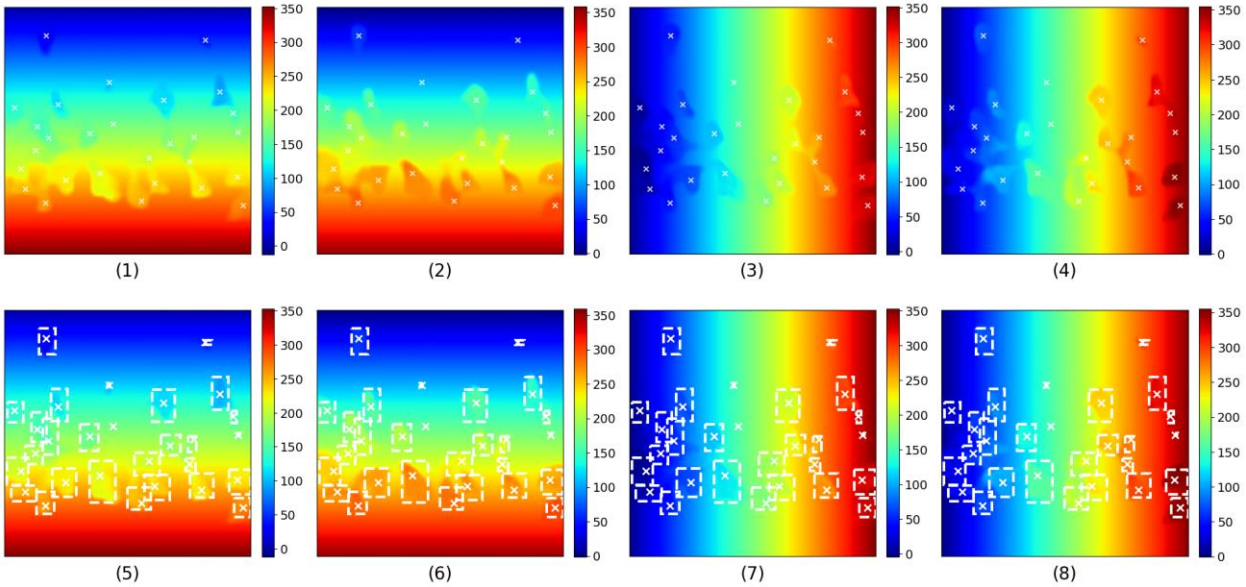

Рисунок 4.27 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для гібридного декодеру з PixelShuffle

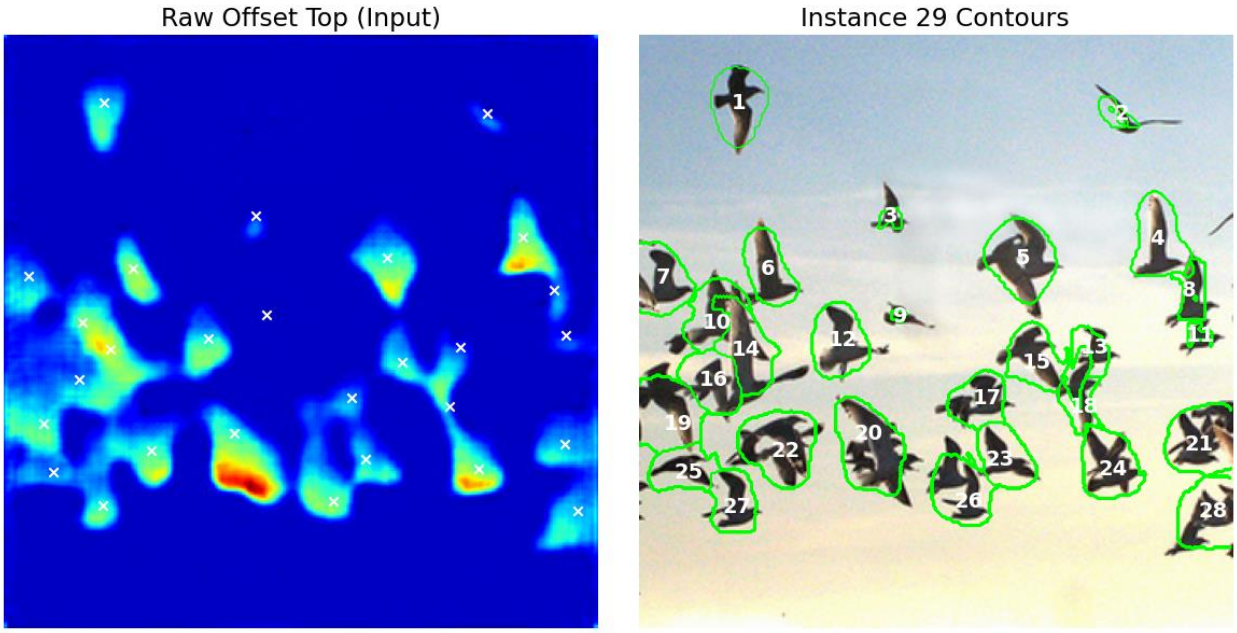

Рисунок 4.28 – Результат постобробки з врахуванням центрів для гібридного декодеру з PixelShuffle

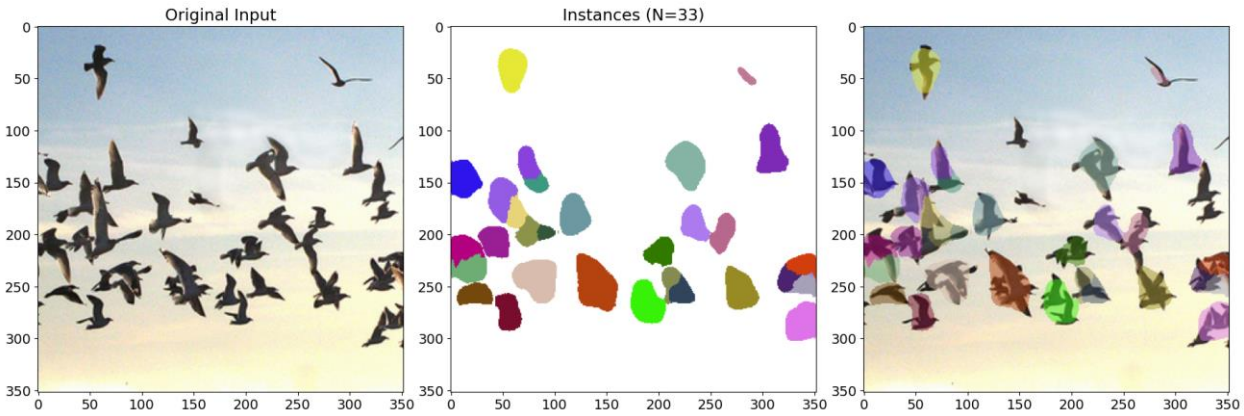

Рисунок 4.29 – Результат постобробки без врахування центрів для гібридного декодеру з PixelShuffle

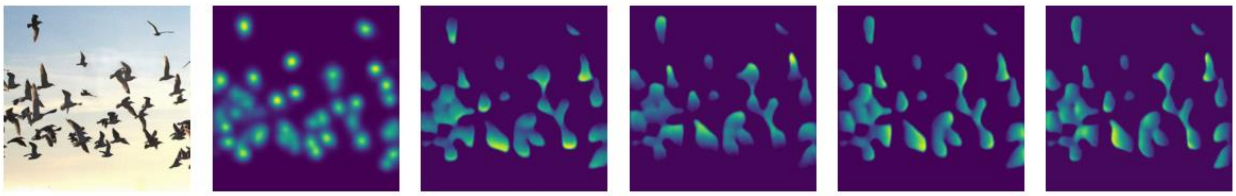

Рисунок 4.30 – Результат роботи мережі для гібридного декодеру з PixelShuffle та CoordConv

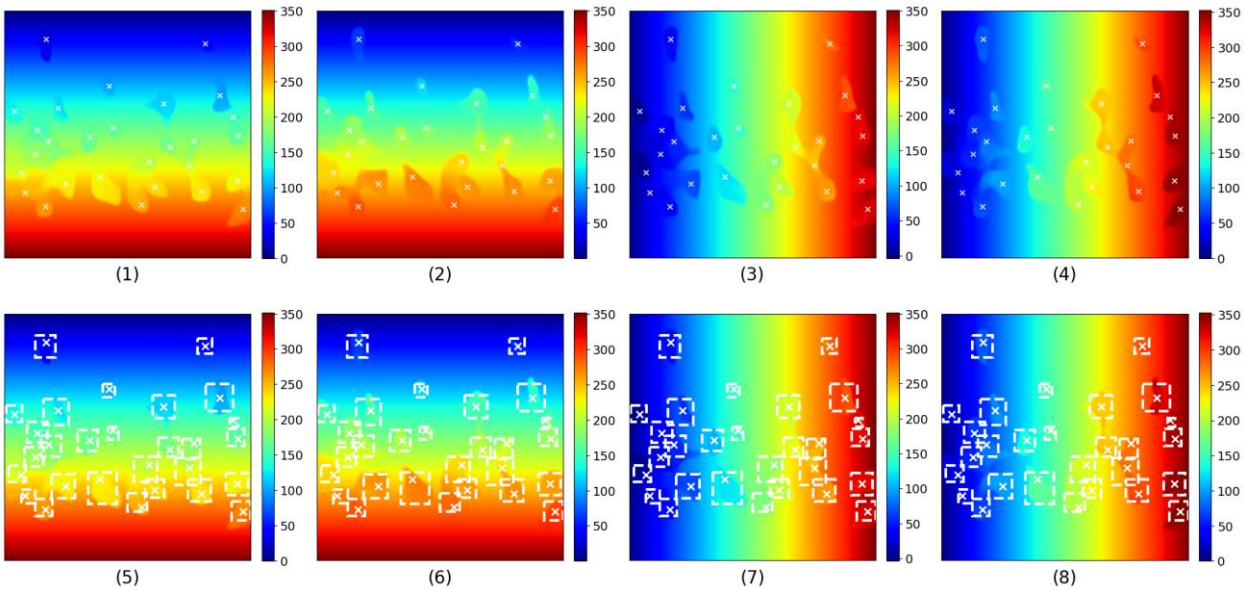

Рисунок 4.31 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для гібридного декодеру з PixelShuffle та CoordConv

