

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**NGUYỄN THÀNH DANH**

**PHÂN ĐOẠN THỰC THỂ NGỤY TRANG  
DỰA TRÊN ĐẶC TRƯNG CÓ TÍNH PHÂN BIỆT CAO**

**LUẬN VĂN THẠC SĨ  
NGÀNH KHOA HỌC MÁY TÍNH**

**Mã số: 8 48 01 01**

**THÀNH PHỐ HỒ CHÍ MINH – 2023**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**NGUYỄN THÀNH DANH**

**PHÂN ĐOẠN THỰC THỂ NGUY TRẠNG  
DỰA TRÊN ĐẶC TRƯNG CÓ TÍNH PHÂN BIỆT CAO**

**LUẬN VĂN THẠC SĨ**

**NGÀNH KHOA HỌC MÁY TÍNH**

**Mã số: 8 48 01 01**

**NGƯỜI HƯỚNG DẪN KHOA HỌC**

**TS. NGUYỄN VINH TIỆP**

**THÀNH PHỐ HỒ CHÍ MINH – 2023**

# DANH SÁCH HỘI ĐỒNG PHẢN BIỆN

Hội đồng Phản biện Luận văn Thạc sĩ được thành lập theo Quyết định số 1302/QĐ-ĐHCNTT, ngày 13 tháng 12 năm 2023 của Hiệu trưởng Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh.

1. Chủ tịch: PGS.TS. Lê Hoàng Thái
2. Thư ký: TS. Dương Việt Hằng
3. Phản biện 1: PGS.TS. Nguyễn Thanh Bình
4. Phản biện 2: TS. Mai Tiến Dũng
5. Ủy viên: TS. Nguyễn Tấn Trần Minh Khang

# LỜI CAM ĐOAN

Tôi xin cam đoan Luận văn Thạc sĩ với đề tài "Phân đoạn thực thể nguy trang dựa trên đặc trưng có tính phân biệt cao" là công trình nghiên cứu khoa học của tôi và những nội dung được trình bày trong luận văn này là hoàn toàn trung thực. Các cá nhân, tổ chức hỗ trợ tôi trong quá trình thực hiện luận văn đã được đề cập trong Lời cảm ơn. Các công trình khoa học được tôi tham khảo có trích dẫn rõ ràng và liệt kê cụ thể, chính xác trong phần Tài liệu tham khảo. Tôi hoàn toàn chịu trách nhiệm về nội dung của luận văn này.

Học viên cao học



Nguyễn Thành Danh



# LỜI CẢM ƠN

Với tất cả lòng kính trọng và biết ơn, tôi gửi lời tri ân đến những người Thầy đã hướng dẫn tôi trong quá trình thực hiện Luận văn Thạc sĩ này: Thầy Nguyễn Vinh Tiệp, Trưởng Phòng Thí nghiệm Truyền thông Đa phương tiện (PTN TTĐPT) đã đồng hành với tôi trong quá trình thực hiện luận văn; Anh Tâm Nguyễn, Giáo sư tại Đại học Dayton, đã truyền cảm hứng và hỗ trợ tôi trong các nghiên cứu về thực thể ngụy trang; Thầy Ngô Đức Thành, Trưởng Khoa Khoa học máy tính, đã giúp đỡ tôi các vấn đề trong quá trình thực hiện luận văn; và tất cả các Thầy Cô đã giúp đỡ tôi trong suốt quá trình học tập và làm việc tại Trường Đại học Công nghệ Thông tin.

Bên cạnh đó, tôi gửi lời cảm ơn đặc biệt đến Khoa Khoa học máy tính, và PTN TTĐPT đã tạo điều kiện và cung cấp một môi trường làm việc phù hợp để tôi thực hiện đề tài này. Tôi cũng gửi lời cảm ơn đến Quỹ Đổi mới sáng tạo Vingroup (VinIF) đã tạo điều kiện và đồng hành với tôi trong một cách cụ thể với chương trình học bổng Thạc sĩ, Tiến sĩ trong nước năm 2022 (VINIF.2022.ThS.104).

Tôi cũng không quên gửi lời cảm ơn đến bạn bè, đặc biệt là các bạn sinh viên mà tôi có cơ hội gặp gỡ và làm việc tại PTN TTĐPT, cũng như tất cả mọi người, dù cách này hay cách khác, đã đồng hành với tôi trong suốt quãng thời gian vừa qua.

Cuối cùng, tôi xin gửi lời cảm ơn đến gia đình, đã nuôi dưỡng và luôn luôn ủng hộ tôi bằng tình yêu thương để tôi trưởng thành và phát triển trên con đường mà tôi đã chọn, cụ thể hóa bằng luận văn mà tôi đã hoàn thành hôm nay.

Học viên cao học



Nguyễn Thành Danh

# TÓM TẮT

Ngụy trang là một cơ chế tự vệ của các loài động vật trong tự nhiên để chúng ẩn mình vào môi trường xung quanh để tránh được kẻ thù nguy hiểm. Nghiên cứu về đối tượng ngụy trang ở cấp độ thực thể là một chủ đề nghiên cứu đầy thách thức và có nhiều ứng dụng trong thực tiễn. Với thực thể có tính chất ngụy trang, khả năng phân biệt giữa đặc trưng ngụy trang của thực thể và cảnh vật xung quanh có thể giúp các mô hình học máy thực hiện tốt hơn tác vụ của chúng. Các đặc trưng này được xem là đặc trưng có tính phân biệt cao.

Trong ngữ cảnh thực tế, các thực thể động vật ngụy trang thường khó phát hiện bằng mắt thường do tập tính lẩn trốn kẻ thù hay ngụy trang vào môi trường tự nhiên, dẫn đến việc thu thập dữ liệu bị giới hạn. Vì thế, câu hỏi đặt ra là làm sao nhận định và khai thác các đặc trưng có tính phân biệt cao của các thực thể này so với môi trường để giúp mô hình học máy sử dụng dữ liệu hiệu quả hơn trong quá trình huấn luyện? Đồng thời, trong điều kiện ít dữ liệu huấn luyện, làm cách nào để các mô hình có khả năng đưa ra dự đoán chính xác hơn?

Trong luận văn này, chúng tôi tập trung nghiên cứu ở khía cạnh khai thác các đặc trưng có tính phân biệt cao giữa thực thể và vùng nền để giải quyết bài toán phân đoạn thực thể ngụy trang. Hơn nữa, chúng tôi cũng đề xuất hướng giải quyết trong trường hợp đặc thù ít dữ liệu huấn luyện của thực thể ngụy trang. Cụ thể, đóng góp của chúng tôi trong luận văn này gồm 03 điểm chính sau:

- Nghiên cứu, đề xuất **mô hình một giai đoạn CE-OST [CT1]** dựa trên kiến trúc Transformer tăng cường đặc trưng biên cạnh (contour emphasis) để phân đoạn thực thể ngụy trang.
- Nghiên cứu, đề xuất **mô hình hai giai đoạn FS-CDIS [CT2, CT3]** đặc thù cho ngữ cảnh ít dữ liệu để giải quyết bài toán phân đoạn thực thể ngụy trang dựa trên kỹ thuật học tương phản (contrastive learning) với hàm mất mát ba thành phần ở cấp độ thực thể (Instance Triplet Loss) và sử dụng bộ nhớ lưu trữ thực thể (Instance Memory Storage).

- Đề xuất **tập dữ liệu ảnh CAMO-FS** [CT2, CT3] cho bài toán phát hiện và phân đoạn thực thể nguy trang, được tinh chỉnh và tạo lập cấu trúc phục vụ hướng tiếp cận học ít dữ liệu.

Chúng tôi tiến hành thực nghiệm đánh giá độ chính xác các mô hình đề xuất trên các tập dữ liệu chuẩn tiêu biểu trong nghiên cứu đối tượng nguy trang như COD10K, NC4K, CAMO++ và tập dữ liệu đề xuất CAMO-FS.

# Mục lục

<b>1</b>	<b>TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU</b>	<b>1</b>
1.1	Giới thiệu đề tài . . . . .	1
1.2	Định nghĩa bài toán . . . . .	4
1.3	Mục tiêu của luận văn . . . . .	7
1.3.1	Tăng cường đặc trưng phân biệt ở vùng biên cạnh để phân đoạn hiệu quả thực thể nguy trang . . . . .	7
1.3.2	Khai thác đặc trưng phân biệt với ít dữ liệu huấn luyện để phân đoạn thực thể nguy trang . . . . .	8
1.4	Đóng góp chính của luận văn . . . . .	9
1.5	Bố cục luận văn . . . . .	10
<b>2</b>	<b>CÔNG TRÌNH LIÊN QUAN</b>	<b>11</b>
2.1	Tổng quan nghiên cứu về thực thể nguy trang . . . . .	11
2.2	Các kiến trúc phân đoạn thực thể nguy trang . . . . .	13
2.2.1	Phân đoạn thực thể với kiến trúc hai giai đoạn . . . . .	13
2.2.2	Phân đoạn thực thể với kiến trúc một giai đoạn . . . . .	19
2.2.3	Hướng tiếp cận sử dụng ít dữ liệu huấn luyện . . . . .	24
2.3	Các hướng tiếp cận khai thác đặc trưng có tính phân biệt cao . . . . .	26
2.3.1	Tăng cường đặc trưng biên cạnh . . . . .	26
2.3.2	Phương pháp học tương phản . . . . .	30
2.4	Các tập dữ liệu chuẩn về thực thể nguy trang . . . . .	31
2.5	Tạm kết . . . . .	34

<b>3</b>	<b>Mô hình CE-OST khai thác đặc trưng vùng biên cạnh</b>	<b>36</b>
3.1	Tổng quan . . . . .	36
3.2	Mô hình Transformer một giai đoạn CE-OST . . . . .	37
3.2.1	Giới thiệu mô hình . . . . .	37
3.2.2	Khối tăng cường đặc trưng biên cạnh . . . . .	38
3.2.3	Khối Transformer phân đoạn thực thể nguy trang . . . . .	42
3.3	Thực nghiệm . . . . .	45
3.3.1	Cấu hình thực nghiệm. . . . .	45
3.3.2	Kết quả thực nghiệm . . . . .	46
3.3.3	Thực nghiệm loại suy. . . . .	47
3.4	Tạm kết . . . . .	51
<b>4</b>	<b>Mô hình FS-CDIS học đặc trưng phân biệt với ít mẫu dữ liệu</b>	<b>52</b>
4.1	Tổng quan . . . . .	52
4.2	Bộ dữ liệu đề xuất CAMO-FS . . . . .	53
4.3	Mô hình FS-CDIS phân đoạn thực thể nguy trang với ít mẫu dữ liệu . . . . .	59
4.3.1	Giới thiệu mô hình . . . . .	59
4.3.2	Khai thác đặc trưng nguy trang với kỹ thuật học tương phản . . . . .	61
4.3.3	Củng cố đặc trưng nguy trang với Bộ nhớ lưu trữ thực thể . . . . .	63
4.4	Thực nghiệm . . . . .	65
4.4.1	Cấu hình thực nghiệm . . . . .	65
4.4.2	Kết quả thực nghiệm . . . . .	66
4.4.3	Thực nghiệm loại suy . . . . .	70
4.5	Tạm kết . . . . .	73
<b>5</b>	<b>KẾT LUẬN</b>	<b>75</b>
5.1	Kết quả đạt được . . . . .	75
5.2	Hướng phát triển . . . . .	77
5.2.1	Cải tiến các đặc trưng có tính phân biệt cao . . . . .	77
5.2.2	Áp dụng hướng tiếp cận cho bài toán trên ảnh y khoa . . . . .	77

**CÔNG BỐ KHOA HỌC**

**78**

**Tài liệu tham khảo**

**79**

# Danh sách hình vẽ

1.1	Ảnh chụp một số cá thể động vật quý hiếm trong tự nhiên. Các loài động vật này mang những màu sắc, đường nét tương đồng với môi trường xung quanh hoặc có tập tính ẩn mình để săn mồi. Các yếu tố này khiến chúng trở nên khó bị phát hiện. . . . .	2
1.2	Minh họa ý tưởng phát hiện các đối tượng hay thực thể có tính chất nguy trang có thể áp dụng tự động trong các chiến dịch tìm kiếm và giải cứu con người hay các loài động vật khi sử dụng hình ảnh từ thiết bị bay không người lái hoặc hình ảnh từ các camera giám sát. . . . .	3
1.3	Minh họa đầu vào và đầu ra của bài toán phân đoạn thực thể nguy trang. Đầu vào là ảnh chứa các thực thể nguy trang, đầu ra là mặt nạ phân đoạn ngữ nghĩa của từng thực thể nguy trang. . . . .	4
1.4	Một số hình ảnh minh họa cho dữ liệu ảnh thực thể nguy trang được trích từ tập dữ liệu CAMO++ [36]. . . . .	5
1.5	So sánh sự tương đồng về mặt thị giác của các vùng ảnh trong ảnh có chứa thực thể nguy trang. Thứ tự gán nhãn: (a) ảnh gốc, (b) ảnh với đường biên cạnh, (c) ô cắt vùng thực thể, (d) ô cắt vùng nền, (e, f) ô cắt vùng biên vật thể và vùng nền. . . . .	7
1.6	Ý tưởng tổng quát của mô hình học dựa trên ít mẫu dữ liệu huấn luyện. Quá trình học gồm 2 giai đoạn: giai đoạn cơ sở cung cấp tri thức tổng quát cho mô hình và giai đoạn tinh chỉnh giúp mô hình học dựa trên ít mẫu dữ liệu huấn luyện gán nhãn. . . . .	8
2.1	Kiến trúc mô hình Mask Scoring RCNN [30] với nhánh MaskIOU là điểm cải tiến chính được đề xuất. . . . .	14

