

# BÁO CÁO ĐỒ ÁN CUỐI KỲ

Môn học: CS519 - PHƯƠNG PHÁP LUẬN NCKH

Lớp: CS519.011

GV: PGS.TS. Lê Đình Duy

Trường ĐH Công Nghệ Thông Tin, ĐHQG-HCM



# PHÂN ĐOẠN THỰC THỂ TRONG THẾ GIỚI MỞ SỬ DỤNG XÁC SUẤT VỀ ĐỐI TƯỢNG TRÊN INSTANCE ACTIVATION MAP

Trần Hoàng Bảo Ly - 21521109



# Tóm tắt

- Lớp: CS519.011
- Link Github của nhóm:  
<https://github.com/4ursmile/CS519.011>
- Link YouTube video:  
[https://www.youtube.com/watch?v=siuVTBAxUV0&ab\\_channel=Asucansee](https://www.youtube.com/watch?v=siuVTBAxUV0&ab_channel=Asucansee)



Trần Hoàng Bảo Ly



# Giới thiệu

- Phân đoạn thực thể có nhiều ứng dụng: Xe tự hành, tự động hóa, y tế,...
- Mô hình phân đoạn thực thể truyền thống phụ thuộc vào sự hạn chế của tập huấn luyện.
- Bài toán phân đoạn thực thể thế giới mở (OWIS) gồm 2 mục tiêu [1]:
  - 1) Phân đoạn các thực thể đã biết và cả các thực thể chưa biết.
  - 2) Học tăng cường để mở rộng các lớp thực thể mới.

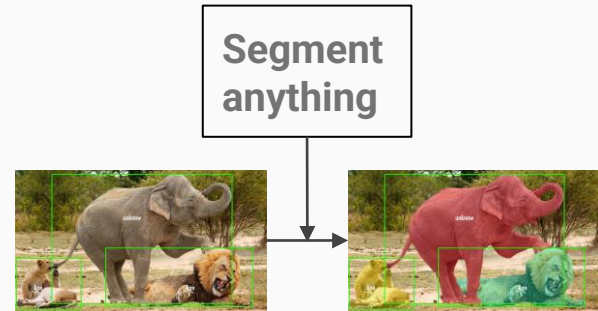


# Giới thiệu

- Các phương pháp hiện tại.



Chỉ phát hiện thực thể, không quan tâm đến lớp đối tượng [2, 3].



Phụ thuộc vào mô hình phát hiện đối tượng trong thế giới mở [4, 5].

# Mục tiêu

- Nghiên cứu phát triển mô hình phân đoạn thực thể cho thế giới mở, đồng thời giải quyết cả hai mục tiêu 1) và 2)
- Áp dụng phương pháp đánh giá mô hình trong thế giới mở từ [1] để đề xuất phương thức đánh giá toàn diện cho bài toán phân đoạn thực thể trong thế giới mở, đáp ứng cả hai mục tiêu 1) và 2).



# Nội dung

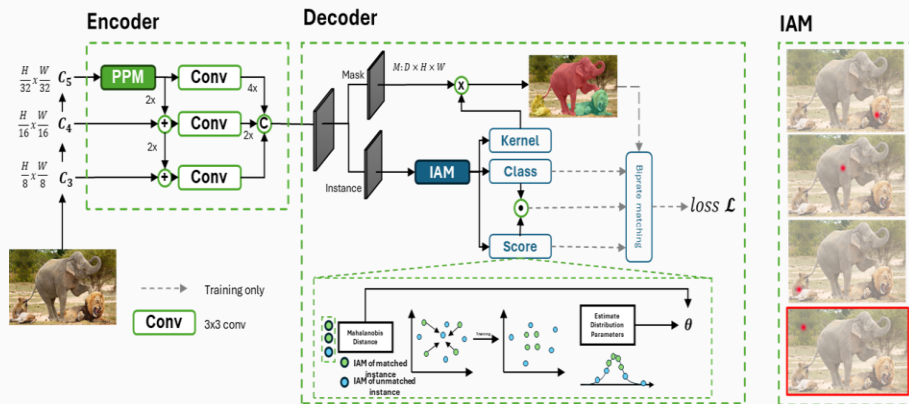
- Nghiên cứu và khảo sát kết hợp các phương pháp máy học thể giới mở [4], phương pháp phân đoạn thực thể.
- Điều chỉnh kiến trúc mô hình phân đoạn thực thể dựa trên [4] và [6], tạo ra mô hình phân đoạn thực thể trong thể giới mở.
- Tạo bộ dữ liệu để huấn luyện và đánh giá mô hình phân đoạn thực thể trong thể giới mở.
- Huấn luyện mô hình với bộ dữ liệu mới, sau đó so sánh và đánh giá hiệu suất với các phương pháp và thiết lập khác.
- Xây dựng công cụ command line và Python API



# Phương pháp

Tìm hiểu, nghiên cứu và áp dụng phương pháp xác xuất về đối tượng vào mô hình SparseInst[6].

Nghiên cứu và xây dựng bộ dữ liệu để đánh giá mô hình OWIS.



Dataset	
30 classes	
151K Images	
Pascal VOC	MS COCO
21K	130K



# Kết quả dự kiến

- Báo cáo về mô hình phân đoạn thực thể trong thế giới mở mới, bao gồm phương pháp giải quyết bài toán, kết quả thực nghiệm, so sánh và đánh giá đối với các thiết lập khác nhau và mô hình OWIS hiện tại.
- Tập dữ liệu chứa hơn 150K ảnh và 30 lớp thực thể để huấn luyện và đánh giá mô hình.
- Cung cấp công cụ command line và Python API để sử dụng, huấn luyện, đánh giá, và tái tạo.



# Tài liệu tham khảo

- [1]. K. J. Joseph, Salman H. Khan, Fahad Shahbaz Khan, Vineeth N. Balasubramanian: Towards Open World Object Detection. CVPR 2021: 5830-5840.
- [2]. Weiyao Wang, Matt Feiszli, Heng Wang, Jitendra Malik, Du Tran: Open-World Instance Segmentation: Exploiting Pseudo Ground Truth From Learned Pairwise Affinity. CVPR 2022: 4412-4422.
- [3]. TXizhe Xue, Dongdong Yu, Lingqiao Liu, Yu Liu, Satoshi Tsutsui, Ying Li, Zehuan Yuan, Ping Song: Transformer-based Open-world Instance Segmentation with Cross-task Consistency Regularization. ACM Multimedia 2023: 2507-2515.
- [4]. Orr Zohar, Kuan-Chieh Wang, Serena Yeung: PROB: Probabilistic Objectness for Open World Object Detection. CVPR 2023: 11444-11453.
- [5]. Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, Lei Zhang: Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. CoRR abs/2303.05499 (2023).
- [6]. Tianheng Cheng, Xinggang Wang, Shaoyu Chen, Wenqiang Zhang, Qian Zhang, Chang Huang, Zhaoxiang Zhang, Wenyu Liu: Sparse Instance Activation for Real-Time Instance Segmentation. CVPR 2022: 4423-4432.
- [7]. Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, C. Lawrence Zitnick: Microsoft COCO: Common Objects in Context. ECCV (5) 2014: 740-755.
- [8]. Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, Andrew Zisserman: The Pascal Visual Object Classes (VOC) Challenge. Int. J. Comput. Vis. 88(2): 303-338 (2010).