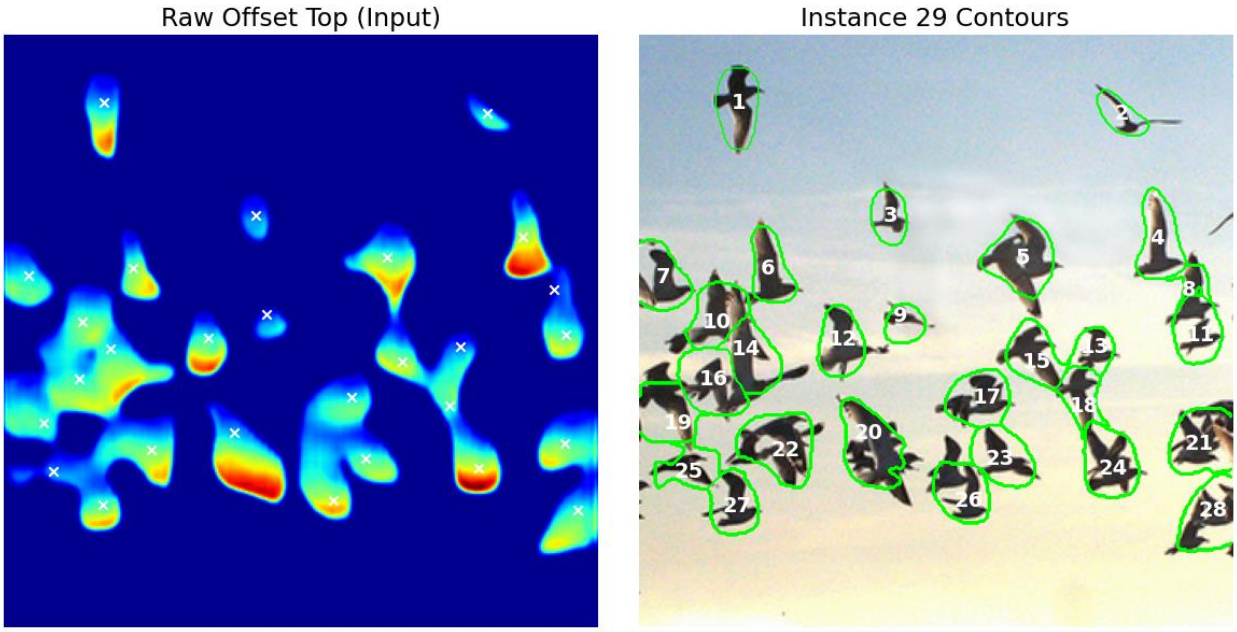

Рисунок 4.32 – Результат постобробки з врахуванням центрів для гібридного декодеру з PixelShuffle та CoordConv

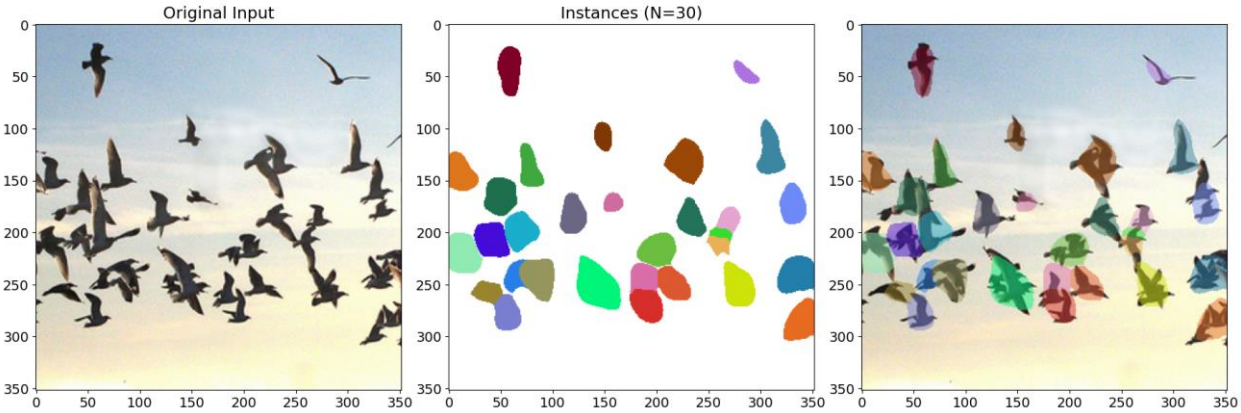

Рисунок 4.33 – Результат постобробки без врахування центрів для гібридного декодеру з PixelShuffle та CoordConv

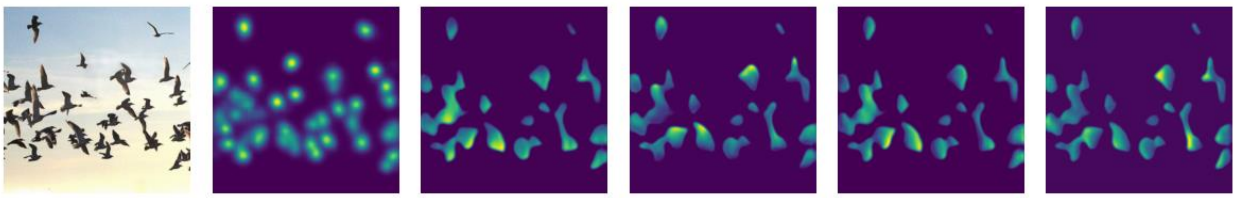

Рисунок 4.34 – Результат роботи мережі для архітектури з ієрархічним декодером та білінійною інтерполяцією

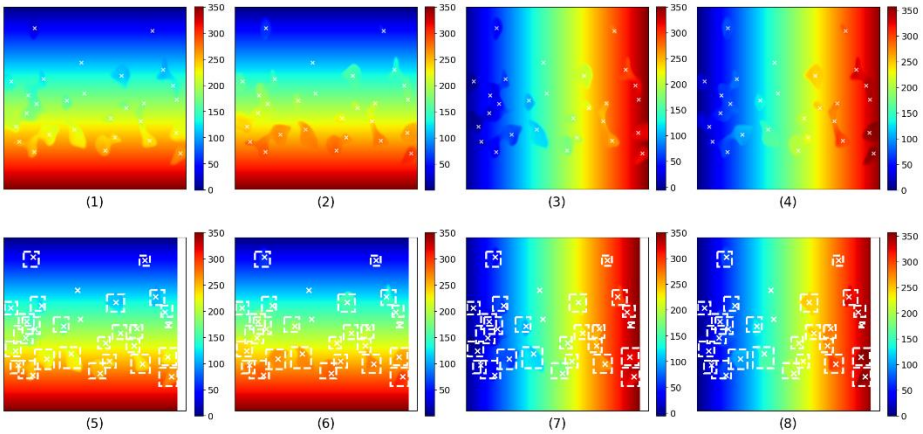

Рисунок 4.35 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для архітектури з ієрархічним декодером та білінійною інтерполяцією

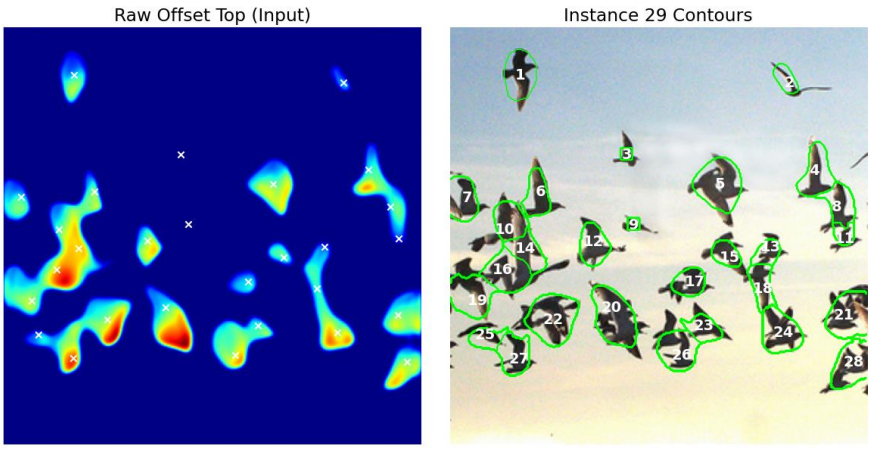

Рисунок 4.36 – Результат постобробки з врахуванням центрів для архітектури з ієрархічним декодером та білінійною інтерполяцією