2.2	Kiến trúc mô hình Cascade R-CNN [2]. "I" là ảnh đầu vào, "conv" là lớp tích chập rút trích đặc trưng, "pool" là bộ trích xuất đặc trưng theo vùng (region-wise), "H" là đầu ra theo các tác vụ, "B" là kết quả khung bao, "C" là kết quả phân loại, và "B0" là các vùng đề xuất khởi tạo của mạng.	15
2.3	Kiến trúc mô hình PANet [44] hạn chế việc mất mát thông tin và tăng cường thông tin cho tác vụ phân đoạn thực thể. . . . .	16
2.4	Kiến trúc mô hình HTC [6] cải thiện luồng thông tin bằng cách xếp tầng (cascading) và xử lý đa tác vụ tại mỗi giai đoạn (multi-tasking). . . . .	16
2.5	Kiến trúc mô hình BlendMask [5] với khả năng tổng hợp đặc trưng ở độ phân giải cao thông qua một mạng FPN. . . . .	17
2.6	Kiến trúc mô hình Mask Transfiner [33] với cơ chế sử dụng cây tứ phân (Quadtree) để học đặc trưng tại các vùng dễ gặp lỗi và mô-đun Quadtree Transformer để tạo ra mặt nạ phân đoạn hiệu quả. . . . .	18
2.7	Kiến trúc mô hình Mask DCNet [48] với 2 khối chức năng lần lượt phục vụ ở cấp độ điểm ảnh và cấp độ thực thể. . . . .	19
2.8	Kiến trúc mô hình YOLACT [1]. Kiến trúc xây dựng trên nền tảng RetinaNet [42] với backbone ResNet-101 + FPN. Các khối màu xám là những khối chức năng không được cập nhật trong quá trình huấn luyện.	20
2.9	Kiến trúc mô hình CondInst [71] với các đầu ra phân đoạn (mask heads) chuyên biệt cho từng thực thể của ảnh đầu vào. . . . .	21
2.10	Kiến trúc mô hình SOLO-v1 [78] với hai nhánh phân loại và phân đoạn ngữ nghĩa. . . . .	22
2.11	Kiến trúc mô hình SOTR [23] với các mô-đun tận dụng điểm mạnh của kiến trúc CNNs và Transformer. . . . .	23
2.12	Kiến trúc mô hình OSFormer [59] với khối Transformer và hai khối chức năng được đề xuất là DCIN và CFF. . . . .	23
2.13	Kiến trúc mô hình TFA [77] phát hiện và phân đoạn thực thể với hướng tiếp cận học ít dữ liệu (few-shot learning). Đây là một trong những kiến trúc nền tảng cho hướng tiếp cận này. . . . .	25



2.14	Kiến trúc mô hình HED [82] phát hiện biên cạnh vật thể. . . . .	28
2.15	Kiến trúc mô hình COB [50] phát hiện biên cạnh có hướng. . . . .	29
2.16	Trực quan hóa hướng tiếp cận học tương phản với các mẫu biểu diễn Positive, Negative, và Anchor. Quá trình học tương phản sẽ tìm cách thu hẹp khoảng cách giữa các điểm biểu diễn Positive và Anchor, trong khi đẩy xa khoảng cách giữa các điểm biểu diễn Negative và Anchor. . . . .	30
2.17	Một số hình ảnh trích từ tập dữ liệu MoCA [35] với nhãn khung bao và optical flow. . . . .	32
2.18	Một số hình ảnh ngẫu trang trích từ tập dữ liệu COD10K [12] với nhãn phân đoạn thực thể. COD10K chứng minh tính đa dạng với nhiều điều kiện xuất hiện thực thể ngẫu trang khác nhau như bị che khuất (OC), nhiều thực thể (MO), thực thể nhiều kích thước khác nhau (BO) . . . . .	33
2.19	Một số hình ảnh ngẫu trang trích từ tập dữ liệu CAMO++ [36]. . . . .	34
3.1	Tổng quan mô hình CE-OST (Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation) phân đoạn thực thể ngẫu trang dựa trên kiến trúc Transformer một giai đoạn có tăng cường đặc trưng biên cạnh. . . . .	38
3.2	Sơ đồ minh họa hoạt động của Lưới điều kiện (Grid-Condition) trong mô-đun Tăng cường đặc trưng biên cạnh (Contour Emphasis). . . . .	42
3.3	Trực quan hóa kết quả trên tập dữ liệu CAMO++ [36] với mô hình CE-OST sử dụng backbone PVT. Ngưỡng tin cậy được sử dụng là 0.5. . . . .	49
3.4	Một số hình ảnh minh họa cho ảnh đầu vào được tăng cường đặc trưng biên cạnh, dữ liệu từ tập CAMO++ [36]. Dòng đầu tiên thể hiện các vùng được phóng đại. . . . .	50
4.1	Phân phối lớp của các mẫu dữ liệu ngẫu trang dưới dạng wordcloud giữa tập dữ liệu CAMO-FS đề xuất và COD10K [12]. . . . .	55
4.2	Phân phối độ phân giải ảnh trên các tập dữ liệu về thực thể ngẫu trang: CAMO-FS, CAMO++ [36], CAMO [38], và COD10K [12]. . . . .	57
4.3	Độ lệch tâm thực thể của các bộ dữ liệu ngẫu trang . . . . .	58

- 4.4 Mô hình FS-CDIS đề xuất cho phân đoạn thực thể nguy trang sử dụng ít dữ liệu huấn luyện. . . . . 60
- 4.5 Mô hình hóa hàm mất mát ba thành phần với các đặc trưng tiền cảnh ( $\mathbb{F}_{fg}$ ), vùng nền ( $\mathbb{F}_{bg}$ ), và điểm neo ( $\mathbb{F}_{ac}$ ). Hàm mất mát ba thành phần có mục tiêu thu hẹp khoảng cách giữa các đặc trưng tiền cảnh với điểm neo, và gia tăng khoảng cách giữa các đặc trưng vùng nền với điểm neo. 62
- 4.6 Mô hình hóa bộ nhớ lưu trữ thực thể. Bộ nhớ lưu trữ thực thể lưu trữ thông tin vùng nền và vùng thực thể theo từng lớp ngữ nghĩa và sử dụng chúng để tính toán sự phân biệt cho nhánh phân đoạn thực thể. Bộ nhớ lưu trữ thực thể được cập nhật liên tục trong quá trình huấn luyện với mỗi mẫu dữ liệu mới theo từng lớp ngữ nghĩa. . . . . 64
- 4.7 Kết quả so sánh định tính giữa mô hình baseline MTFA [20] và các đề xuất của chúng tôi. Kết quả được lấy từ cấu hình 5 mẫu huấn luyện. “Memory” là Instance Memory Storage và “Triplet” là Instance Triplet Loss. Các thực thể được dự đoán với ngưỡng tin cậy là 0.5, số lượng dự đoán không đáng tin cậy phần lớn đã được loại bỏ. Hai hàng cuối cùng thể hiện các trường hợp mà hàm mất mát ba thành phần hoặc bộ nhớ lưu trữ thực thể chưa giải quyết được. . . . . 74

# Danh sách bảng

2.1	Thống kê một số tập dữ liệu về thực thể nguy trang (chỉ xét dữ liệu ảnh/video chứa thực thể nguy trang). . . . .	31
3.1	So sánh với các mô hình tiên tiến nhất trên tập dữ liệu COD10K [12] và NC4K [49] (cùng sử dụng mô hình cơ sở ResNet-101 [26]) . . . . .	47
3.2	Thực nghiệm loại suy về các mô hình backbone của CE-OST trên tập dữ liệu COD10K [12], NC4K [49], và CAMO++ [36]. . . . .	48
4.1	Số lượng mẫu thu thập thêm cho bộ dữ liệu đề xuất CAMO-FS. . . . .	55
4.2	Tỉ lệ số lượng thực thể nguy trang trung bình trên ảnh của tập dữ liệu CAMO-FS đề xuất. . . . .	56
4.3	So sánh độ chính xác trên tập dữ liệu CAMO-FS giữa các mô hình tiên tiến như MTFA [20], Mask RCNN <sup>†</sup> [28], iFS-RCNN [56], và mô hình đề xuất FS-CDIS với hàm mất mát ba thành phần ở cấp độ thực thể (-ITL) và bộ nhớ lưu trữ thực thể (-IMS). Đề xuất của chúng tôi cải thiện độ chính xác trên các mô hình cơ sở (các dòng tô màu xám). . . . .	67
4.4	Độ chính xác của hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể của chúng tôi trên mô hình MTFA [77]. Kết quả tốt nhất được <b>in đậm</b> . # kí hiệu số lượng mẫu, “Memory” là Instance Memory Storage and “Triplet” là Instance Triplet Loss. . . . .	69
4.5	Thực nghiệm loại suy trên các mô hình cơ sở với 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu <b>đỏ</b> , và <i>xanh</i> . “Memory” là Instance Memory Storage và “Triplet” là Instance Triplet Loss. .	70

- 4.6 Thực nghiệm loại suy trên trọng số  $\alpha$  và tham số *margin* của hàm mất mát ba thành phần ở cấp độ thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*. . . . . 71
- 4.7 Thực nghiệm loại suy trên tham số sức chứa *capacity* của bộ nhớ lưu trữ thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*. . . . . 72
- 4.8 Kết quả thực nghiệm loại suy trên trọng số  $\beta$  của bộ nhớ lưu trữ thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*. . . . . 73

## Chương 1

# TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU

### 1.1 Giới thiệu đề tài

**Bối cảnh thực tiễn.** Bài toán phân đoạn ở cấp độ thực thể trong lĩnh vực thị giác máy tính có những ứng dụng tiềm năng trong thực tế nhờ vào khả năng đọc hiểu hình ảnh ở mức độ chi tiết cao, cụ thể là ở cấp độ thực thể. Trong luận văn này, chúng tôi giải quyết bài toán phân đoạn thực thể trên các đối tượng đặc thù có yếu tố nguy trang. Với mục tiêu giải quyết bài toán trên đối tượng có yếu tố nguy trang đặc thù, chúng tôi hướng đến các ứng dụng tiềm năng khi có thể phát hiện tự động các đối tượng có tính chất nguy trang, có thể kể đến như các nhiệm vụ tìm kiếm và giải cứu con người, động vật trong tự nhiên hay để bảo tồn các loài động, thực vật [37, 38].

Cụ thể hơn, chúng tôi trình bày một số ngữ cảnh ứng dụng điển hình của bài toán phân đoạn thực thể nguy trang với những ví dụ thực tế sau đây:

**Tìm kiếm và bảo tồn các loài động vật quý hiếm.** Kết quả của bài toán này có thể hỗ trợ đắc lực trong công tác tìm kiếm và bảo tồn những loài động vật quý hiếm trong điều kiện tự nhiên phức tạp. **Hình 1.1** trình bày ảnh chụp thực tế của 2 cá thể động vật quý hiếm. **Hình 1.1-a** là ảnh chụp một cá thể mang<sup>1</sup> lớn được nhìn thấy ở Công viên Quốc

<sup>1</sup>Mang, còn gọi là hoẵng, ki, mển hay mển, là một dạng hươu, nai cổ nhất được biết đến.

gia Virachey của tỉnh Ratanakirri, Campuchia năm 2021<sup>2</sup>. Hình 1.1-b là ảnh chụp một cá thể hổ hoang dã trưởng thành được ghi nhận năm 1999 tại Vườn Quốc gia Pù Mát, Việt Nam<sup>3</sup>. Có thể nhận ra từ ảnh chụp tự nhiên của 2 cá thể động vật này, chúng không có yếu tố màu sắc nổi bật hay các đặc trưng tách biệt so với môi trường xung quanh. Có thể nói đây là bản năng sinh tồn của chúng để tránh kẻ thù hoặc để ngụy trang khi săn mồi. Đặc trưng này không chỉ tồn tại ở 2 loài động vật kể trên, chúng ta có thể dễ dàng tìm ra các loài động vật khác cũng có cùng cơ chế sinh tồn như vậy trong tự nhiên. Trong các tình huống cần tìm kiếm, giải cứu hay phát hiện các loài động vật này với mục tiêu để cứu hộ hay bảo tồn, kết quả của bài toán phân đoạn cung cấp khả năng tự động hóa quá trình tìm kiếm, giúp giải phóng phần nào nhân lực.



(a) Một cá thể mang lớn được nhìn thấy ở Công viên Quốc gia Virachey của tỉnh Ratanakirri, Campuchia năm 2021.

Nguồn tin: vodenglish.news

(b) Một cá thể hổ hoang dã trưởng thành được ghi nhận năm 1999 tại Vườn Quốc gia Pù Mát, Việt Nam.

Nguồn tin: vietnamnet.vn

HÌNH 1.1: Ảnh chụp một số cá thể động vật quý hiếm trong tự nhiên. Các loài động vật này mang những màu sắc, đường nét tương đồng với môi trường xung quanh hoặc có tập tính ẩn mình để săn mồi. Các yếu tố này khiến chúng trở nên khó bị phát hiện.

**Phục vụ công tác tìm kiếm và giải cứu.** Một ví dụ khác với quy mô công tác tìm kiếm và giải cứu lớn hơn được ứng dụng trong các chiến dịch tìm kiếm cứu nạn con người khi xảy ra sự cố hoặc thiên tai. Thông thường, các chiến dịch tìm kiếm cứu nạn được tổ chức ở quy mô lớn đòi hỏi một lượng lớn về nhân lực và vật lực, tiêu tốn chi phí đáng kể. Liệu chúng ta có thể giảm bớt chi phí này nhờ vào sự hỗ trợ của máy móc? Một trong những hướng giải quyết tiềm năng là chúng ta có thể sử dụng thiết bị bay không người lái (*drone*) để thu thập hình ảnh tại khu vực được khoanh vùng tìm kiếm. Các hình ảnh

<sup>2</sup><https://vodenglish.news/first-confirmed-sighting-in-cambodia-of-endangered-deer-official>, truy cập vào tháng 09/2023.

<sup>3</sup><https://vietnamnet.vn/chuyen-ve-buc-anh-duy-nhat-chup-ca-the-ho-rung-o-viet-nam-809941.html>, truy cập vào tháng 09/2023





HÌNH 1.2: Minh họa ý tưởng phát hiện các đối tượng hay thực thể có tính chất nguy trạng có thể áp dụng tự động trong các chiến dịch tìm kiếm và giải cứu con người hay các loài động vật khi sử dụng hình ảnh từ thiết bị bay không người lái hoặc hình ảnh từ các camera giám sát.

thu được sẽ được mô hình học máy xử lý và đưa ra dự đoán các vị trí có tồn tại đối tượng được quan tâm, trong trường hợp này có thể là đồ vật, động vật, hoặc là con người. Hình 1.2 minh họa cho ý tưởng nói trên. Thiết bị bay không người lái có khả năng thay thế con người hoạt động trong các môi trường phức tạp, nguy hiểm như núi đá, rừng rậm, hay sa mạc... Hình ảnh thu được cho phép con người nhận biết tình hình tại các vị trí cần triển khai nhiệm vụ tìm kiếm cứu nạn và giảm thiểu các rủi ro có thể xảy ra với đội cứu hộ.

Bên cạnh các ngữ cảnh ứng dụng thực tế nói trên, bài toán phân đoạn thực thể nguy trạng còn có nhiều ứng dụng trong các lĩnh vực khác, có thể kể đến như lĩnh vực truyền thông, giải trí, y khoa hay quân sự. Kể ra một số ví dụ thực tiễn trên đây, chúng tôi mong muốn phần nào chứng minh tiềm năng ứng dụng của các kết quả do bài toán phân đoạn thực thể nguy trạng mang lại, thúc đẩy sự phát triển của các nghiên cứu về thực thể nguy trạng trong cộng đồng khoa học.