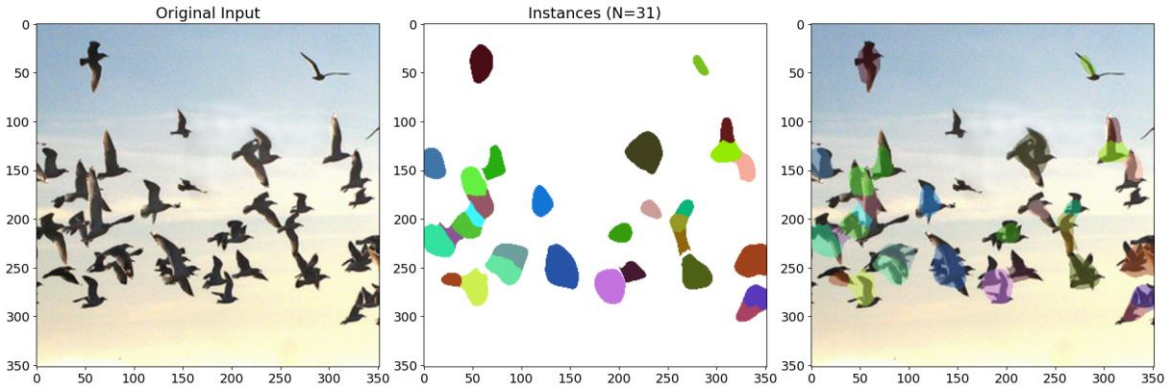

Рисунок 4.37 – Результат постобробки без врахування центрів для архітектури ієрархічного декодеру з білінійною інтерполяцією

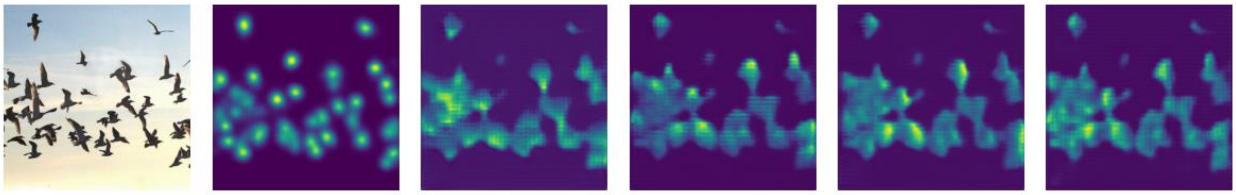

Рисунок 4.38 – Результат роботи мережі для ієрархічного декодеру з PixelShuffle

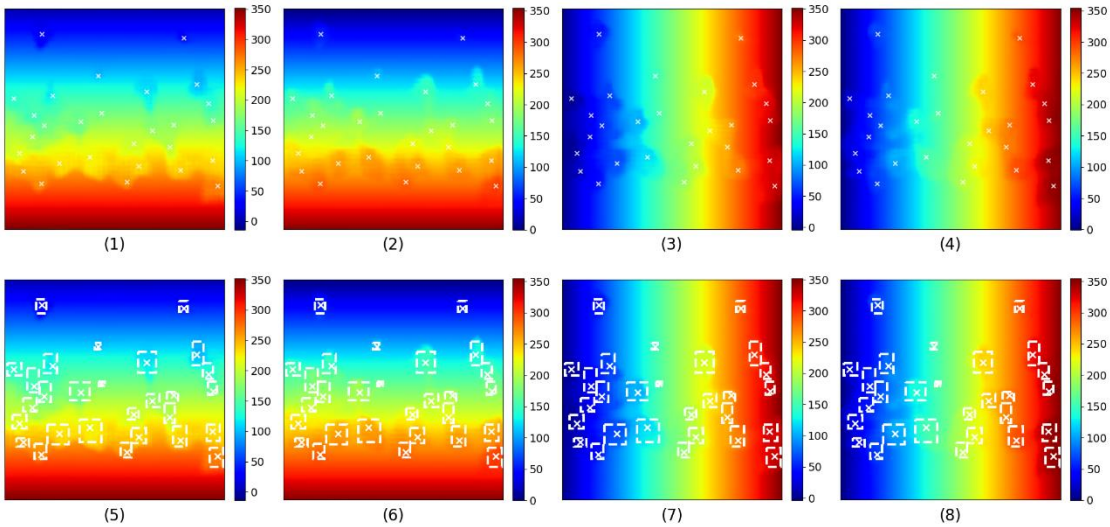

Рисунок 4.39 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для ієрархічного декодеру з PixelShuffle

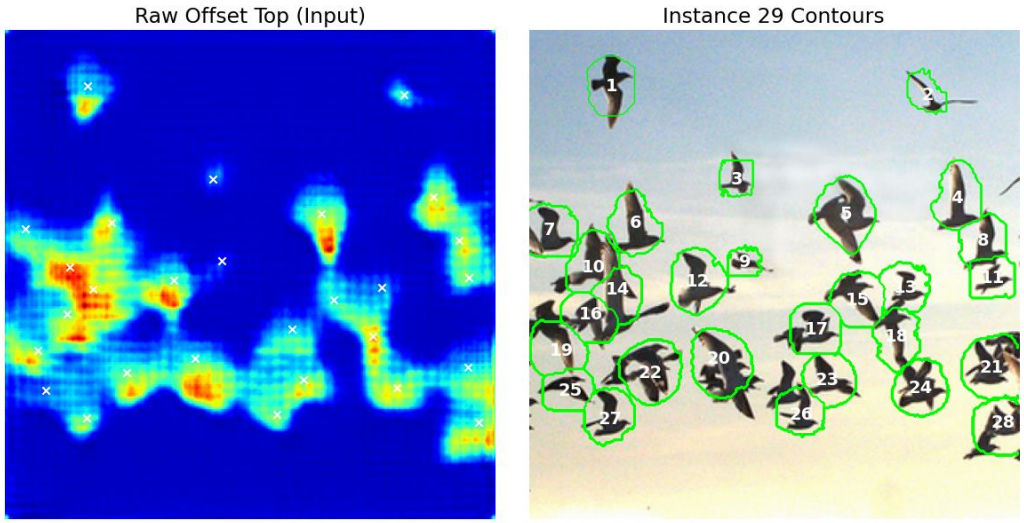

Рисунок 4.40 – Результат постобробки з врахуванням центрів для ієрархічного декодеру з PixelShuffle

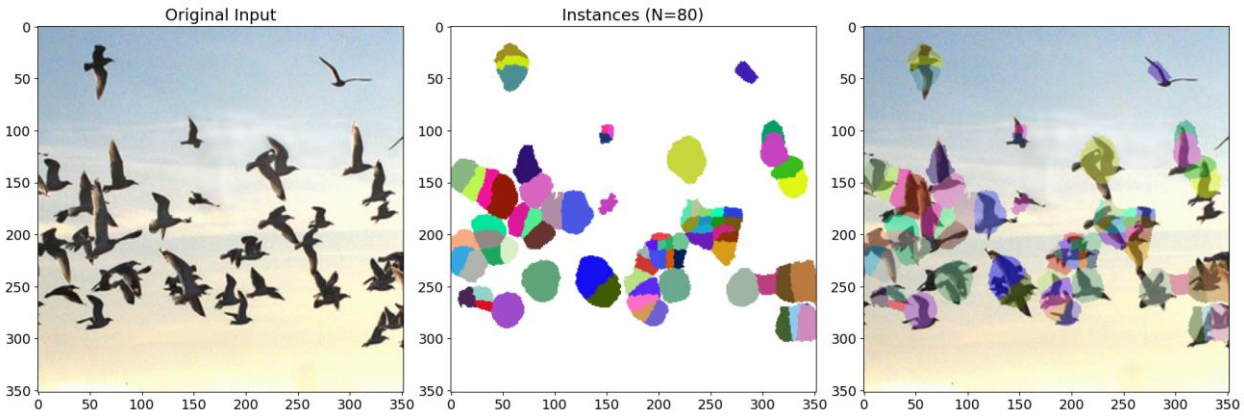

Рисунок 4.41 – Результат постобробки без врахування центрів для ієрархічного декодеру з PixelShuffle

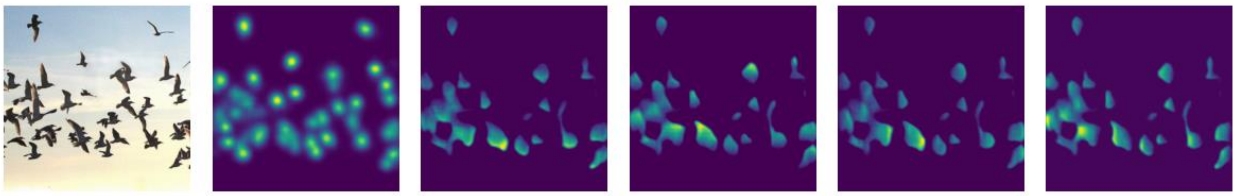

Рисунок 4.42 – Результат роботи мережі для ієрархічного декодеру з PixelShuffle та CoordConv