## 1.2 Định nghĩa bài toán

**Phát biểu bài toán.** Trong luận văn này, chúng tôi giải quyết bài toán phân đoạn ngữ nghĩa ở cấp độ thực thể trên các ảnh có chứa các thực thể đặc thù mang yếu tố nguy trang. Bài toán được định nghĩa với **đầu vào** là ảnh có chứa các thực thể nguy trang và **đầu ra** là bản đồ phân đoạn ngữ nghĩa theo từng thực thể nguy trang riêng biệt có trong ảnh. **Hình 1.3** sau đây mô tả một cách trực quan đầu vào và đầu ra của bài toán. Với mỗi ảnh đầu vào có chứa thực thể nguy trang, mô hình phân đoạn thực thể cần trả về bản đồ phân đoạn ngữ nghĩa của mỗi thực thể nguy trang trong ảnh (*mặt nạ phân đoạn màu xanh và màu đỏ*).

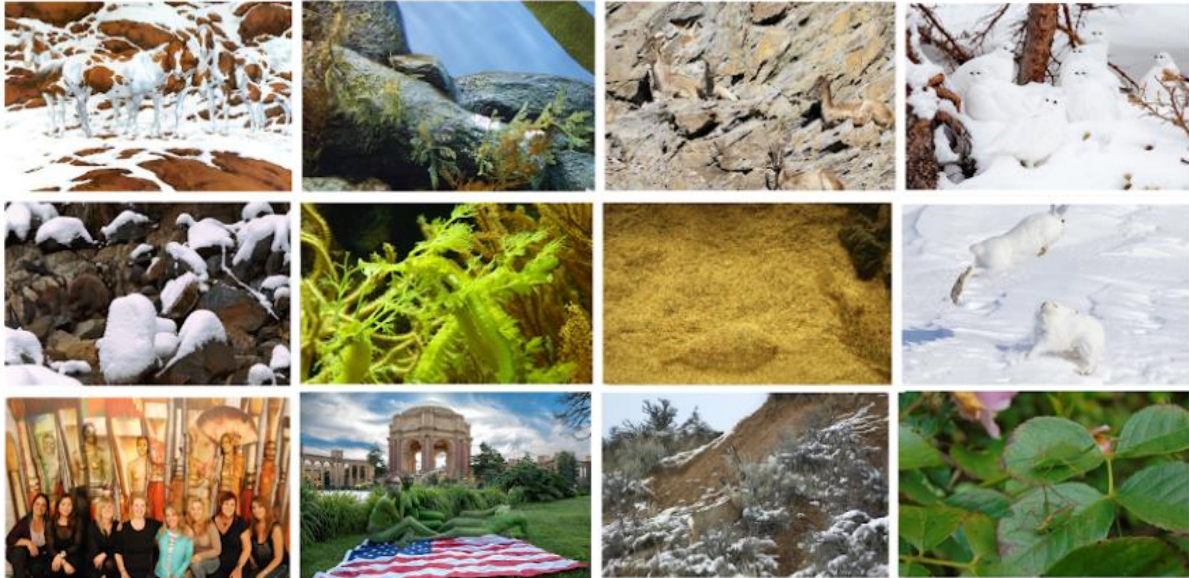


HÌNH 1.3: Minh họa đầu vào và đầu ra của bài toán phân đoạn thực thể nguy trang. Đầu vào là ảnh chứa các thực thể nguy trang, đầu ra là mặt nạ phân đoạn ngữ nghĩa của từng thực thể nguy trang.

Trong luận văn này, chúng tôi tiếp cận giải quyết bài toán phân đoạn thực thể nguy trang bằng phương pháp **khai thác và tận dụng các đặc trưng có tính phân biệt cao** của các thực thể nguy trang trong ảnh. Nguy trang là một cơ chế tự vệ của các loài động vật trong tự nhiên để chúng ẩn mình vào môi trường xung quanh, từ đó tránh được kẻ thù nguy hiểm. Bài toán phân đoạn đối tượng nguy trang ở cấp độ thực thể là một chủ đề nghiên cứu mới và thách thức bởi mức độ chi tiết ngay cả mắt người cũng khó nhận biết được. Với đối tượng có tính chất nguy trang đặc thù, việc phân biệt đặc trưng nguy trang của thực thể và vùng nền sẽ giúp các mô hình học máy có khả năng thực hiện tốt hơn tác vụ của chúng. Các đặc trưng đặc thù này được xem là đặc trưng có tính phân biệt cao vì chúng giúp mô hình đưa ra quyết định chính xác hơn. Trong ngữ cảnh thực tế, các



thực thể nguy trang thường lẫn trốn kẻ thù hay nguy trang vào môi trường tự nhiên, dẫn đến việc thu thập dữ liệu có giới hạn. Vì thế, việc nhận định và khai thác các đặc trưng có tính phân biệt cao của các thực thể này so với môi trường sẽ giúp mô hình sử dụng dữ liệu hiệu quả hơn trong quá trình huấn luyện và đưa ra dự đoán chính xác hơn.



HÌNH 1.4: Một số hình ảnh minh họa cho dữ liệu ảnh thực thể nguy trang được trích từ tập dữ liệu CAMO++ [36].

Kế thừa từ các công trình trước đây nghiên cứu các bài toán trên đối tượng nguy trang [12, 36, 38, 49, 54, 86, 88] cũng như các tập dữ liệu đặc thù phục vụ bài toán này [12, 35, 36, 38, 49, 60, 69], các thực thể nguy trang được xác định gồm có các loài động vật, bao gồm cả con người, có khả năng nguy trang để làm cho bản thân khó bị phát hiện bằng cách hòa lẫn với màu sắc, chất liệu của môi trường xung quanh. Ảnh chứa các thực thể nguy trang hay ảnh nguy trang là ảnh có chứa các động vật, hay con người mà ở đó các chủ thể này có màu sắc, chất liệu tương đồng với vùng hậu cảnh. Bằng mắt thường, con người khi nhìn vào các ảnh nguy trang này sẽ khó phân biệt được đâu là các thực thể nguy trang, đâu là vùng nền. Hình 1.4 cung cấp một số hình ảnh minh họa cho các ảnh chụp có chứa thực thể nguy trang được trích từ tập dữ liệu CAMO++ [36]. Liệu chúng ta có dễ dàng tìm ra thực thể nguy trang trong những bức ảnh này? Nếu bạn gặp khó khăn trong việc nhận diện các thực thể nguy trang, chúng tôi đã chuẩn bị câu trả lời cho câu hỏi này tại Hình 2.19 với các nhãn thực thể phân đoạn.

**Các thách thức.** Lĩnh vực nghiên cứu về đối tượng nguy trang nói chung và bài toán phân đoạn thực thể nguy trang mà luận văn này hướng đến nói riêng có những thách thức đặc thù. Các khó khăn này có thể chia thành 2 nhóm chính gồm có thách thức liên quan đến dữ liệu và thách thức liên quan đến phương pháp giải quyết bài toán:

- **Quy mô các tập dữ liệu nguy trang gán nhãn còn hạn chế.** Về mặt tự nhiên, các động vật nguy trang có tập tính nguy trang để lẩn trốn kẻ thù. Chúng ẩn mình vào môi trường xung quanh và gây khó khăn cho việc phát hiện bằng mắt thường. Vì thế, việc thu thập dữ liệu trong thực tế tốn thời gian và công sức. Hơn nữa, với bài toán phân đoạn thực thể, các hình ảnh này cần được gán nhãn phân đoạn với yêu cầu chính xác đến cấp độ điểm ảnh. Việc gán nhãn này tiêu tốn nhiều thời gian và nhân lực, vì thế, các tập dữ liệu sơ khởi cho bài toán này thường gặp giới hạn về nhãn phân đoạn. Cũng vì lẽ đó, số lượng tập dữ liệu chuẩn với kích thước lớn và chất lượng cao không nhiều. Chúng tôi đã thực hiện khảo sát về vấn đề này và trình bày chi tiết trong phần Các công trình liên quan.
- **Chưa có nhiều mô hình phân đoạn thực thể nguy trang.** Các công trình trước đây về phân đoạn thực thể thường được thực hiện trên miền dữ liệu tổng quát với các tập dữ liệu lớn như MS-COCO [43] hay ImageNet [10]. Tuy các mô hình này đạt kết quả cao trên dữ liệu tổng quát, chúng gặp khó khăn khi áp dụng trực tiếp lên dữ liệu nguy trang đặc thù mà chúng tôi hướng đến. Các mô hình đòi hỏi số lượng dữ liệu huấn luyện lớn, điều mà dữ liệu về đối tượng nguy trang chưa đáp ứng tốt. Đồng thời, các đề xuất về kiến trúc mô hình cũng chưa khai thác được tốt yếu tố đặc thù của thực thể nguy trang để giúp phân biệt chúng với vùng nền một cách hiệu quả.

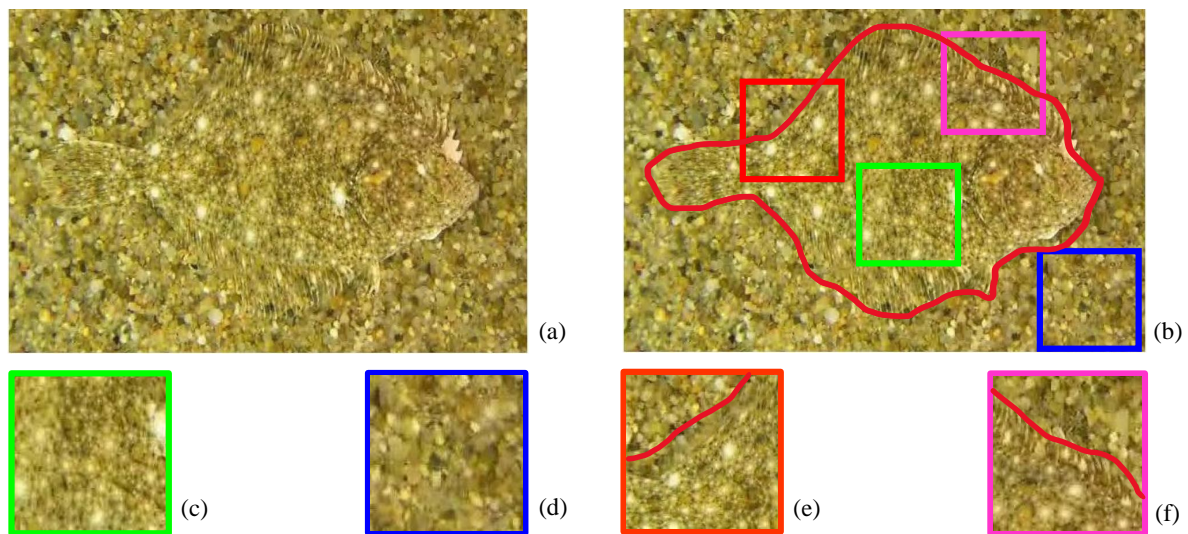
**Lý do thực hiện đề tài.** Bài toán phân đoạn đối tượng nguy trang ở cấp độ thực thể là một chủ đề nghiên cứu đầy thách thức và còn nhiều khía cạnh chưa được khai thác trong lĩnh vực thị giác máy tính. Thêm vào đó, cộng đồng nghiên cứu chưa tập trung giải quyết bài toán phân đoạn các đối tượng nguy trang này ở cấp độ thực thể và việc khai thác các đặc trưng có tính phân biệt cao giữa các thực thể nguy trang và vùng nền. Việc đầu tư nghiên cứu bài toán này sẽ góp phần vào sự phát triển của cộng đồng khoa học

nói chung, và hướng nghiên cứu đặc thù này nói riêng. Hơn nữa, việc phát hiện tự động các đối tượng có tính chất nguy trang có nhiều ứng dụng thiết thực, có thể kể đến như các nhiệm vụ tìm kiếm và giải cứu con người và động vật trong tự nhiên hay bảo tồn động, thực vật đã trình bày ở phần trên.

### 1.3 Mục tiêu của luận văn

Bài toán phân đoạn thực thể nguy trang là một bài toán khó và còn nhiều thách thức. Trong luận văn này, khi giải quyết bài toán phân đoạn thực thể nguy trang, chúng tôi hướng đến 2 mục tiêu chính sau đây:

#### 1.3.1 Tăng cường đặc trưng phân biệt ở vùng biên cạnh để phân đoạn hiệu quả thực thể nguy trang

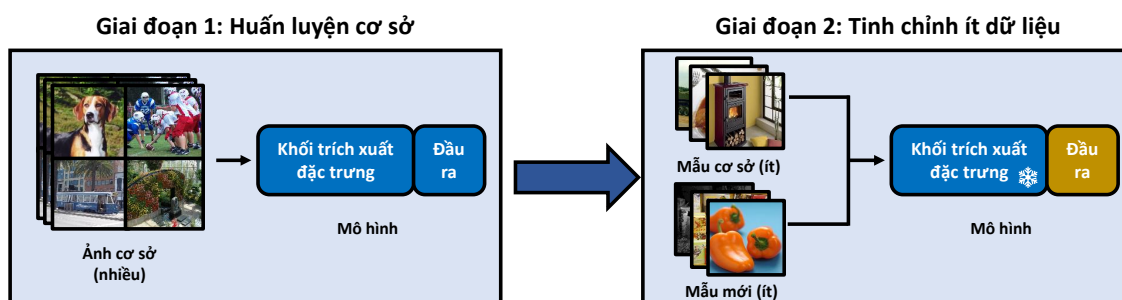


HÌNH 1.5: So sánh sự tương đồng về mặt thị giác của các vùng ảnh trong ảnh có chứa thực thể nguy trang. Thứ tự gán nhãn: (a) ảnh gốc, (b) ảnh với đường biên cạnh, (c) ô cắt vùng thực thể, (d) ô cắt vùng nền, (e, f) ô cắt vùng biên vật thể và vùng nền.