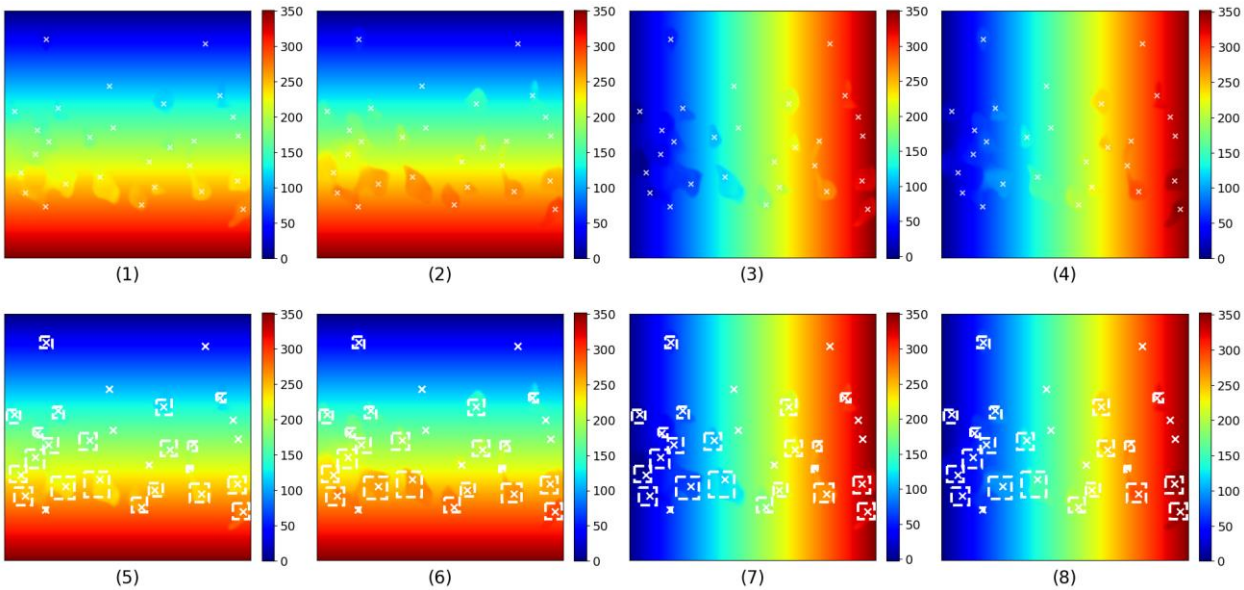

Рисунок 4.43 – Виявлення об’єктів за передбаченими відстанями до їхніх країв (1-4) та обчислення їх обмежувальних рамок (5-8) для ієрархічного декодеру з PixelShuffle та CoordConv

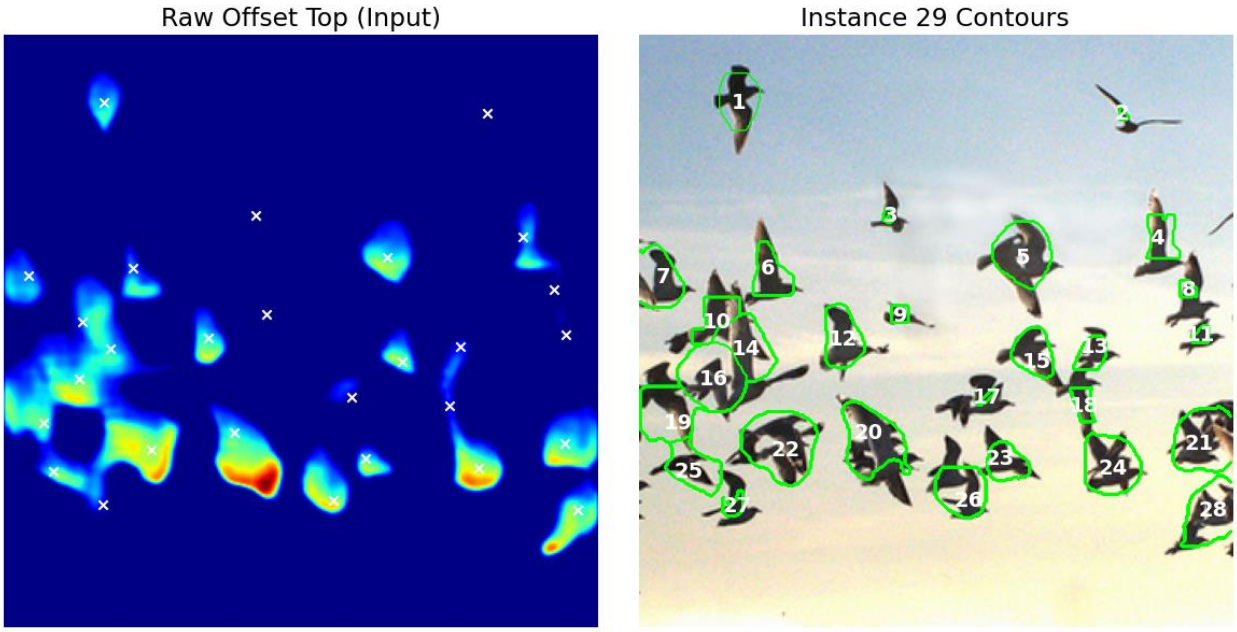

Рисунок 4.44 – Результат постобробки з врахуванням центрів для ієрархічного декодеру з PixelShuffle та CoordConv

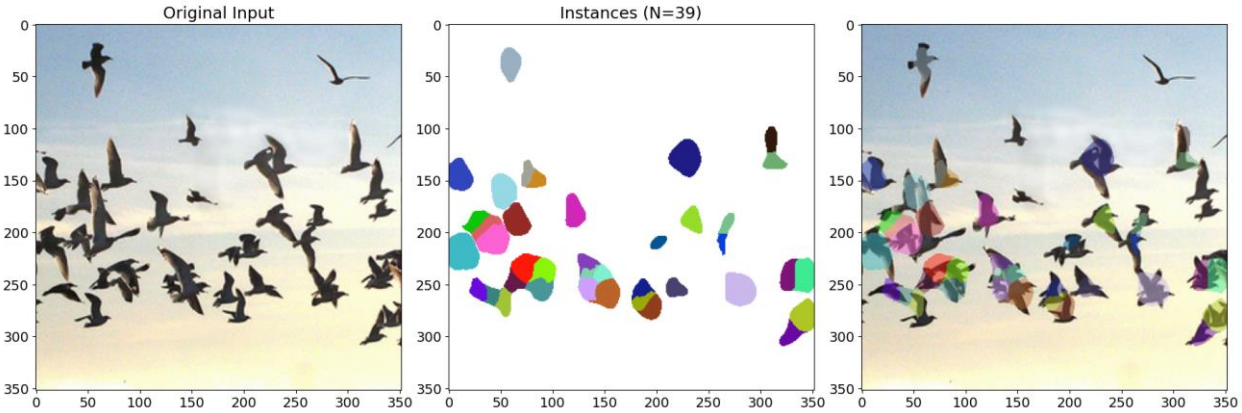

Рисунок 4.45 – Результат постобробки без врахування центрів для ієрархічного декодеру з PixelShuffle та CoordConv

У таблиці 4.1 наведені результати тестування кожної з архітектури. Метрику було додатково нормовано за розмірами зображення та максимумом між кількістю класів отриманих та істинних $K_{\max}$:

$$\hat{\rho}(X,Y) = \frac{\rho(X,Y)}{(H \cdot W)^2 \left(1 - \frac{1}{K_{max}}\right)}. \tag{4.2}$$

Таблиця 4.1 – Порівняння конфігурацій декодерів

Архітектура	**Mean Dice**	**Norm Partition metric**
Bilinear	0,2170	0,2698
PixelShuffle	0,1867	0,2754
PixelShuffle + CoordConv	0,2232	0,2690
Hybrid bilinear	0,2198	0,2747
Hybrid + PixelShuffle	**0,2374**	**0,2672**
Hybrid + PixelShuffle + CoordConv	0,2058	0,2804
UNet bilinear	0,2127	0,2710
UNet + PixelShuffle	0,2257	0,2706
UNet + PixelShuffle + CoordConv	0,2022	0,2802

Аналіз результатів експериментів дозволяє зробити такі висновки щодо ефективності різних архітектурних конфігурацій декодера відстаней.

Найвищий показник mean Dice score (0,2374) досягнуто гібридною архітектурою з механізмом PixelShuffle без координатних згорток. Ця конфігурація також демонструє найкращий результат за запропонованою метрикою розбиттів (0,2672), що свідчить про узгодженість обох метрик оцінювання та підтверджує ефективність багаторівневого злиття ознак із шарів L3, L7 та L9 трансформера.

Порівняння базових механізмів збільшення роздільної здатності показує, що білінійна інтерполяція (mean Dice = 0,2170) перевершує стандартний PixelShuffle (mean Dice = 0,1867) у базовій конфігурації. Проте додавання координатних згорток до PixelShuffle суттєво покращує результати (mean Dice = 0,2232), що підтверджує важливість позиційної інформації для задачі прогнозування відстаней до границь обмежувальної рамки.

Примітним є той факт, що додавання координатних згорток не завжди призводить до покращення якості. У гібридній архітектурі з PixelShuffle

додавання CoordConv погіршує mean Dice з 0,2374 до 0,2058. Аналогічна тенденція спостерігається для U-Net подібної архітектури (з 0,2257 до 0,2022). Це може пояснюватися надлишковістю позиційної інформації при використанні багаторівневого злиття ознак, де просторовий контекст вже достатньо представлений через допоміжні з'єднання з різних шарів енкодера, або те, що за три епохи модель ще не навчилася якісно аналізувати такі дані.