Đặc trưng của các thực thể nguy trang, đơn cử là các loài động vật nguy trang, là tương thích màu sắc, hình dạng, chất liệu của vỏ bề ngoài của chúng cho phù hợp với môi trường xung quanh. Bằng cách này, chúng có thể đánh lừa kẻ thù để trở nên "vô hình" và hòa lẫn vào cảnh vật. Tuy nhiên, khả năng ẩn mình và hòa lẫn với môi trường

tồn tại những nhược điểm nhất định. Các mảng màu, hình dạng của thực thể nguy trang được chuẩn bị tốt ở các khu vực trung tâm hay các vùng lớn trên cơ thể. Nhưng khi chúng ta nhìn từ một góc độ nào đó, các khu vực tiếp giáp giữa thực thể này và môi trường sẽ có những yếu tố chưa tương đồng. Hình 1.5 minh họa giúp hình dung rõ nét hơn về những yếu tố chưa tương đồng của các thực thể nguy trang này và vùng nền. Tại các khu vực tiếp giáp, hay còn gọi là biên cạnh, các đặc trưng nguy trang dường như mất đi tính liên tục của chúng. Các hoa văn, màu sắc dù cho được nguy trang tốt, vẫn thể hiện sự đứt đoạn, không thống nhất giữa thực thể và môi trường xung quanh. Về mặt định nghĩa, một điểm ảnh được coi là thuộc biên cạnh của vật thể nếu tại đó diễn ra sự thay đổi đột ngột về màu sắc, hoặc cường độ. Dựa trên quan sát này, chúng tôi nhận định rằng, **việc khai thác và tận dụng tốt các đặc trưng biên cạnh có khả năng cải thiện độ chính xác phân đoạn** của các mô hình học sâu tiến tiến.

### 1.3.2 Khai thác đặc trưng phân biệt với ít dữ liệu huấn luyện để phân đoạn thực thể nguy trang



HÌNH 1.6: Ý tưởng tổng quát của mô hình học dựa trên ít mẫu dữ liệu huấn luyện. Quá trình học gồm 2 giai đoạn: giai đoạn cơ sở cung cấp tri thức tổng quát cho mô hình và giai đoạn tinh chỉnh giúp mô hình học dựa trên ít mẫu dữ liệu huấn luyện gắn nhãn.

Các thực thể động vật nguy trang thường có xu hướng ẩn mình để lẩn trốn kẻ thù nguy hiểm, khiến cho việc nhận ra chúng trong tự nhiên trở thành một nhiệm vụ khó khăn. Hình 1.4 minh họa một số hình ảnh về các động vật nguy trang trong tự nhiên gây thách thức trong việc nhận diện bằng mắt thường. Điều này làm cho quá trình thu thập dữ liệu thực tế về các loài này cũng gặp trở ngại. Hơn nữa, bài toán mà chúng tôi hướng



đến là phân đoạn ở cấp độ thực thể. Vì thế, việc gán nhãn dữ liệu phân đoạn trên các hình ảnh này đòi hỏi sự cẩn thận và chi tiết cao, tiêu tốn nhiều công sức. Chính vì thế, chúng tôi đặt ra mục tiêu làm sao để có thể **sử dụng ít dữ liệu huấn luyện gán nhãn** về các thực thể nguy trang này mà vẫn đảm bảo hoàn thành được tác vụ phân đoạn thực thể. Mục tiêu này hướng đến việc giảm thiểu chi phí gán nhãn đồng thời sử dụng các kiến trúc được thiết kế cho điều kiện ít dữ liệu huấn luyện. Hơn nữa, việc sử dụng ít mẫu dữ liệu huấn luyện đòi hỏi khả năng khai thác hiệu quả các thông tin đặc trưng nguy trang của mô hình trong các mẫu dữ liệu huấn luyện có sẵn. **Hình 1.6** minh họa ý tưởng tổng quát của mô hình học dựa trên ít mẫu dữ liệu huấn luyện được chúng tôi áp dụng trong mục tiêu này. Với cơ chế học ít dữ liệu, mô hình được huấn luyện trước hết với giai đoạn cơ sở nhằm cung cấp tri thức tổng quan, và sau đó học trên ít dữ liệu gán nhãn ở giai đoạn tinh chỉnh nhằm đưa ra dự đoán trên các lớp ngữ nghĩa mục tiêu.

## 1.4 Đóng góp chính của luận văn

Với hai mục tiêu chính nêu trên, chúng tôi đề xuất phương pháp để cải thiện độ chính xác của các mô hình học sâu tiên tiến trong tác vụ phân đoạn thực thể nguy trang với 03 đóng góp về mặt khoa học như sau:

- Một là, nghiên cứu, đề xuất mô hình CE-OST [CT1] dựa trên kiến trúc Transformer với khối tăng cường đặc trưng biên cạnh (contour emphasis) để phân đoạn hiệu quả các thực thể nguy trang.
- Hai là, nghiên cứu, đề xuất mô hình FS-CDIS [CT2, CT3] sử dụng ít dữ liệu huấn luyện để giải quyết bài toán phân đoạn thực thể nguy trang và khai thác đặc trưng phân biệt dựa trên kỹ thuật học tương phản (contrastive learning) với hàm mất mát ba thành phần ở cấp độ thực thể (instance triplet loss) và sử dụng bộ nhớ lưu trữ thực thể (instance memory storage).
- Ba là, đề xuất tập dữ liệu ảnh CAMO-FS [CT2, CT3] cho bài toán phát hiện và phân đoạn thực thể nguy trang, được tinh chỉnh và tạo lập cấu trúc cho hướng tiếp cận học ít dữ liệu.

## **1.5** **Bố cục luận văn**

Luận văn này gồm 5 chương với bố cục như sau:

- Chương 1: Giới thiệu. Chương này trình bày tổng quan đề tài, các thách thức, lý do thực hiện đề tài cũng như mục tiêu và các đóng góp khoa học của luận văn.
- Chương 2: Các công trình liên quan. Chương này trình bày các kiến thức nền tảng và các nghiên cứu về thực thể nguy trang có liên quan đến đề tài luận văn như tổng quan về nghiên cứu trên đối tượng nguy trang, các mô hình phân đoạn, và các tập dữ liệu chuẩn phục vụ nghiên cứu.
- Chương 3: Phân đoạn thực thể nguy trang với mô hình CE-OST khai thác đặc trưng biên cạnh. Chương này trình bày chi tiết các đóng góp của luận văn với mô hình CE-OST, thực nghiệm và các cải tiến trong việc khai thác và tận dụng đặc trưng biên cạnh để phân đoạn hiệu quả thực thể nguy trang.
- Chương 4: Phân đoạn thực thể nguy trang sử dụng ít dữ liệu huấn luyện với mô hình FS-CDIS. Chương này trình bày chi tiết các đóng góp của luận văn với mô hình FS-CDIS, thực nghiệm và các cải tiến trong việc phân đoạn thực thể nguy trang với ngữ cảnh ít dữ liệu huấn luyện và khai thác đặc trưng phân biệt dựa trên kỹ thuật học tương phản.
- Chương 5: Kết luận. Chương này tóm tắt nội dung luận văn và đề cập đến hướng phát triển đề tài.

## Chương 2

# CÔNG TRÌNH LIÊN QUAN

Trong chương này, chúng tôi trình bày tóm lược các nghiên cứu về thực thể ngẫu trang có liên quan đến luận văn này. Các nghiên cứu này gồm có các công trình về phân đoạn thực thể ngẫu trang với các hướng tiếp cận một giai đoạn, hai giai đoạn và sử dụng ít dữ liệu huấn luyện. Chúng tôi cũng trình bày các hướng tiếp cận giúp khai thác đặc trưng có tính phân biệt cao như sử dụng đặc trưng biên cạnh hay các phương pháp học tương phản. Cuối cùng, chúng tôi đề cập đến các tập dữ liệu đặc thù cho nghiên cứu trên thực thể ngẫu trang.

### 2.1 Tổng quan nghiên cứu về thực thể ngẫu trang

**Bối cảnh thực tiễn.** Trong lĩnh vực thị giác máy tính, các nghiên cứu về thực thể ngẫu trang được giới hạn trong các tác vụ thực hiện trên dữ liệu ảnh và video có chứa thực thể ngẫu trang. Chúng ta có các bài toán như phân loại, phát hiện đối tượng, phân đoạn ngữ nghĩa, hay phân đoạn thực thể ngẫu trang, theo thứ tự tăng dần về độ khó và về mức độ chi tiết mà mô hình học máy hiểu về thực thể ngẫu trang đó. Trong phần này, chúng tôi khảo sát các công trình liên quan đến bài toán phân đoạn đối tượng ngẫu trang ở cấp độ thực thể (hay phân đoạn thực thể ngẫu trang). Đây là tác vụ có mức độ chi tiết cao, đòi hỏi mô hình học máy có khả năng nhận biết và tìm ra vị trí chính xác của các điểm ảnh thuộc về thực thể ngẫu trang trong ảnh đầu vào.

Trước hết, chúng tôi nhắc lại định nghĩa về thực thể ngẫu trang. Cho trước một bức ảnh, khi xác định các vùng quan tâm (như khung bao - *bounding box*, hay mặt nạ ngữ

nghĩa - *polygon masks*) đại diện cho một đối tượng hay thực thể được quan tâm trong ảnh mà các đối tượng này có xu hướng bị nhầm lẫn là vùng nền thì các đối tượng này được xem là đối tượng hay thực thể nguy trang. Theo đó, đối tượng hay thực thể nguy trang được định nghĩa là một tập các khung bao hay một tập các điểm ảnh biểu diễn thực thể nguy trang [38]. Mặc dù các nghiên cứu trên thực thể nguy trang có nhiều ứng dụng trong thực tiễn, hướng nghiên cứu này vẫn chưa được khai phá triệt để, đặc biệt là hướng nghiên cứu khai thác các đặc trưng có tính phân biệt cao của thực thể nguy trang, hay ứng dụng trong ngữ cảnh ít dữ liệu của thực thể nguy trang.

**Thực trạng nghiên cứu về đối tượng nguy trang.** Cũng như phần lớn các bài toán trong thị giác máy tính, bài toán có hai hướng tiếp cận chính là sử dụng đặc trưng cấp thấp và sử dụng đặc trưng học sâu. Các hướng tiếp cận trước đây chủ yếu khai thác đặc trưng cấp thấp như màu sắc, biên cạnh, chất liệu, hay độ sáng [40, 61] để thực hiện các tác vụ trên thực thể nguy trang. Nhiều năm gần đây, cùng với sự bùng nổ của các mạng học sâu, các tác vụ như phân loại, phát hiện hay phân đoạn thực thể nguy trang đã đạt được nhiều thành tựu đáng kể. Zhai và cộng sự [86] tận dụng kỹ thuật học dựa trên đồ thị để huấn luyện mô hình phát hiện được biên cạnh và vùng chứa đối tượng nguy trang. Sau đó, PFNet [54] được đề xuất phát hiện động vật nguy trang dựa trên mô phỏng khả năng săn mồi trong tự nhiên của các loài động vật ăn thịt. Năm 2019, Le và cộng sự [38] giới thiệu Anabranh, một mô hình kết hợp bài toán phân loại và phân đoạn trên đối tượng nguy trang. Hướng tiếp cận này có khả năng tương thích với các kiến trúc mạng tích chập đầy đủ (fully convolution network). Vào năm 2020, SINet [12] ra đời với mục tiêu bắt chước hành vi săn mồi của các loài động vật để xác định vị trí và nhận biết đối tượng nguy trang trong tự nhiên. Lyu và cộng sự [49] thiết kế một kiến trúc mạng có khả năng xếp hạng dự đoán các đối tượng nguy trang trong khi vẫn có khả năng xác định vị trí và phân đoạn chúng để tăng cường độ chính xác dự đoán. Cùng thời gian đó, TINet [88] khai thác yếu tố tương tác để tinh chỉnh các đặc trưng có liên quan đến chất liệu và phân đoạn thực thể nguy trang ở đa cấp độ đặc trưng. Le và cộng sự [36] tiếp tục nghiên cứu về thực thể nguy trang với hướng tiếp cận kết hợp đa mô hình để cải thiện khả năng nắm bắt ngữ cảnh, từ đó hỗ trợ phát hiện thực thể nguy trang tốt hơn.



## 2.2 Các kiến trúc phân đoạn thực thể nguy trang

Chúng tôi tiếp tục trình bày các kiến trúc có liên quan để phục vụ hiểu biết về bài toán phân đoạn thực thể nguy trang với 3 hướng tiếp cận là sử dụng mô hình hai giai đoạn, mô hình một giai đoạn, và hướng tiếp cận sử dụng ít dữ liệu huấn luyện. Trong đó, hướng tiếp cận một và hai giai đoạn cùng thể hiện khía cạnh kiến trúc thiết kế của các mô hình, còn hướng tiếp cận sử dụng ít dữ liệu huấn luyện là một mô hình bài toán hoàn toàn khác, ở đó tập trung khai thác thông tin từ số lượng ít mẫu dữ liệu cho trước để mô hình học hiệu quả. Các phần dưới đây trình bày những nét chính về các mô hình được đề cập, giúp người đọc nắm bắt các thông tin quan trọng, phục vụ việc so sánh các mô hình này với mô hình đề xuất của chúng tôi.

### 2.2.1 Phân đoạn thực thể với kiến trúc hai giai đoạn

Với hướng tiếp cận hai giai đoạn, chúng ta có thể kể đến các công trình sử dụng một quy trình truyền thống gồm hai bước phát hiện và phân đoạn để khởi tạo các vùng quan tâm (ROI) với khung bao rồi sau đó sẽ tạo ra mặt nạ phân đoạn ngữ nghĩa theo từng khung bao đó [71]. Các công bố tiêu biểu cho hướng tiếp cận này có thể kể đến như: Mask RCNN [28], Mask Scoring RCNN [30], Cascade Mask RCNN [2], PANet [44], HTC [6], BlendMask [5], Mask Transfuser [33] hay DCNet [48]. Sau đây là những nét chính về các phương pháp hai giai đoạn phục vụ bài toán phân đoạn thực thể mà chúng tôi đã khảo sát.

#### Mô hình Mask RCNN [28]

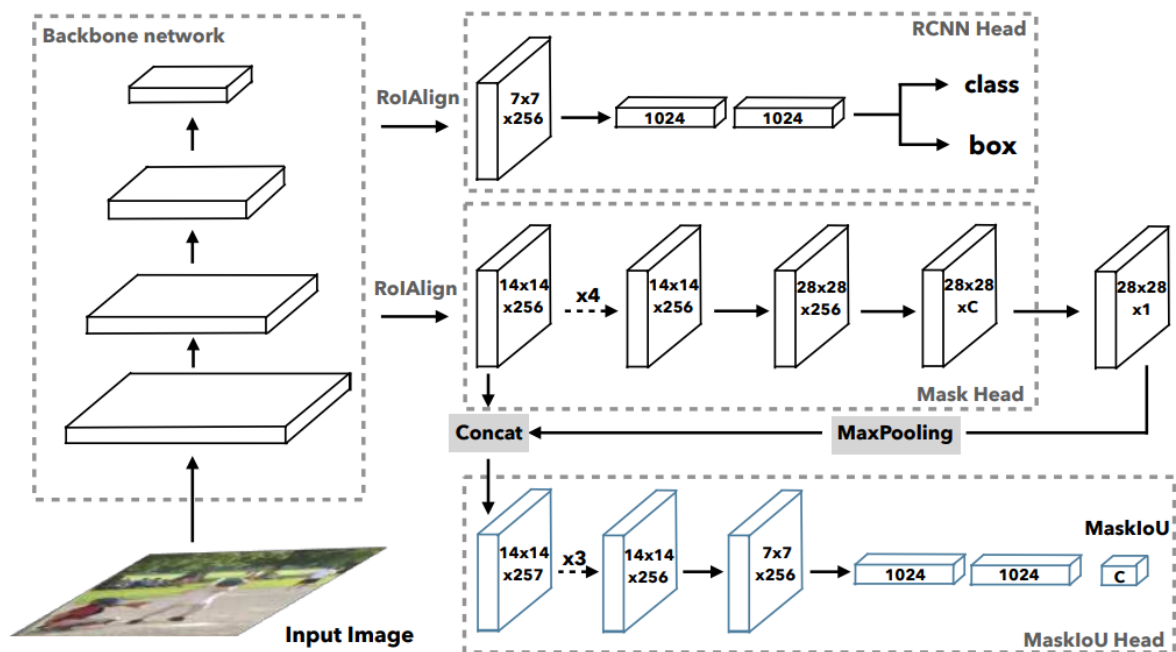
Mô hình kinh điển nhất là Mask RCNN [28], được đề xuất từ những năm 2017, dựa trên nền tảng kiến trúc Faster RCNN [63] cho bài toán phát hiện đối tượng. Mask R-CNN đề xuất việc thêm vào một nhánh dành riêng cho tác vụ phân đoạn ngữ nghĩa ở cấp độ thực thể. Theo đó, ba đầu ra sẽ giải quyết riêng biệt ba tác vụ khác nhau, lần lượt là phân loại, phát hiện đối tượng và phân đoạn thực thể. Quá trình huấn luyện mạng có thể được tiến hành độc lập để huấn luyện hay đóng băng các đầu ra này tùy theo mục đích tác vụ cần tối ưu. Đầu ra phân đoạn thực thể sử dụng các vùng quan tâm đề xuất ROIs từ đầu

ra phát hiện đối tượng để xác định các thực thể cần phân đoạn. Ý tưởng của mô hình này là tiền đề cho nhiều công trình về phân đoạn thực thể sau này.

**Mô hình Mask Scoring RCNN [30]**

Cùng hướng tiếp cận dựa trên Mask RCNN, Mask Scoring RCNN [30] có thêm một nhánh MaskIOU cho tác vụ đánh giá mặt nạ ngữ nghĩa. Cụ thể, nhánh MaskIOU sử dụng đặc trưng thực thể và mặt nạ dự đoán được để tính điểm IoU giữa mặt nạ dự đoán và mặt nạ nhãn. Từ đó, tối ưu được tác vụ phân đoạn thực thể mà Mask R-CNN đang hướng đến.

Hình 2.1 thể hiện trực quan kiến trúc mạng của mô hình với nhánh MaskIOU là điểm cải tiến chính được đề xuất.

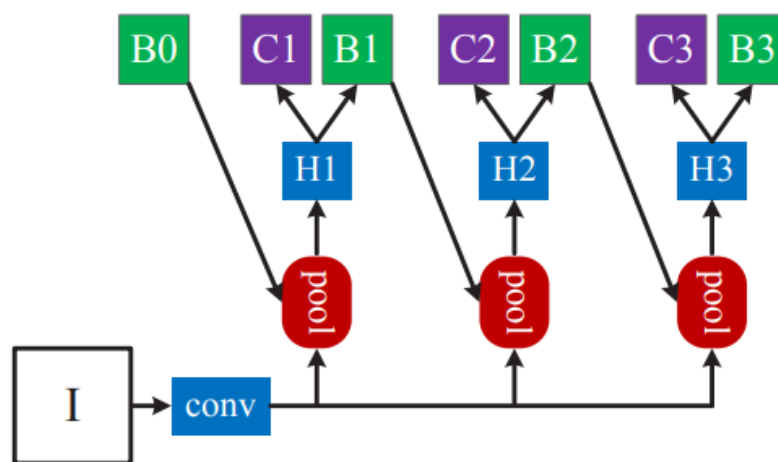


HÌNH 2.1: Kiến trúc mô hình Mask Scoring RCNN [30] với nhánh Mask-IOU là điểm cải tiến chính được đề xuất.

**Mô hình Cascade Mask R-CNN [2]**

Cascade R-CNN là một kiến trúc với nhiều giai đoạn bao gồm một chuỗi các bộ phát hiện đối tượng được huấn luyện với các ngưỡng IOU khác nhau tăng dần để chọn lọc ra các mẫu false positive một cách hiệu quả hơn. Ở phiên bản được nhóm tác giả công bố, mô hình Cascade R-CNN [2] chỉ giải quyết vấn đề phát hiện đối tượng mà thôi. Tuy nhiên, với cùng một cơ chế như đã nhắc đến ở mô hình Mask R-CNN, Cascade R-CNN được thêm vào một nhánh phân đoạn để giải quyết được tác vụ phân đoạn thực thể.

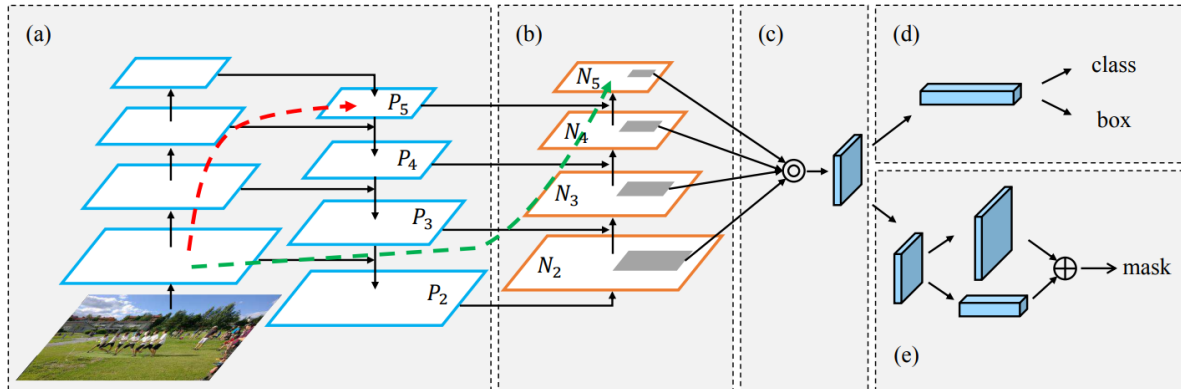
**Hình 2.2** thể hiện kiến trúc của mô hình với các mô-đun nối tiếp theo cơ chế xếp tầng (cascading). Trong trường hợp này, số lượng mô-đun nối tiếp nhau là ba mô-đun, kết quả của mô-đun trước được dùng làm đầu vào cho mô-đun tiếp theo.



HÌNH 2.2: Kiến trúc mô hình Cascade R-CNN [2]. "I" là ảnh đầu vào, "conv" là lớp tích chập rút trích đặc trưng, "pool" là bộ trích xuất đặc trưng theo vùng (region-wise), "H" là đầu ra theo các tác vụ, "B" là kết quả khung bao, "C" là kết quả phân loại, và "B<sub>0</sub>" là các vùng đề xuất khởi tạo của mạng.

### Mô hình PANet [44]

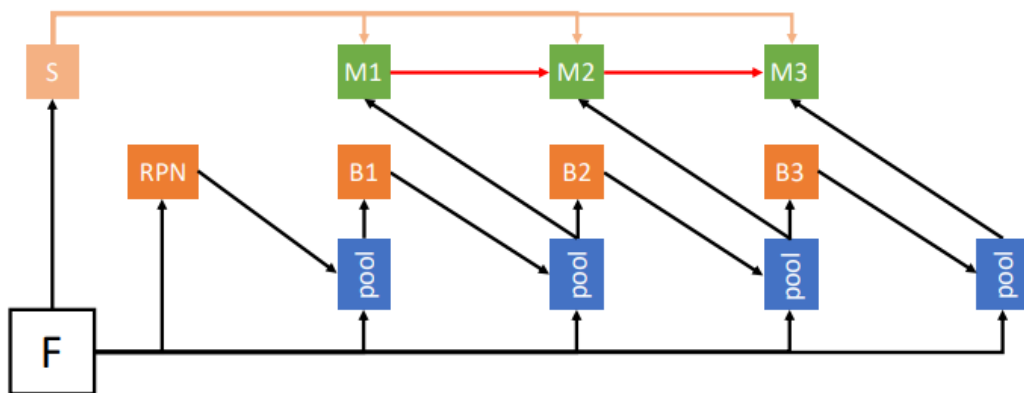
PANet [44] được đề xuất để hạn chế việc mất mát thông tin và tăng cường thông tin cho tác vụ phân đoạn thực thể. Cụ thể, PANet có 3 đóng góp chính: 1) Hạn chế mất mát thông tin và tăng cường đặc trưng đa tầng với thông tin vị trí từ đặc trưng cấp thấp, gọi là mô-đun tăng cường ảnh bottom-up. Cụ thể, PANet khám phá khía cạnh lan truyền ngược của đặc trưng cấp thấp để cải thiện đặc trưng trong quá trình phát hiện đối tượng; 2) Tái tạo thông tin bị mất giữa mỗi vùng đề xuất và đặc trưng tại tất cả cấp độ thông qua lấy mẫu đặc trưng tương thích (adaptive feature pooling). Khối này dùng để kết hợp đặc trưng từ nhiều cấp độ khác nhau cho mỗi đề xuất, giảm nhiễu cho kết quả đầu ra; 3) Nắm bắt thông tin với nhiều góc nhìn khác nhau bằng cách tăng cường cho dự đoán nhãn mặt nạ với mạng kết nối đầy đủ đơn giản (FC). Tăng cường đặc trưng giúp tăng tính đa dạng của thông tin tổng hợp được, giúp mặt nạ phân đoạn có chất lượng tốt hơn. **Hình 2.3** minh họa các thành phần trong mô hình PANet với 3 đóng góp chính nói trên. Trong kiến trúc mô hình PANet, tham số được chia sẻ ở cả hai đầu ra phát hiện đối tượng và phân đoạn thực thể, do đó, độ chính xác của hai tác vụ này được cải thiện đồng thời.



HÌNH 2.3: Kiến trúc mô hình PANet [44] hạn chế việc mất mát thông tin và tăng cường thông tin cho tác vụ phân đoạn thực thể.

### Mô hình HTC [6]

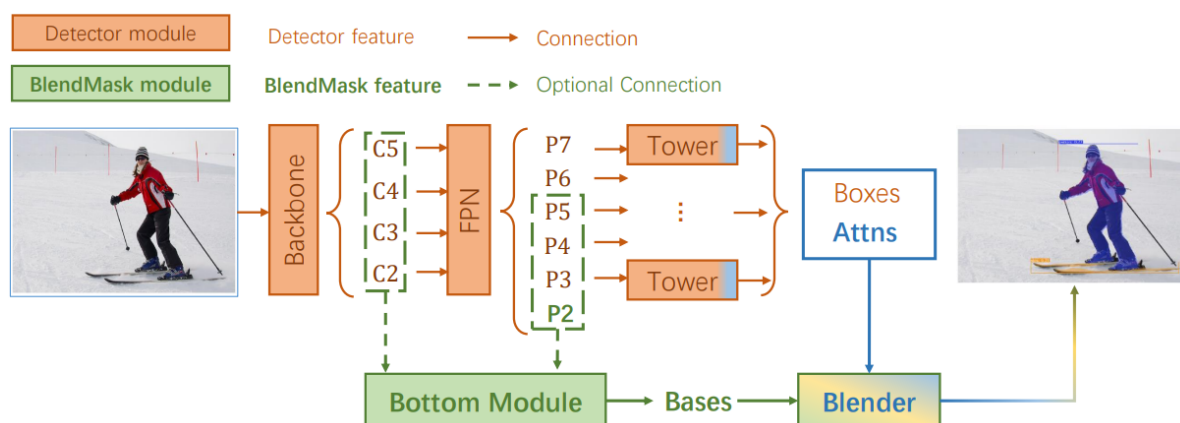
Chen và cộng sự [6] đã công bố mô hình HTC với mục tiêu kết hợp các đặc trưng phân đoạn và đặc trưng phát hiện đối tượng để thực thi tác vụ phân đoạn thực thể. Ý tưởng chính của mô hình là cải thiện luồng thông tin bằng cách xếp tầng (cascading) và xử lý đa tác vụ tại mỗi giai đoạn (multi-tasking), đồng thời tận dụng thông tin về không gian để tăng cường độ chính xác phân đoạn. Theo đó, tại mỗi giai đoạn xử lý, cả hai tác vụ dự đoán khung bao và dự đoán mặt nạ phân đoạn cùng được xử lý tuần tự, mô tả trực quan tại Hình 2.4. HTC có hai điểm chính trong mô hình đề xuất: 1) Thay vì thực hiện tuần tự hai tác vụ phát hiện và phân đoạn, mô hình thực hiện chúng đồng thời; 2) HTC sử dụng một nhánh tích chập đầy đủ FCNs [67] để rút trích đặc trưng ngữ cảnh phục vụ việc phân biệt vùng nền và vùng thực thể có xu hướng khó nhận biết.



HÌNH 2.4: Kiến trúc mô hình HTC [6] cải thiện luồng thông tin bằng cách xếp tầng (cascading) và xử lý đa tác vụ tại mỗi giai đoạn (multi-tasking).

### Mô hình BlendMask [5]

BlendMask [5] thêm vào một bước khởi tạo để tạo ra các đặc trưng dày đặc của thực thể mang yếu tố nhạy cảm với vị trí trên mỗi điểm ảnh. Sau đó, mô hình sử dụng một mô-đun kết hợp để gom đặc trưng của từng thực thể tạo thành bản đồ đặc trưng chứa nhãn phân đoạn thực thể cuối cùng. Điểm cải tiến của BlendMask nằm ở khả năng tổng hợp đặc trưng với độ phân giải cao thông qua một mạng FPN để kết hợp đặc trưng tại nhiều kích cỡ khác nhau. Mô-đun kết hợp đặc trưng của BlendMask sẽ kết hợp dựa trên vùng đề xuất và dựa đoán nhãn của thực thể (Bottom module hay Blender), do đó, mô-đun này có khả năng gắn vào các mô hình phát hiện đối tượng khác để giải quyết tác vụ phân đoạn thực thể một cách dễ dàng (Hình 2.5).