U-Net подібні архітектури з ієрархічними допоміжними з'єднаннями демонструють стабільні, але не найвищі результати. Конфігурація U-Net + PixelShuffle (mean Dice = 0,2257) поступається гібридному підходу, що свідчить про перевагу одноразового злиття ознак на початку декодування порівняно з поетапною інтеграцією просторових з'єднань.

Кореляція між метриками mean Dice та запропонованою метрикою розбиттів є помітною. Конфігурації з вищим mean Dice загалом демонструють нижчі значення метрики розбиттів, що відповідає кращій якості. Це підтверджує коректність запропонованої метрики як альтернативного інструменту оцінювання якості сегментації екземплярів.

Порівняння вищезгаданих методів для open-vocabulary сегментації наведено у таблиці 4.2. Вони проводилися на GTX 3090 Ti.

Таблиця 4.2 – Порівняння існуючих моделей та отриманої

Model	**Parameters**	**Fps**	**Mean Dice**
HQ-SAM ViT-B	358M	9,07	0,465
SAM ViT-B	362,1M	9,86	0,491
Grounded SAM	358M + 232.3M	3,71	–
FastSAM without text	72M	25	–
FastSAM text	72M + 151M	0,91	0,212
InstanceCLIPSeg	від 152M до 153,67M	17	0,237

В обраних моделях текстовий пошук здійснюється з використанням сторонніх детекторів. Було використано GroundingDINO для SAM та HQ-SAM, як рекомендовано у відповідних репозиторіях. Для FastSAM детектором є YOLOv8, проте для текстового пошуку застосовується модель CLIP ViT-B/32, що суттєво збільшує кількість параметрів та знижує швидкість виконання. Запропонована модель випереджає всі інші порівнювані моделі за швидкістю пошуку за текстовим описом. За якістю вона зіставна з FastSAM, а також має переваги одноетапного підходу. Середнє значення FPS для InstanceCLIPSeg становить 28 FPS без урахування постобробки та 17 FPS з постобробкою. Воно коливається в межах 1-2 кадру, в залежності від конфігурації і це може бути врахована як похибка.

4.5 Висновки за розділом

Проведено експериментальне дослідження архітектурних конфігурацій декодера відстаней моделі InstanceCLIPSeg. За результатами досліджень сформульовано висновки:

1) Досліджено вплив механізмів відновлення просторової роздільної здатності на якість передбачень. Встановлено, що механізм PixelShuffle з ICNR-ініціалізацією у поєднанні з багаторівневим злиттям ознак забезпечує найкращі результати (mean Dice = 0,2374), перевершуючи базову білінійну інтерполяцію на 9.4%.

2) Проаналізовано роль координатних згорток у архітектурі декодера. Показано, що додавання позиційної інформації є ефективним для базових конфігурацій без допоміжних з’єднань, підвищуючи mean Dice з 0,1867 до 0,2232 для PixelShuffle. Проте при використанні багаторівневого злиття ознак координатні згортки можуть призводити до погіршення результатів через надлишковість позиційної інформації.

3) Порівняно стратегії багаторівневого злиття ознак – одноразове злиття (гібридний підхід) та ієрархічні допоміжні з’єднання (U-Net подібний

підхід). Експериментально підтверджено перевагу гібридного підходу з інтеграцією ознак із шарів L3, L7 та L9 трансформера на початку декодування.

4) Проведено валідацію запропонованої зваженої метрики для порівняння розбиттів. Продемонстровано кореляцію між метрикою розбиттів та стандартною метрикою mean Dice, що підтверджує придатність запропонованого функціоналу для оцінювання якості сегментації екземплярів.

5) Порівняно моделі та визначено оптимальну конфігурацію декодера відстаней для моделі InstanceCLIPSeg – гібридна архітектура з механізмом PixelShuffle, ICNR-ініціалізацією та злиттям ознак із трьох рівнів енкодера без використання координатних згорток у блоках уточнення.

ВИСНОВКИ

Дисертаційне дослідження містить нові наукові результати, які є наслідком вирішення актуального наукового завдання розробки моделей, методів та алгоритмів мультимодальної просторово-семантичної сегментації за нечітким запитом на основі модифікації архітектур глибокого навчання для підвищення ефективності виділення екземплярів об’єктів в умовах open-vocabulary парадигми.

Під час дослідження отримано наведені нижче наукові та практичні результати.

1) Виконаний аналіз проблем та підходів до вирішення задачі сегментації зображень за текстовим запитом в умовах open-set парадигми показав, що існуючі методи сегментації екземплярів реалізовані переважно як двоетапні процедури, де спочатку виконується детекція об’єктів за допомогою спеціалізованого детектора, а потім їх сегментація окремою моделлю. Таке архітектурне рішення обмежує можливості задоволення вимог до швидкості обробки, можливості сегментації за довільним текстовим запитом без перенавчання, а також можливості виділення окремих екземплярів об’єктів в умовах відкритого словника категорій. Проаналізовано еволюцію архітектур нейронних мереж для сегментації – від простих повнозгорткових моделей до сучасних архітектур на основі трансформерів, що демонструють тенденцію до інтеграції механізмів уваги та уніфікації підходів до різних типів сегментації. Класифіковано підходи до сегментації екземплярів за критеріями кількості етапів обробки (одноетапні, двоетапні) та напрямку обробки (top-down, bottom-up). Досліджено особливості open-set та open-vocabulary підходів, включаючи zero-shot learning та few-shot learning. Встановлено, що на цей момент не існує одноетапної open-set моделі для сегментації екземплярів за текстовим запитом, а широко використовувані функції втрат (Dice, IoU, Tversky) не мають строгого математичного обґрунтування як метрики у класичному розумінні теорії метричних просторів та не задовольняють

нерівності трикутника. Таким чином, існуючі підходи до сегментації екземплярів потребують подальшого розвитку з урахуванням потреб одноетапної обробки за нечітким запитом, метричного обґрунтування функцій втрат та ефективної постобробки результатів висхідної сегментації, що свідчить про актуальність розробки методу, моделей та алгоритмів мультимодальної просторово-семантичної сегментації на основі модифікації архітектур глибокого навчання.

2) Вперше запропоновано модель InstanceCLIPSeg для одноетапної сегментації екземплярів за текстовим запитом, яка містить чотири основні компоненти: енкодер зображення на основі Vision Transformer (ViT-B/16), адаптований до роздільної здатності 352×352, текстовий енкодер CLIP, декодер центрів (centers head) для прогнозування теплової карти центрів мас об’єктів та декодер відстаней (offset/distance head) для прогнозування попіксельних відстаней до чотирьох границь обмежувальної рамки. На відміну від двоетапних методів (Grounded-SAM, text FastSAM), запропонована модель виконує виявлення та сегментацію екземплярів за один прохід мережі. Модель використовує механізм умовної модуляції FiLM.

3) Удосконалено метод постобробки результатів висхідної сегментації, який, на відміну від відомих підходів на основі кластеризації, використовує аналіз карт відстаней та адаптивне розширення регіонів від центрів об’єктів з урахуванням просторової зв’язності пікселів та допустимих відхилень.

4) Вперше запропоновано зважену метрику для порівняння сегментацій зображень на основі теорії розбиттів. На відміну від відомих функцій втрат (Dice, Tversky, IoU, Lovász-Softmax), запропонована метрика визначена на просторі розбиттів та задовольняє усі аксіоми метричного простору. Функціонал одночасно містить міри подібності та відмінності класів еквівалентності з урахуванням не лише форми регіонів та їх взаємного просторового розташування, але й різних візуальних властивостей через вагову функцію. Обчислення метрики зводиться до матричних операцій, що

забезпечує просту обчислюваність порівняно з EMD та MiCROM, де точність досягається через розв'язання задач лінійного програмування. Математично доведено, що запропонований функціонал задовольняє всі аксіоми метрики: невід'ємність, рефлексивність, симетрію та нерівність трикутника. Застосування нерівності трикутника дозволяє організувати ієрархічний пошук у просторі розбиттів, а використання зворотної нерівності – ефективно відсіювати несхожі елементи під час порівняння сегментацій.

5) Розроблено підхід до застосування запропонованої метрики як функції втрат для навчання нейронних мереж. Для подолання проблеми недиференційовності операції виділення сегментів запропоновано використання кластеризації KNN з перевіркою зв'язності елементів та апроксимації Straight-Through Estimator (STE), що дозволяє передавати градієнти назад через недиференційовні операції. Експериментально досліджено різні варіанти STE апроксимації: згортку 1×1, Gumbel-Softmax та soft masks, де останні описують ймовірнісне значення приналежності пікселя до жорстких центроїдів. Проведено експериментальне дослідження на синтетичному наборі даних із п'яти фігур (коло, прямокутник, трикутник, п'ятикутник, зірка) з використанням архітектури SegNet, що розв'язує задачу попіксельної регресії на зображеннях 100×100 пікселів. Результати показали, що метрика дозволяє знаходити об'єкти та підкреслювати розділення областей з урахуванням їх зв'язності, що робить її перспективним кандидатом для комбінованих функцій втрат, де вона доповнюватиме функції, що фокусуються на семантиці або приналежності до класу.