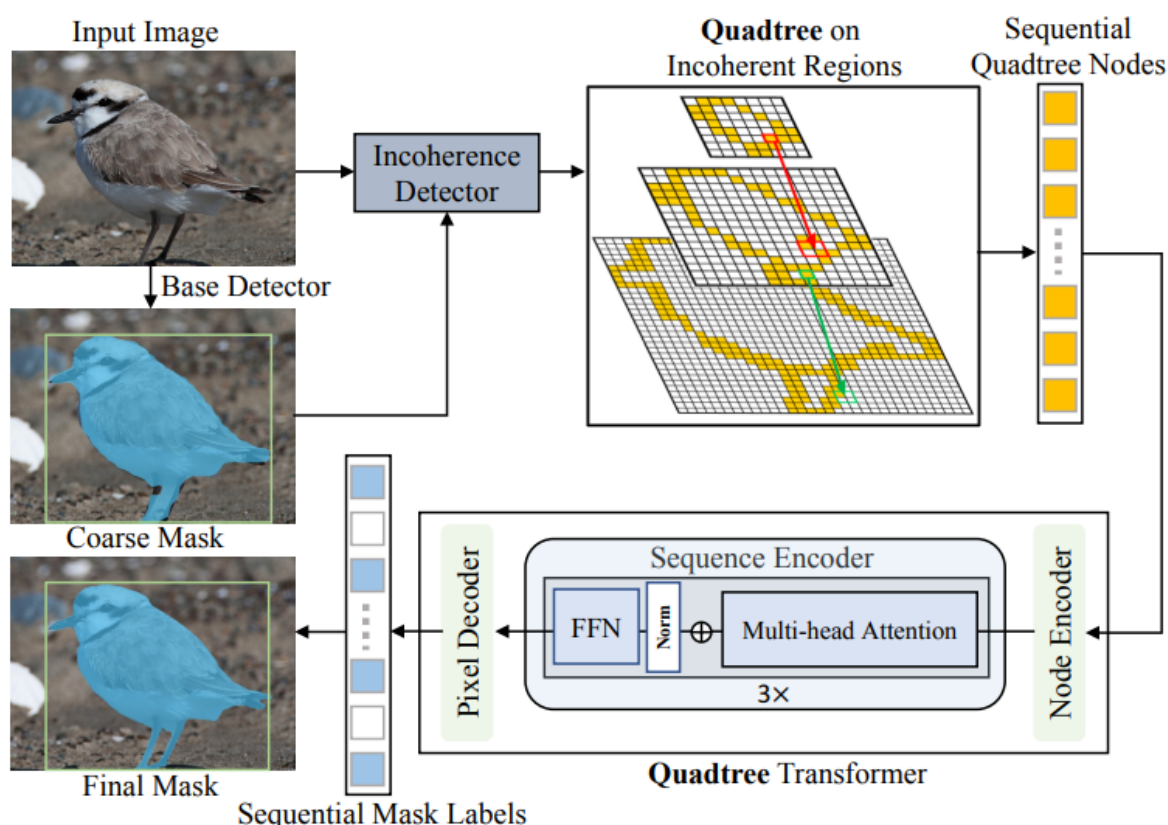


HÌNH 2.5: Kiến trúc mô hình BlendMask [5] với khả năng tổng hợp đặc trưng ở độ phân giải cao thông qua một mạng FPN.

### Mô hình Mask Transfiner [33]

Nhóm tác giả chỉ ra rằng khoảng cách giữa kết quả phân đoạn và kết quả phát hiện đối tượng phụ thuộc vào mức độ chi tiết hay độ phân giải của bản đồ đặc trưng học sâu. Điều này dẫn đến một số mô hình dù có kết quả phát hiện đối tượng tốt, nhưng lại kém hiệu quả khi xử lý tác vụ phân đoạn. Để giải quyết vấn đề đó, Mask Transfiner [33] được đề xuất dựa trên kiến trúc Transformer cho bài toán phân đoạn đối tượng với độ phân giải cao. Trong Hình 2.6, mô hình Transfiner trước hết tìm ra các vùng dễ bị phân đoạn sai, thường xuất hiện ở vùng biên thực thể hoặc vùng có tần số cao. Sau đó, mô hình tập trung học để phát hiện các vùng này thông qua một cấu trúc dạng cây tứ phân (quadtree) thay vì sử dụng tensor thông thường, để xử lý riêng đặc trưng tại các vùng này ở nhiều

mức độ kích thước khác nhau. Kể đến, một mạng tinh chỉnh dựa trên Transformer được sử dụng để mã hóa các node của cây tứ phân, gọi là Quadtree Transformer. Thông qua bộ mã hóa này, mặt nạ phân đoạn đầu ra được tăng cường và cải thiện tại các vị trí được mô hình chú ý nêu trên. Đồng thời, việc học tại các vùng cục bộ này cũng giảm thiểu chi phí tính toán, nhưng đem lại hiệu quả cao vì giải quyết được điểm yếu của kết quả đầu ra.

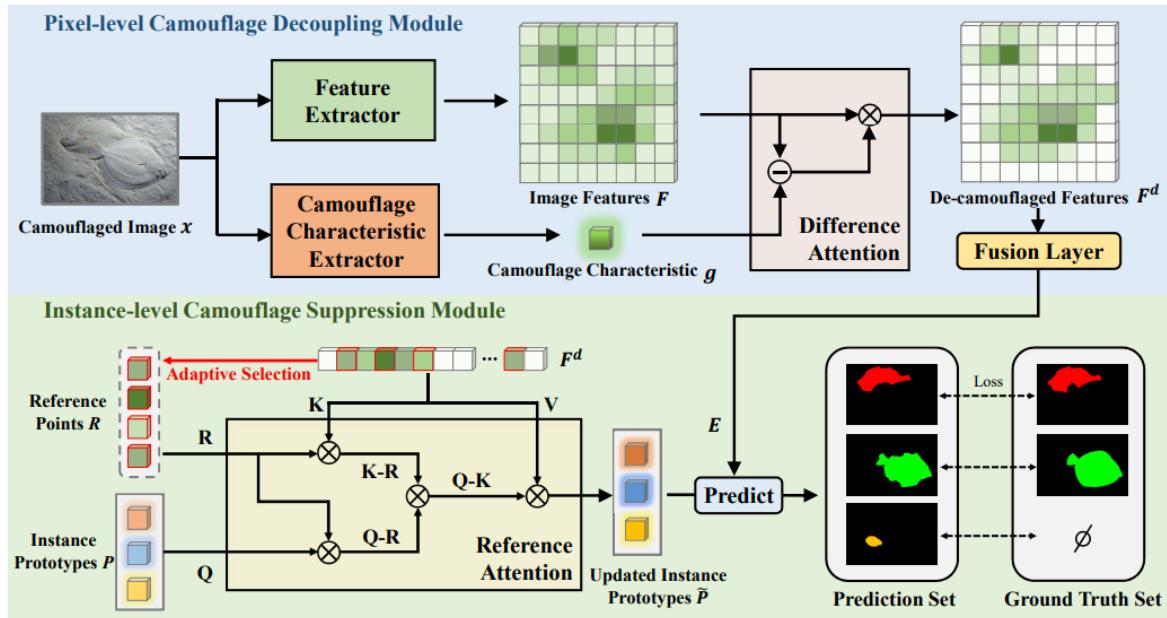


HÌNH 2.6: Kiến trúc mô hình Mask Transfuser [33] với cơ chế sử dụng cây tứ phân (Quadtree) để học đặc trưng tại các vùng dễ gặp lỗi và mô-đun Quadtree Transformer để tạo ra mặt nạ phân đoạn hiệu quả.

### Mô hình DCNet [48]

Gần đây, DCNet [48] được giới thiệu với một cơ chế chống ngụy trang (de-camouflaging mechanism) để trích xuất đặc trưng của các thực thể ngụy trang. Cụ thể, mô hình này tìm cách phá vỡ cơ chế ngụy trang của các thực thể ngụy trang bằng cách chuyển đổi đặc trưng từ không gian đặc trưng sang không gian tần số (frequency domain). Các đặc trưng khó phân biệt ở không gian đặc trưng thông thường sẽ trở nên nổi trội nhờ phương pháp khuếch đại tần số. Qua đó, mô hình sẽ phát hiện các thực thể ngụy trang ở không gian tần số này. DCNet có khả năng thực hiện cơ chế chống ngụy trang ở cấp độ điểm

ảnh và cấp độ thực thể để giải quyết bài toán trên thực thể nguy trang. Kết quả đầu ra của hai khối này được kết hợp để đưa ra dự đoán nhãn mặt nạ thực thể nguy trang cuối cùng. Hình 2.7 trình bày mô hình DCNet phân đoạn thực thể nguy trang với cơ chế chống nguy trang ở cấp độ điểm ảnh và cấp độ thực thể.



HÌNH 2.7: Kiến trúc mô hình Mask DCNet [48] với 2 khối chức năng lần lượt phục vụ ở cấp độ điểm ảnh và cấp độ thực thể.

### 2.2.2 Phân đoạn thực thể với kiến trúc một giai đoạn

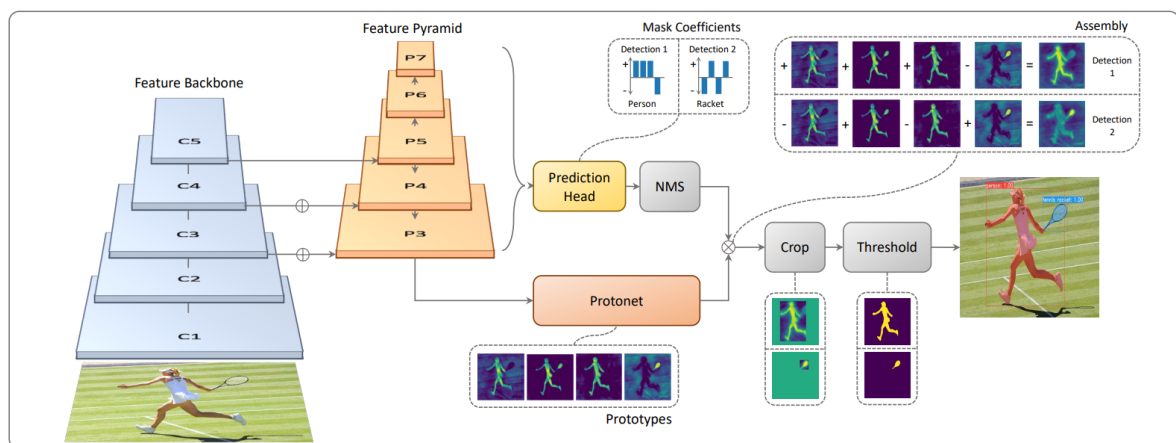
Với hướng tiếp cận một giai đoạn, các mô hình có đặc điểm chung là sử dụng hướng tiếp cận phát hiện đối tượng không sử dụng điểm neo (anchor-free detection) và một số phương pháp tiên tiến gần đây sẽ dựa trên kiến trúc Transformer [73] để xây dựng mô hình. Tiêu biểu cho các phương pháp này có thể kể đến như: YOLACT [1], CondInst [71], SOTR [23], SOLO [78], hay OSFormer [59]. Sau đây là một số nét đặc trưng của các phương pháp trong hướng tiếp cận một giai đoạn để phân đoạn thực thể.

#### Mô hình YOLACT [1]

YOLACT [1] là một trong những phương pháp đầu tiên phân đoạn thực thể theo thời gian thực bằng cách kết hợp kết quả của hai tác vụ: tạo ra một tập những mặt nạ mẫu không chứa vùng cục bộ (non-local prototype masks) và dự đoán tương quan giữa các



mặt nạ thực thể. Mục tiêu của nhóm tác giả là gắn thêm một nhánh phân đoạn vào mô hình phát hiện đối tượng một giai đoạn hiện có theo cùng một phương thức như Mask R-CNN [28] làm với Faster R-CNN [63], nhưng không cần một bước trung gian xác định vị trí. Để làm được điều này, mô hình chia nhỏ nhiệm vụ phức tạp của tác vụ phân đoạn thực thể thành hai nhánh xử lý nhiệm vụ đơn giản hơn, tiến hành song song với nhau. Hình 2.8 trực quan hóa kiến trúc mô hình YOLACT được xây dựng trên nền tảng RetinaNet [42] với backbone ResNet-101 + FPN.

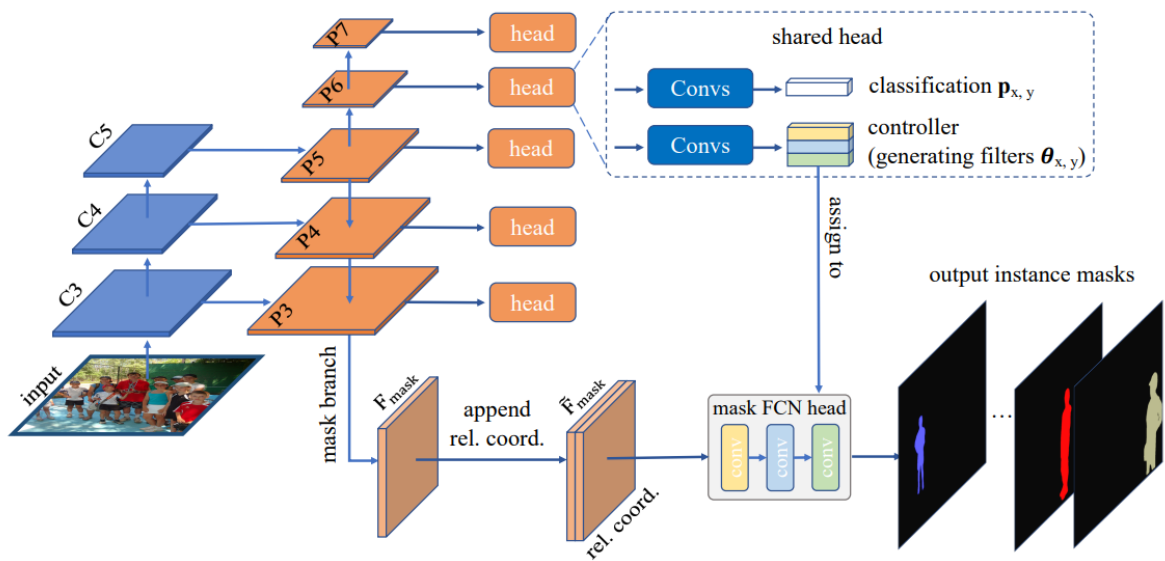


HÌNH 2.8: Kiến trúc mô hình YOLACT [1]. Kiến trúc xây dựng trên nền tảng RetinaNet [42] với backbone ResNet-101 + FPN. Các khối màu xám là những khối chức năng không được cập nhật trong quá trình huấn luyện.

### Mô hình CondInst [71]

CondInst [71] có thể giải quyết bài toán phân đoạn thực thể với mạng tích chập đầy đủ FCNs [67] trong khi loại bỏ việc sử dụng mô-đun xác định vùng quan tâm ROI và mô-đun căn chỉnh đặc trưng (feature alignment). Nếu như phân đoạn ngữ nghĩa chỉ yêu cầu xác định nhãn ngữ nghĩa cho từng điểm ảnh, thì phân đoạn thực thể đòi hỏi xác định thêm điểm ảnh đó thuộc về thực thể nào tồn tại trong bức ảnh đó. Điều này dẫn đến một thách thức lớn cho tác vụ phân đoạn thực thể sử dụng các kiến trúc bottom-up như các mô hình FCNs [67]. Với CondInst [71], ý tưởng tổng quát là sử dụng  $K$  đầu ra phân đoạn cho  $K$  thực thể trong ảnh đầu vào một cách linh hoạt, và mỗi đầu ra phân đoạn sẽ học đặc trưng của riêng thực thể mà đầu ra đó cần dự đoán. Hình 2.9 minh họa tổng quan về mô hình CondInst với các nhánh đầu ra phân đoạn chuyên biệt cho từng thực thể trong ảnh đầu vào.





HÌNH 2.9: Kiến trúc mô hình CondInst [71] với các đầu ra phân đoạn (mask heads) chuyên biệt cho từng thực thể của ảnh đầu vào.

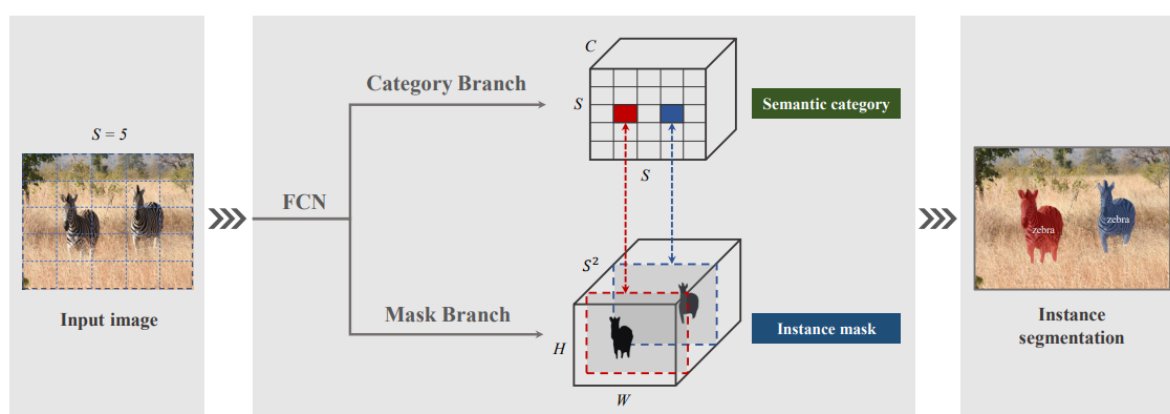
### Mô hình QueryInst [17]

QueryInst [17] là một phương pháp tận dụng các tương quan một-một nội tại của các truy vấn đối tượng và các đặc trưng mặt nạ giữa các giai đoạn khác nhau. Phương pháp này cho phép loại bỏ các tương quan giữa các mặt nạ không tương đồng giữa các giai đoạn và giải quyết vấn đề phân phối đặc trưng không thống nhất khi kế thừa các phương pháp đa giai đoạn không sử dụng truy vấn. QueryInst [17] được công bố là một trong những mô hình đầu tiên với hướng tiếp cận dựa trên truy vấn cho bài toán phân đoạn thực thể, đạt hiệu năng tốt hơn so với một số công trình hai giai đoạn trước đây không dựa trên truy vấn và vẫn giữ được tính chất nhanh đặc thù của mô hình một giai đoạn. Khi so sánh với các mô hình như SOLO [78], CondInst [71], Cascade Mask R-CNN [2] hay HTC [6], QueryInst [17] cho thấy độ chính xác và tốc độ thực thi vượt trội hơn đáng kể. QueryInst [17] được công bố là một trong những mô hình đầu tiên với hướng tiếp cận dựa trên truy vấn cho bài toán phân đoạn thực thể, đạt độ chính xác tốt hơn so với một số công trình hai giai đoạn trước đây không dựa trên truy vấn và vẫn giữ được tính chất nhanh đặc thù của mô hình một giai đoạn.

### Mô hình SOLO [78]

SOLO [78] tái định nghĩa phân đoạn thực thể bằng cách dự đoán phân loại sau đó tạo ra mặt nạ ngữ nghĩa. Phương pháp này tận dụng nhãn ngữ nghĩa để xác định vị trí trung

tâm của thực thể và phân chia việc dự đoán nhãn mặt nạ vào việc học đặc trưng động (dynamic kernel learning). Theo đó, mặt nạ nhãn đầu ra được tạo thành mà không cần sử dụng khung bao. SOLO sử dụng cách thức để xấp xỉ vị trí của tâm thực thể để xác định vị trí của chúng thông qua tâm này. Do đó, bằng cách phân loại các điểm ảnh vào các nhóm theo các tâm, mô hình đã dự đoán được tâm của chính thực thể đó trong không gian đặc trưng. Điểm quan trọng ở đây là việc chuyển đổi tác vụ xác định vị trí thành bài toán phân loại là một tác vụ đơn giản hơn, đồng thời cũng không cần thêm các bước hậu xử lý như gom cụm hay nhúng đặc trưng (embedding). Hình 2.10 thể hiện kiến trúc tổng quát của mô hình SOLO-v1 [78] với hai nhánh phân loại và phân đoạn ngữ nghĩa

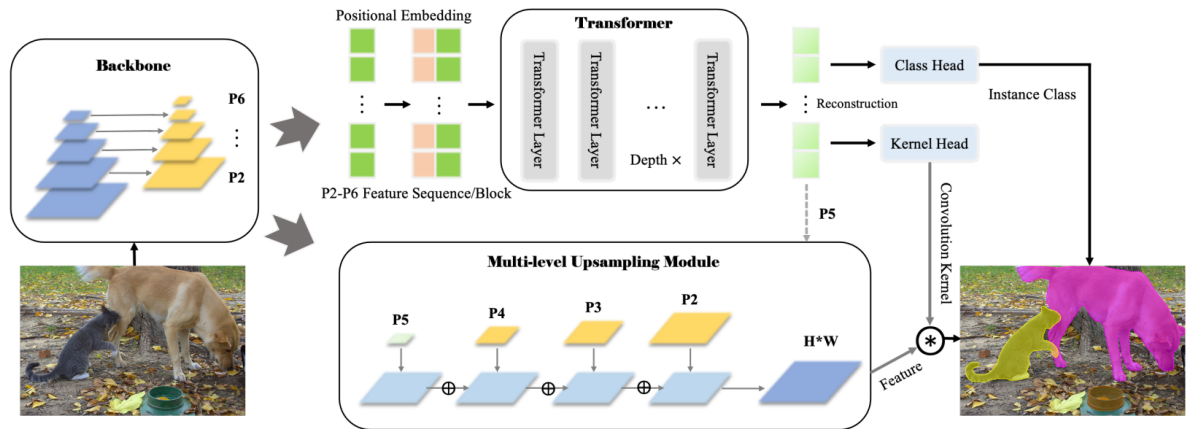


HÌNH 2.10: Kiến trúc mô hình SOLO-v1 [78] với hai nhánh phân loại và phân đoạn ngữ nghĩa.

### Mô hình SOTR [23]

Mô hình SOTR [23] ra đời để cải thiện hiệu quả việc học thông tin đặc trưng vị trí và tạo ra bản đồ phân đoạn mà không sử dụng thêm các bước hậu xử lý với thông tin khung bao. Hơn nữa, nhận thấy sự ảnh hưởng tích cực của kiến trúc Transformer trong các công trình thuộc nhánh xử lý ngôn ngữ tự nhiên, SOTR [23] cũng tận dụng kiến trúc này để trích xuất thông tin toàn cục của ảnh và biểu diễn các đặc trưng phụ thuộc ngữ nghĩa xa (long-distance semantic dependencies). Điểm chính của kiến trúc Transformer chính là cơ chế Self-Attention, giúp trích xuất thông tin đặc trưng đi kèm với thông tin vị trí. Do đó, các kiến trúc dựa trên Transformer sẽ phân biệt tốt hơn các thực thể khi thi hành tác vụ phân đoạn thực thể trên ảnh. Tuy nhiên, các kiến trúc Transformer cũng có điểm yếu khi chưa trích xuất tốt thông tin đặc trưng cấp thấp, khiến cho các thực thể có

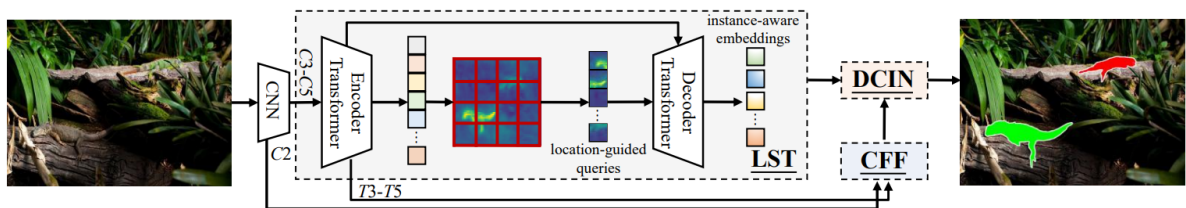
kích thước nhỏ dễ bị bỏ qua. Đồng thời, việc tính toán cũng yêu cầu nhiều tài nguyên hơn. Hình 2.11 mô hình hóa kiến trúc tổng thể của mô hình SOTR [23] với hướng tiếp cận bottom-up và các mô-đun tận dụng điểm mạnh của kiến trúc CNNs và Transformer.



HÌNH 2.11: Kiến trúc mô hình SOTR [23] với các mô-đun tận dụng điểm mạnh của kiến trúc CNNs và Transformer.

### Mô hình OSFormer [59]

OSFormer [59] cải tiến kiến trúc Transformer [73] với khả năng nhạy bén hơn trong việc tích hợp thông tin về vị trí và đề xuất thêm một mô-đun kết hợp từ thô-đến-mịn (coarse-to-fine fusion) để kết hợp thông tin ngữ cảnh với thông tin đặc trưng trích xuất từ bộ mã hóa và đặc trưng từ mạng học sâu. Mô hình OSFormer [59] được đề xuất với kiến trúc Transformer một giai đoạn đặc thù cho phân đoạn thực thể ngẫu trang. Mô hình này được huấn luyện và công bố kết quả trên các tập dữ liệu chuẩn trong nghiên cứu đối tượng ngẫu trang. Một trong những điểm cộng của mô hình này là quá trình huấn luyện diễn ra end-to-end, nghĩa là không cần thêm các bước chuyển đổi trung gian trong quá trình huấn luyện. Để tận dụng tốt các gợi ý về vị trí của thực thể ngẫu trang, OSFormer



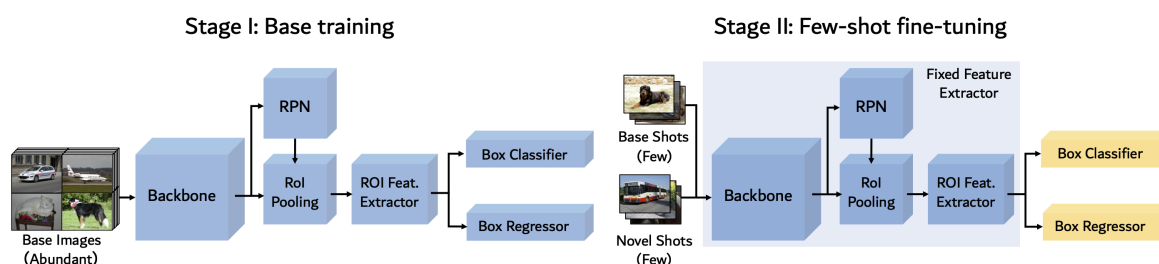
HÌNH 2.12: Kiến trúc mô hình OSFormer [59] với khối Transformer và hai khối chức năng được đề xuất là DCIN và CFF.

sử dụng mô-đun location-sensing transformer (LST) gồm một bộ mã hóa với các lớp tích chập kết hợp (blend-convolution) để trích xuất thông tin đặc trưng đa kích thước và một bộ giải mã với các truy vấn hướng vị trí (location-guided query) để giải mã các đặc trưng nguy trang. Cuối cùng, một mô-đun kết hợp mặt nạ coarse-to-fine fusion (CFF) được đề xuất để kết hợp đặc trưng ở các cấp độ từ đơn giản đến chi tiết giúp tạo ra mặt nạ phân đoạn thực thể chính xác hơn. Hình 2.12 thể hiện kiến trúc tổng quan của mô hình với khối Transformer và hai khối chức năng được đề xuất là khối chuẩn hóa tích chập theo thực thể (DCIN) và CFF.

### 2.2.3 Hướng tiếp cận sử dụng ít dữ liệu huấn luyện

**Phát hiện đối tượng ít dữ liệu huấn luyện.** Khi có một số mẫu có sẵn của các lớp đã cho với các khung bao tương ứng của chúng, FSOD nhằm mục đích học từ những dữ liệu hạn chế này để giúp các mô hình thích ứng với các lớp mới. Cho đến nay, một số công trình [14, 32, 77, 84] đã được đề xuất để giải quyết FSOD. Các công trình sơ khởi [32, 84] chủ yếu đưa ra giải pháp để giải quyết vấn đề khan hiếm dữ liệu của FSOD thông qua các phương pháp học bằng cách kết hợp thông tin hỗ trợ từ các meta-data để bổ sung cho quá trình huấn luyện. Cụ thể, Bingyi [32] đề xuất một mô hình đánh trọng số đặc trưng (Feature Reweighting) tận dụng phương pháp đề xuất tự do của một mô hình một giai đoạn nổi tiếng như YOLO [62] để nâng cao hiệu suất FSOD. Mạng tích hợp một siêu mô hình nhằm mục đích tạo ra các vector trọng số từ các mẫu hỗ trợ để làm nổi bật sự chú ý đến các đặc trưng từ mạng YOLO. Ngược lại, Meta RCNN [84] dựa trên phương pháp đề xuất hai giai đoạn như Mask RCNN [28]. Fan và cộng sự [14] gần đây đề xuất tận dụng các hình ảnh hỗ trợ từ một tập dữ liệu FSOD khổng lồ để tạo ra kết quả đáng kể kết hợp với mạng được đề xuất của họ gọi là Attention-RPN, Các Bộ Phát Hiện Đa Quan Hệ (Multi-Relation Detectors). Attention-RPN chỉ dẫn mô hình được huấn luyện để tập trung vào hình ảnh cho tác vụ phát hiện đối tượng. Khác biệt hơn, Wang và cộng sự [77] đơn giản áp dụng Faster RCNN với hai giai đoạn tinh chỉnh để chuyển giao kiến thức rộng lớn từ dữ liệu phong phú trong mô hình cơ sở để tinh chỉnh cái mới bằng cách đóng băng toàn bộ mạng, ngoại trừ lớp kết nối đầy đủ cho phân loại đối tượng. Thông

qua cơ chế này, mô hình này cải thiện hiệu suất ít mẫu đáng kể mà không cần phức tạp hóa việc huấn luyện mô hình. Hình 2.13 trình bày tổng quan các khối trong kiến trúc mô hình TFA, đây được xem như là mô hình cơ sở để đề xuất nhiều mô hình trong hướng tiếp cận này.