6) Набув подальшого розвитку підхід до відновлення роздільної здатності в декодерах сегментаційних мереж. Досліджено та порівняно три механізми збільшення просторової роздільної здатності: білінійну інтерполяцію, транспоновану згортку та PixelShuffle, – у поєднанні зі стратегіями багаторівневого злиття ознак та координатними згортками. Для механізму PixelShuffle застосовано метод ICNR-ініціалізації, що забезпечує однакові початкові значення для всіх підпікселів, усуваючи блочні артефакти

на початку навчання. Порівняно дві стратегії багаторівневого злиття ознак: одноразове злиття (гібридний підхід з інтеграцією активацій із шарів L3, L7 та L9 трансформера на початку декодування) та ієрархічні допоміжні з'єднання (U-Net подібний підхід з поетапною інтеграцією просторових з'єднань). Для дев'яти конфігурацій декодера відстаней проведено систематичне експериментальне дослідження на наборах даних LVIS та PhraseCut. Встановлено, що використання PixelShuffle з ICNR-ініціалізацією у гібридній архітектурі зі злиттям ознак із трьох рівнів трансформера забезпечує найкращу якість передбачень (mean Dice 0,2374), перевершуючи базову білінійну інтерполяцію (mean Dice 0,2170) на 9.4%, та дозволяє уникнути артефактів типу «шахової дошки», характерних для стандартної транспонованої згортки. Показано, що додавання координатних згорток є ефективним для базових конфігурацій без допоміжних з'єднань, підвищуючи mean Dice з 0,1867 до 0,2232 для PixelShuffle, проте при використанні багаторівневого злиття ознак координатні згортки можуть призводити до погіршення результатів (з 0,2374 до 0,2058 для гібридної архітектури).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Experimental study of distance map decoder architectural configurations for instance segmentation by text query. System technologies, 2(163), 3–12. https://doi.org/10.34185/1562-9945-2-163-2026-01

2. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Metrical loss functions for image segmentation based on convolutional neural networks. Applied Aspects of Information Technology, 9(2), 172–184. https://doi.org/10.15276/aait.09.2026.12

3. Kovtunenko, A. R., & Mashtalir, S. V. (2025). Improved segmentation model to identify object instances based on textual prompts. Herald of Advanced Information Technology, 8(1), 54–66. https://doi.org/10.15276/hait.08.2025.4

4. Kovtunenko, A. R., & Mashtalir, S. V. (2025). A Review of Modern Neural Network Architectures for Image Segmentation. Management Information System and Devises, (185), 53–62. https://doi.org/10.30837/0135-1710.2025.185.043

5. Ковтуненко, А. Р. (2025). Мультимодальна висхідна сегментація об’єктів за текстовим запитом. У III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (с. 11–12). Західноукраїнський національний університет

6. Ковтуненко, А., & Машталір, В. (2024). Аналіз та порівняння функцій втрат для вирішення задачі сегментації. У Радіоелектроніка та молодь у XXI столітті. Т. 7: Конференція «Комп’ютерний зір, системний аналіз та математичне моделювання». Press of the Kharkiv National University of Radioelectronics. https://doi.org/10.30837/iyf.cvsamm.2024.057

7. Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Learning to prompt for vision-language models. International Journal of Computer Vision. https://doi.org/10.1007/s11263-022-01653-1

8. Buettner, K., & Kovashka, A. (2024). Investigating the role of attribute context in vision-language models for object recognition and detection. У 2024 IEEE/CVF winter conference on applications of computer vision (WACV). IEEE. https://doi.org/10.1109/wacv57701.2024.00539

9. Zhu, C., & Chen, L. (2024). A survey on open-vocabulary detection and segmentation: Past, present, and future. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–20. https://doi.org/10.1109/tpami.2024.3413013

10. Xu, J., Hou, J., Zhang, Y., Feng, R., Wang, Y., Qiao, Y., & Xie, W. (2023). Learning open-vocabulary semantic segmentation models from natural language supervision. У 2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52729.2023.00287

11. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PmLR

12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. У 2023 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv51070.2023.00371

13. Xu, P., Zhu, X., & Clifton, D. A. (2023). Multimodal learning with transformers: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–20. https://doi.org/10.1109/tpami.2023.3275156

14. Liang, P. P., Zadeh, A., & Morency, L.-P. (2024). Foundations & trends in multimodal machine learning: Principles, challenges, and open questions. ACM Computing Surveys. https://doi.org/10.1145/3656580

15. Zhang, J., Huang, J., Jin, S., & Lu, S. (2024). Vision-Language models for vision tasks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–20. https://doi.org/10.1109/tpami.2024.3369699

16. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., ... & Zhang, L. (2024). Grounded sam: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159

17. Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. У 2016 fourth international conference on 3D vision (3DV). IEEE. https://doi.org/10.1109/3dv.2016.79

18. Rahman, M. A., & Wang, Y. (2016). Optimizing intersection-over-union in deep neural networks for image segmentation. У Advances in visual computing (с. 234–244). Springer International Publishing. https://doi.org/10.1007/978-3-319-50835-1_22

19. Luddecke, T., & Ecker, A. (2022). Image segmentation using text and image prompts. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.00695

20. Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., ... & Wang, J. (2023). Fast segment anything. arXiv preprint arXiv:2306.12156

21. Salehi, S. S. M., Erdogmus, D., & Gholipour, A. (2017). Tversky loss function for image segmentation using 3D fully convolutional deep networks. У Machine learning in medical imaging (с. 379–387). Springer International Publishing. https://doi.org/10.1007/978-3-319-67389-9_44

22. Berman, M., Triki, A. R., & Blaschko, M. B. (2018). The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. У 2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2018.00464

23. Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432.

24. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). Real-Time single image and video super-

resolution using an efficient sub-pixel convolutional neural network. У 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2016.207

25. Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1. https://doi.org/10.1109/tpami.2021.3059968

26. Emek Soylu, B., Guzel, M. S., Bostanci, G. E., Ekinci, F., Asuroglu, T., & Acici, K. (2023). Deep-Learning-Based approaches for semantic segmentation of natural scene images: A review. Electronics, 12(12), 2730. https://doi.org/10.3390/electronics12122730

27. Sharma, R., Saqib, M., Lin, C. T., & Blumenstein, M. (2022). A survey on object instance segmentation. SN Computer Science, 3(6). https://doi.org/10.1007/s42979-022-01407-3

28. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J. M., Luo, P., & Lu, T. (2022). Panoptic segformer: Delving deeper into panoptic segmentation with transformers. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.00134

29. Elharrouss, O., Al-Maadeed, S., Subramanian, N., Ottakath, N., Almaadeed, N., & Himeur, Y. (2021). Panoptic segmentation: A review. arXiv preprint arXiv:2111.10250

30. Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time instance segmentation. У 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2019.00925

31. Wang, X., Kong, T., Shen, C., Jiang, Y., & Li, L. (2020). SOLO: Segmenting objects by locations. У Computer vision – ECCV 2020 (с. 649–665). Springer International Publishing. https://doi.org/10.1007/978-3-030-58523-5_38

32. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., & Luo, P. (2020). PolarMask: Single shot instance segmentation with polar

representation. У 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.01221

33. Zhang, R., Tian, Z., Shen, C., You, M., & Yan, Y. (2020). Mask encoding for single shot instance segmentation. У 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.01024

34. Lee, Y., & Park, J. (2020). CenterMask: Real-Time Anchor-Free Instance Segmentation. У 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.01392

35. He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 386–397. https://doi.org/10.1109/tpami.2018.2844175

36. Zhang, G., Lu, X., Tan, J., Li, J., Zhang, Z., Li, Q., & Hu, X. (2021). RefineMask: Towards high-quality instance segmentation with fine-grained features. У 2021 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr46437.2021.00679

37. Ke, L., Danelljan, M., Li, X., Tai, Y.-W., Tang, C.-K., & Yu, F. (2022). Mask transfiner for high-quality instance segmentation. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.00437

38. Chen, X., Girshick, R., He, K., & Dollar, P. (2019). TensorMask: A foundation for dense object segmentation. У 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2019.00215

39. Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R., & Urtasun, R. (2020). PolyTransform: Deep polygon transformer for instance segmentation. У 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.00915

40. Ke, L., Tai, Y.-W., & Tang, C.-K. (2021). Deep Occlusion-Aware Instance Segmentation with Overlapping BiLayers. У 2021 IEEE/CVF conference

on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr46437.2021.00401

41. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., & Tao, D. (2022). A survey on vision transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1. https://doi.org/10.1109/tpami.2022.3152247

42. Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. ACM Computing Surveys. https://doi.org/10.1145/3505244

43. Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R. R., Cheng, M.-M., & Hu, S.-M. (2022). Attention mechanisms in computer vision: A survey. Computational Visual Media. https://doi.org/10.1007/s41095-022-0271-y

44. Shelhamer, E., Long, J., & Darrell, T. (2017). Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 640–651. https://doi.org/10.1109/tpami.2016.2572683

45. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. У Lecture notes in computer science (с. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

46. Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481–2495. https://doi.org/10.1109/tpami.2016.2644615

47. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062

48. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis

and Machine Intelligence, 40(4), 834–848. https://doi.org/10.1109/tpami.2017.2699184

49. Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2019). Rethinking atrous convolution for semantic image segmentation. arXiv 2017. arXiv preprint arXiv:1706.05587, 2, 1

50. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-Decoder with atrous separable convolution for semantic image segmentation. У Computer vision – ECCV 2018 (с. 833–851). Springer International Publishing. https://doi.org/10.1007/978-3-030-01234-2_49

51. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/tpami.2016.2577031

52. Takikawa, T., Acuna, D., Jampani, V., & Fidler, S. (2019). Gated-SCNN: Gated shape cnns for semantic segmentation. У 2019 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2019.00533

53. Wu, H., Zhang, J., Huang, K., Liang, K., & Yu, Y. (2019). Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. arXiv preprint arXiv:1903.11816.

54. Cheng, B., Schwing, A., & Kirillov, A. (2021). Per-pixel classification is not all you need for semantic segmentation. Advances in neural information processing systems, 34, 17864-17875

55. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End object detection with transformers. У Computer vision – ECCV 2020 (с. 213–229). Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_13

56. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. Advances in neural information processing systems, 34, 12077-12090

57. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.00135

58. Jain, J., Li, J., Chiu, M., Hassani, A., Orlov, N., & Shi, H. (2023). OneFormer: One transformer to rule universal image segmentation. У 2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52729.2023.00292

59. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. У 2021 IEEE/CVF international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv48922.2021.00986

60. Ding, J., Xue, N., Xia, G.-S., & Dai, D. (2022). Decoupling zero-shot semantic segmentation. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.01129

61. Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., Wang, X.-Z., & Wu, Q. M. J. (2022). A review of generalized zero-shot learning methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–20. https://doi.org/10.1109/tpami.2022.3191696

62. Xian, Y., Schiele, B., & Akata, Z. (2017). Zero-Shot learning — the good, the bad and the ugly. У 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2017.328

63. Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a Few Examples. ACM Computing Surveys, 53(3), 1–34. https://doi.org/10.1145/3386252

64. Gharoun, H., Momenifar, F., Chen, F., & Gandomi, A. (2024). Meta-learning approaches for few-shot learning: A survey of recent advances. ACM Computing Surveys. https://doi.org/10.1145/3659943

65. Luo, X., Wu, H., Zhang, J., Gao, L., Xu, J., & Song, J. (2023, July). A closer look at few-shot classification again. In International Conference on Machine Learning (pp. 23103-23123). PMLR

66. Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Conditional prompt learning for vision-language models. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.01631

67. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., & Bai, X. (2022). A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. У Lecture notes in computer science (с. 736–753). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-19818-2_42

68. Wu, J., Li, X., Xu, S., Yuan, H., Ding, H., Yang, Y., Li, X., Zhang, J., Tong, Y., Jiang, X., Ghanem, B., & Tao, D. (2024). Towards open vocabulary learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–20. https://doi.org/10.1109/tpami.2024.3361862

69. Li, B., Weinberger, K. Q., Belongie, S., Koltun, V., & Ranftl, R. (2022). Language-driven semantic segmentation. arXiv preprint arXiv:2201.03546

70. Bayoudh, K., Knani, R., Hamdaoui, F., & Mtibaa, A. (2021). A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets. The Visual Computer. https://doi.org/10.1007/s00371-021-02166-7

71. Zhang, Y., Sidibé, D., Morel, O., & Mériaudeau, F. (2020). Deep multimodal fusion for semantic image segmentation: A survey. Image and Vision Computing, 104042. https://doi.org/10.1016/j.imavis.2020.104042

72. Yuan, Y., Li, Z., & Zhao, B. (2025). A survey of multimodal learning: Methods, applications, and future. ACM Computing Surveys. https://doi.org/10.1145/3713070

73. Hu, R., Rohrbach, M., Andreas, J., Darrell, T., & Saenko, K. (2017). Modeling relationships in referential expressions with compositional modular networks. У 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2017.470

74. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., & Jia, J. (2024). Lisa: Reasoning segmentation via large language model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (c. 9579-9589)

75. Ji, L., Du, Y., Dang, Y., Gao, W., & Zhang, H. (2024). A survey of methods for addressing the challenges of referring image segmentation. Neurocomputing, 583, 127599. https://doi.org/10.1016/j.neucom.2024.127599

76. Yang, Z., Wang, J., Tang, Y., Chen, K., Zhao, H., & Torr, P. H. S. (2022). LAVT: Language-aware vision transformer for referring image segmentation. У 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr52688.2022.01762

77. Danelljan, M., Ke, L., Liu, Y., Tai, Y.-W., Tang, C.-K., Ye, M., & Yu, F. (2023). Segment anything in high quality. У Advances in neural information processing systems 36 (c. 29914–29934). Neural Information Processing Systems Foundation, Inc. (NeurIPS). https://doi.org/10.52202/075280-1303

78. Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From yolov1 to yolov8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680–1716. https://doi.org/10.3390/make5040083

79. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2024). Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. У Lecture notes in computer science (c. 38–55). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-72970-6_3

80. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., & Martel, A. L. (2021). Loss odyssey in medical image segmentation. Medical Image Analysis, 71, 102035. https://doi.org/10.1016/j.media.2021.102035

81. Li, C., Liu, K., & Liu, S. (2025). A survey of loss functions in deep learning. Mathematics, 13(15), 2417. https://doi.org/10.3390/math13152417

82. Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2), 318–327. https://doi.org/10.1109/tpami.2018.2858826

83. Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Jorge Cardoso, M. (2017). Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. У Deep learning in medical image analysis and multimodal learning for clinical decision support (с. 240–248). Springer International Publishing. https://doi.org/10.1007/978-3-319-67558-9_28

84. Jadon, S. (2020). A survey of loss functions for semantic segmentation. У 2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB). IEEE. https://doi.org/10.1109/cibcb48159.2020.9277638

85. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. У 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2019.00075

86. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07), 12993–13000. https://doi.org/10.1609/aaai.v34i07.6999

87. Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., & Zuo, W. (2021). Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE transactions on cybernetics, 52(8), 8574-8586

88. Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., & Ben Ayed, I. (2021). Boundary loss for highly unbalanced segmentation. Medical Image Analysis, 67, 101851. https://doi.org/10.1016/j.media.2020.101851

89. Karimi, D., & Salcudean, S. E. (2020). Reducing the hausdorff distance in medical image segmentation with convolutional neural networks. IEEE

Transactions on Medical Imaging, 39(2), 499–513. https://doi.org/10.1109/tmi.2019.2930068

90. Chen, X., Williams, B. M., Vallabhaneni, S. R., Czanner, G., Williams, R., & Zheng, Y. (2019). Learning active contour models for medical image segmentation. У 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2019.01190

91. Chan, T. F., & Vese, L. A. (2001). Active contours without edges. IEEE Transactions on Image Processing, 10(2), 266–277. https://doi.org/10.1109/83.902291

92. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. У 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2015.7298682

93. Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. У Computer vision – ECCV 2016 (с. 499–515). Springer International Publishing. https://doi.org/10.1007/978-3-319-46478-7_31

94. Hadsell, R., Chopra, S., & LeCun, Y. (б. д.). Dimensionality reduction by learning an invariant mapping. У 2006 IEEE computer society conference on computer vision and pattern recognition - volume 2 (CVPR'06). IEEE. https://doi.org/10.1109/cvpr.2006.100

95. Van Dongen, S. (2000). Performance criteria for graph clustering and Markov cluster experiments. Report-Information systems, (12), 1-36

96. Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. У The fifth ACM SIGKDD international conference. ACM Press. https://doi.org/10.1145/312129.312186

97. Mirkin, B. (2013). Mathematical classification and clustering (Vol. 11). Springer Science & Business Media

98. Meilă, M. (2007). Comparing clusterings—an information based distance. Journal of Multivariate Analysis, 98(5), 873–895. https://doi.org/10.1016/j.jmva.2006.11.013

99. Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. International journal of computer vision, 40(2), 99-121

100. Stehling, R. O., Nascimento, M. A., & Falcão, A. X. (2002). MiCRoM: A metric distance to compare segmented images. У Recent advances in visual information systems (с. 12–23). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45925-1_2

101. Maddison, C. J., Mnih, A., & Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712

102. Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144

103. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. У 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2016.90

104. Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929

105. Perez, E., Strub, F., De Vries, H., Dumoulin, V., & Courville, A. (2018). FiLM: Visual reasoning with a general conditioning layer. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1). https://doi.org/10.1609/aaai.v32i1.11671

106. Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., & Chen, L.-C. (2020). Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. У 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.01249

107. Sun, B., Kuen, J., Lin, Z., Mordohai, P., & Chen, S. (2023). PRN: Panoptic refinement network. У 2023 IEEE/CVF winter conference on applications of computer vision (WACV). IEEE. https://doi.org/10.1109/wacv56688.2023.00395

108. Odena, A., Dumoulin, V., & Olah, C. (2016). Deconvolution and checkerboard artifacts. Distill, 1(10). https://doi.org/10.23915/distill.00003

109. Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256). JMLR Workshop and Conference Proceedings

110. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. У 2015 IEEE international conference on computer vision (ICCV). IEEE. https://doi.org/10.1109/iccv.2015.123

111. Aitken, A., Ledig, C., Theis, L., Caballero, J., Wang, Z., & Shi, W. (2017). Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. arXiv preprint arXiv:1707.02937

112. Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr

113. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A comprehensive survey on transfer learning. Proceedings of the IEEE, 109(1), 43–76. https://doi.org/10.1109/jproc.2020.3004555

114. Gupta, A., Dollar, P., & Girshick, R. (2019). LVIS: A dataset for large vocabulary instance segmentation. У 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2019.00550

115. Wu, C., Lin, Z., Cohen, S., Bui, T., & Maji, S. (2020). PhraseCut: Language-based image segmentation in the wild. У 2020 IEEE/CVF conference on

computer vision and pattern recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr42600.2020.01023

116. Liu, R., Lehman, J., Molino, P., Petroski Such, F., Frank, E., Sergeev, A., & Yosinski, J. (2018). An intriguing failing of convolutional neural networks and the coordconv solution. Advances in neural information processing systems, 31

ДОДАТОК А

Список публікацій за темою дисертації

1. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Experimental study of distance map decoder architectural configurations for instance segmentation by text query. System technologies, 2(163), 3–12. https://doi.org/10.34185/1562-9945-2-163-2026-01

Ключові слова: сегментація екземплярів, декодер відстаней, PixelShuffle, конволюція координат, об’єднання ознак, CLIP, сегментація з відкритим словником, InstanceCLIPSeg.

Категорія: Б

2. Kovtunenko, A. R., & Mashtalir, S. V. (2026). Metrical loss functions for image segmentation based on convolutional neural networks. Applied Aspects of Information Technology, 9(2), 172–184. https://doi.org/10.15276/aait.09.2026.12

Ключові слова: сегментація зображень, функції втрат, згорткові нейронні мережі, глибоке навчання, метрика, розбиття, комп’ютерний зір.

Категорія: Б.

3. Kovtunenko, A. R., & Mashtalir, S. V. (2025). Improved segmentation model to identify object instances based on textual prompts. Herald of Advanced Information Technology, 8(1), 54–66. https://doi.org/10.15276/hait.08.2025.4

Ключові слова: глибоке навчання, сегментація зображень, згорткові нейронні мережі, архітектури-трансформери, контрастна мовно-образна підготовка, сегментація з нефіксованим набором класів

Категорія: А.

4. Kovtunenko, A. R., & Mashtalir, S. V. (2025). A Review of Modern Neural Network Architectures for Image Segmentation. Management Information System and Devises, (185), 53–62. https://doi.org/10.30837/0135-1710.2025.185.043

Ключові слова: комп'ютерний зір, сегментація зображень, нейронні мережі,

архітектури-трансформери, сегментація без попереднього навчання Категорія: Б.

5. Ковтуненко, А. Р. (2025). Мультимодальна висхідна сегментація об’єктів за текстовим запитом. У III Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі» (с. 11–12). Західноукраїнський національний університет
Тези доповіді.

6. Ковтуненко, А., & Машталір, В. (2024). Аналіз та порівняння функцій втрат для вирішення задачі сегментації. У Радіоелектроніка та молодь у XXI столітті. Т. 7: Конференція «Комп’ютерний зір, системний аналіз та математичне моделювання». Press of the Kharkiv National University of Radioelectronics. (с. 57–59) https://doi.org/10.30837/iyf.cvsamm.2024.057 Тези доповіді.

ДОДАТОК Б

Приклади та порівняння результатів роботи моделей

Рисунок Б.1 – Результат роботи InstanceCLIPSeg за запитом «a photo of a bottle»

Рисунок Б.2 – Результат роботи InstanceCLIPSeg за запитом «a photo of a milk»

Рисунок Б.3 – Результат роботи InstanceCLIPSeg за запитом «a photo of a man in a black jacket»

Рисунок Б.4 – Результат роботи InstanceCLIPSeg за запитом «a photo of a light-reflective vest»

Рисунок Б.5 – Результат роботи InstanceCLIPSeg за запитом «a photo of a bus»

Рисунок Б.6 – Результат роботи InstanceCLIPSeg за запитом «a photo of a gray car in the middle»

Рисунок Б.7 – Порівняння моделей за запитом «apple»
(ліворуч – HQ-SAM, праворуч – InstanceCLIPSeg)

Рисунок Б.8 – Порівняння моделей за запитом «banana»
(ліворуч – HQ-SAM, праворуч – InstanceCLIPSeg)

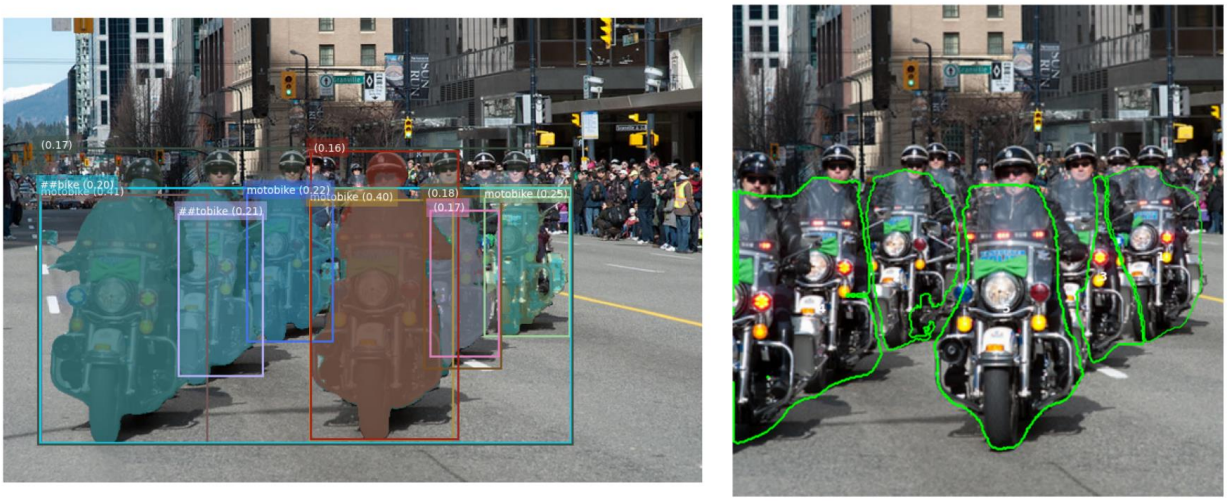

Рисунок Б.9 – Порівняння моделей за запитом «motobike»
(ліворуч – HQ-SAM, праворуч – InstanceCLIPSeg)

ДОДАТОК В

Акти впровадження

ЗАТВЕРДЖУЮ
Директор ТОВ Сайтосс
В. В. Луців
«30» квітня 2026 р.

АКТ

про використання результатів дисертації «Нейромережеві методи мультимодальної просторово-семантичної сегментації за нечітким запитом» Ковтуненка Андрія Романовича

Відповідальний за впровадження директор ТОВ Сайтосс Луців В.В. цим актом підтверджує, що результати дисертаційної роботи Ковтуненка А.Р., яку виконано на кафедрі інформатики Харківського національного університету радіоелектроніки, впроваджено у ТОВ Сайтосс з метою організації швидкого пошуку необхідних подій у системі безпеки за допомогою нечітких текстових запитів.

Результати, які отримано у ході дисертаційного дослідження, знайшли практичне застосування у ТОВ Сайтосс у вигляді створеного інтелектуального модуля пошуку відеофрагментів на основі аналізу мультимодальних даних. У роботі автор обґрунтовує ефективність використання розроблених технологій розв'язання задач пошуку, що базуються на застосуванні нейромережевих моделей для просторово-семантичної сегментації відеопотоку та семантичної обробки нечітких запитів користувача, наводячи конкретні приклади втілення даного методу в діяльності ТОВ Сайтосс.

Планується і надалі використовувати наукові розробки і досвід роботи при проєктуванні систем безпеки, покращенні функцій відеоаналітики та управлінні відеоархівами через використання методів, що запропоновані Ковтуненком А.Р.

Використання результатів дисертаційної роботи Ковтуненка Андрія Романовича дозволило реалізувати можливість швидкого пошуку необхідних подій у системі безпеки за текстовим запитом, що суттєво прискорило дану процедуру та значно підвищило загальну ефективність оперативного моніторингу.

директор ТОВ Сайтосс В. В. Луців

УКРАЇНА
ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
«САЙТОСС»
№37365865
місто Харків

ЗАТВЕРДЖУЮ
Перший проректор ХНУРЕ
_____ Андрій ЄРОХІН
«__» травня 2026 року

АКТ

про впровадження в освітній процес ХНУРЕ
результатів дисертаційної роботи Ковтуненка Андрія Романовича
«Нейромережеві методи мультимодальної просторово-семантичної сегментації за нечітким запитом»,
поданої на здобуття ступеня доктора філософії за спеціальністю 122 Комп'ютерні науки

Комісія у складі голови: завідувача кафедри Інформатики, к.т.н., доц. Кобиліна О.А. та членів комісії: д.т.н., проф. кафедри Інформатики Шафроненко А. Ю., к.т.н., доц. кафедри Інформатики Руденко Д.О. розглянула матеріали дисертаційної роботи Ковтуненка А.Р. і склала акт про впровадження результатів дослідження в освітній процес Харківського національного університету радіоелектроніки.

Склад впровадження:

1. Нейромережеві методи мультимодальної просторово-семантичної сегментації зображень на основі нечітких текстових запитів;
2. Архітектури моделей глибокого навчання для спільної обробки та інтеграції візуальних і семантичних даних.

Комісія встановила, що результати дисертаційної роботи Ковтуненка А.Р. були впроваджені в освітній процес кафедри Інформатики Харківського національного університету радіоелектроніки при проведенні практичних і лабораторних занять з дисциплін «Розпізнавання образів» для здобувачів першого (бакалаврського) та «Комп'ютерний зір» для здобувачів другого (магістерського) рівнів вищої освіти зі спеціальності F3 Комп'ютерні науки.

Голова комісії _____ Олег КОБИЛІН
Члени комісії _____ Аліна ШАФРОНЕНКО
_____ Діана РУДЕНКО