HÌNH 2.13: Kiến trúc mô hình TFA [77] phát hiện và phân đoạn thực thể với hướng tiếp cận học ít dữ liệu (few-shot learning). Đây là một trong những kiến trúc nền tảng cho hướng tiếp cận này.

**Phân đoạn nhị phân đối tượng ngẫu trang.** Trước khi diễn ra sự bùng nổ của các kiến trúc mạng nơ-ron học sâu, hầu hết các công trình phân đoạn nhị phân đối tượng ngẫu trang đều dựa trên các đặc trưng thủ công (đặc trưng cấp thấp), đặc biệt là các loại đặc trưng liên quan đến hình thái bên ngoài của đối tượng như màu sắc, hình dáng, hướng quay, hay độ sáng. Các công trình đầu tiên trên đối tượng ngẫu trang là phát hiện đối tượng ngẫu trang trên vùng nền có đặc trưng tương đồng ([19, 83]). Các công trình này phát hiện đối tượng thông qua các đặc trưng về màu sắc, cường độ, độ sáng, hình dạng, hay biên cạnh, hay gọi chung là các đặc trưng cấp thấp. Một số phương pháp [18, 29, 47, 57, 58, 85] dựa trên đặc trưng cấp thấp hay đặc trưng thủ công đã được đề xuất để xử lý bài toán phát hiện đối tượng ngẫu trang. Tuy nhiên, các phương pháp này chỉ hiệu quả với các ảnh có vùng nền đơn giản. Các loại đặc trưng cấp thấp được khai thác bằng các thuật toán đơn giản, hay dựa trên các giá trị màu nên chưa tận dụng triệt để mối tương quan giữa các loại đặc trưng này và thực thể ngẫu trang. Do đó, độ chính xác của các phương pháp này không ấn tượng khi phát hiện hay phân đoạn đối tượng ngẫu trang được đặt trong các ảnh có vùng nền phức tạp.

**Phân đoạn ngữ nghĩa với ít dữ liệu huấn luyện.** Gần đây, bài toán phân đoạn sử dụng ít mẫu dữ liệu đã thu hút sự chú ý của cộng đồng. Như đã đề cập ở trên, công trình đầu tiên Meta RCNN bắt nguồn từ Mask RCNN, do đó, Meta RCNN đồng thời thực hiện

2 tác vụ phát hiện và phân đoạn. Liu và cộng sự [45] sử dụng một mạng tham chiếu chéo cho phân đoạn hình ảnh tổng quát. Các tác giả đề xuất một cơ chế tham chiếu chéo và một mô-đun tinh chỉnh mặt nạ nhân để hỗ trợ cụ thể cho nhiệm vụ phân đoạn. Trước đó, Dong và cộng sự [11] đề xuất một thành phần học nguyên mẫu trong một khung phân đoạn ngữ nghĩa học để lấy thông tin phân biệt từ các đặc trưng để giúp phân đoạn các đối tượng tốt hơn. Cũng vậy, Wang và cộng sự [75] giới thiệu một phương pháp căn chỉnh nguyên mẫu học các biểu diễn đặc trưng cụ thể từ một số ít mẫu hình ảnh để thực hiện phân đoạn trên các hình ảnh truy vấn. Gần đây, Liu và cộng sự đề xuất một mạng tích chập nguyên mẫu động để giải quyết vấn đề phân đoạn ngữ nghĩa với ít mẫu. Công trình của [65] đề xuất học nguyên mẫu nhận thức bối cảnh. [72] giới thiệu phương pháp mô hình sinh cho nhiệm vụ này. Gần đây, Nguyen và cộng sự [56] đã đề xuất iFS-RCNN, một bộ phân tách thực thể thông qua một phương pháp gia tăng. Gao và cộng sự [21] đề xuất khung DCFS, một bộ phân loại tách rời hiệu quả tăng cường hiệu suất của các bộ phát hiện và phân tách đối tượng. Han và cộng sự [25] gợi ý một khung dựa trên biến đổi tham chiếu hai lần (RefT) để nâng cao các đặc trưng trong các nhiệm vụ phân đoạn. Cũng trong phương pháp biến đổi, Wang và cộng sự [74] giới thiệu DTN để trực tiếp phân tách các thực thể đối tượng mục tiêu từ các lớp tùy ý cho trước các hình ảnh tham chiếu. Nhìn chung, các phương pháp đã nêu trên tập trung vào các đối tượng tổng quát.

## 2.3 Các hướng tiếp cận khai thác đặc trưng có tính phân biệt cao

Trong luận văn này, hướng tiếp cận khai thác đặc trưng có tính phân biệt cao của các thực thể nguy trang được tiến hành thông qua hai phương pháp chính, đó là sử dụng đặc trưng biên cạnh và sử dụng phương pháp học tương phản.

### 2.3.1 Tăng cường đặc trưng biên cạnh

**Sử dụng đặc trưng thủ công.** Cũng giống với đa số các phương pháp trong thị giác máy tính, từ ban đầu, để giải quyết vấn đề phát hiện biên cạnh, cộng đồng nghiên cứu

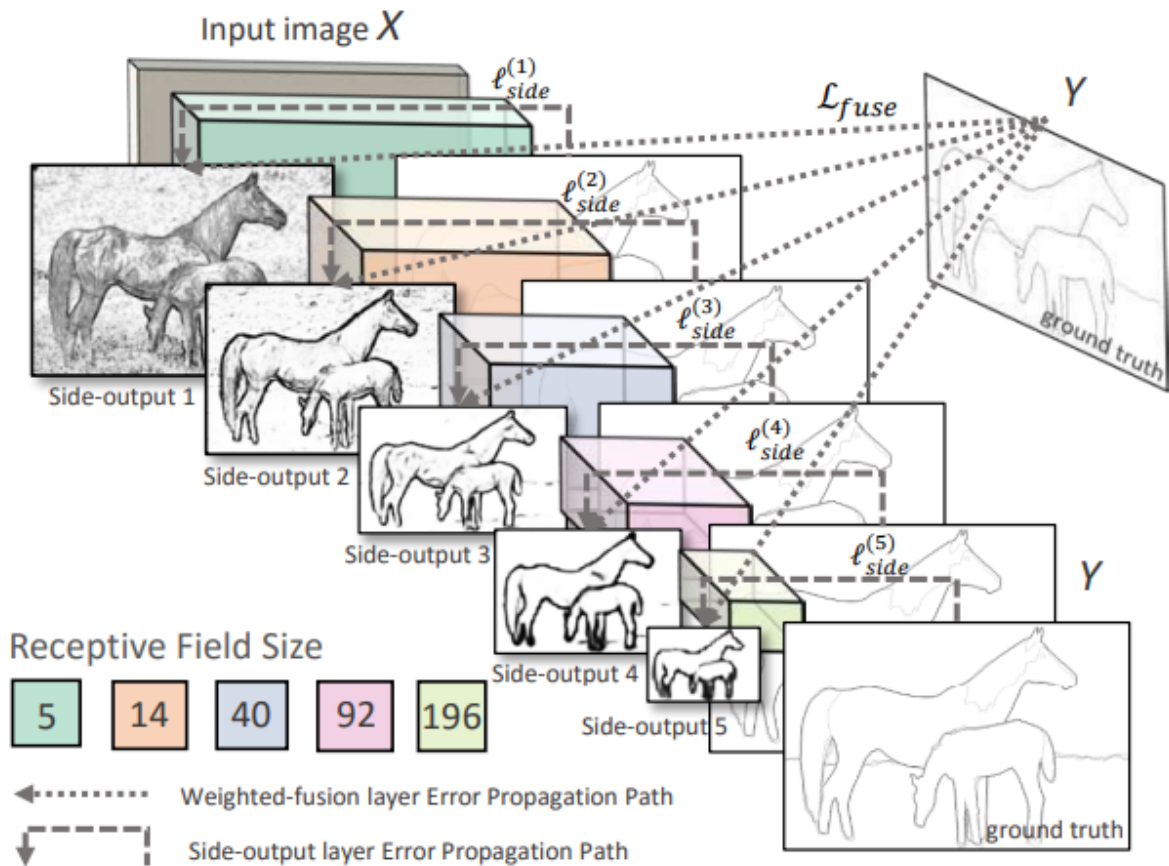


đã dựa vào các lý thuyết toán học để áp dụng lên các ma trận màu của ảnh mà phát hiện biên cạnh. Chúng ta có thể tạm chia thành hai hướng tiếp cận là phát hiện trực tiếp và phát hiện gián tiếp. Phát hiện trực tiếp biên cạnh vật thể dựa trên sự biến thiên mức xám trên ảnh đó. Vì thế, các kĩ thuật dựa trên đạo hàm bậc nhất như Gradient, hay đạo hàm bậc hai như Laplace. Đây là hai phương thức điển hình nhất. Với hướng phát hiện gián tiếp biên cạnh, chúng ta sử dụng một bước đệm phân vùng ảnh. Việc phân vùng ảnh giúp cho các phần của bức ảnh được chia tách, và tại các biên của những phân vùng này, chúng ta có được biên của đối tượng. Các kĩ thuật phân vùng ảnh sử dụng phương pháp gom cụm như KMeans sẽ hữu dụng trong trường hợp này. Một số phương pháp chúng ta thường nghe nhắc đến như sử dụng bộ lọc Sobel với ma trận có giá trị phù hợp để trích xuất biên cạnh, hay thuật toán phát hiện biên cạnh Canny, là những cái tên thường được nhắc đến khi xử lý biên cạnh sử dụng đặc trưng thủ công. Nhìn chung, các phương pháp dựa trên đặc trưng thủ công sẽ có ưu điểm nhất định về thời gian thực thi, song độ chính xác bị hạn chế vì các yếu tố màu sắc, chất liệu trên ảnh màu thường phức tạp. Hơn nữa, trong luận văn này chúng tôi hướng đến các đối tượng nguy trang, những đối tượng đặc thù này sẽ làm lộ rõ yếu điểm của các phương pháp dựa trên đặc trưng cấp thấp.

**Sử dụng đặc trưng học sâu.** Các phương pháp dựa trên học sâu thông thường sẽ đi theo cấu trúc sử dụng một mạng tích chập để trích xuất đặc trưng của ảnh đầu vào. Các cải tiến cho tác vụ phát hiện biên cạnh sẽ được đề xuất để cải tiến kiến trúc mạng này hoặc thêm vào các mô-đun chuyên biệt để hậu xử lý đặc trưng, từ đó tăng cường khả năng của mô hình học sâu trong việc phát hiện biên cạnh. Tiêu biểu cho các phương pháp phát hiện biên cạnh với kiến trúc mạng học sâu, có thể kể đến như HED - Holistically-Nested Edge Detection [82] và COB – Convolutional Oriented Boundaries [50, 51]. Một trong những thách thức với tác vụ huấn luyện mô hình phát hiện biên cạnh chính là các tập dữ liệu, chúng thường giới hạn ở các đối tượng tổng quát và chưa được nghiên cứu đề xuất cho các trường hợp đặc thù như đối tượng nguy trang.

### **Phương pháp HED - Holistically-Nested Edge Detection [82].**

HED [82] sử dụng kiến trúc mạng CNNs với đa kích thước trích xuất đặc trưng để học đặc trưng biên cạnh (Hình 2.14). HED giải quyết hai vấn đề chính: 1) học và dự đoán



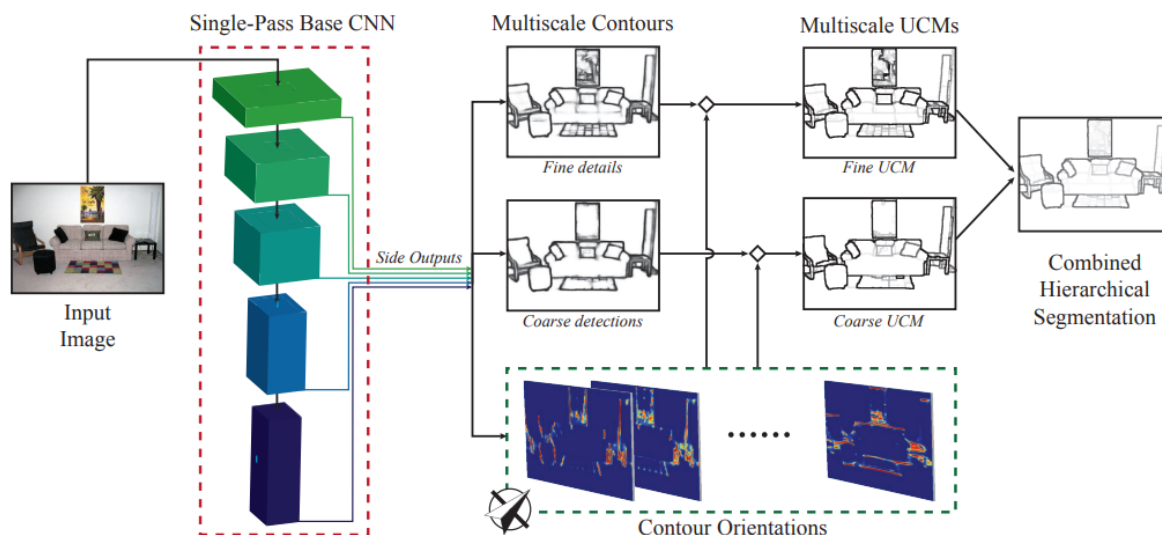
HÌNH 2.14: Kiến trúc mô hình HED [82] phát hiện biên cạnh vật thể.

một cách toàn diện dựa trên kiến trúc mạng tích chập đầy đủ FCNs [67] với tác vụ dự đoán ảnh-ảnh. Cụ thể, mô hình nhận ảnh đầu vào và trả về ảnh chứa bản đồ biên cạnh; 2) học các đặc trưng được tổng hợp ở đa kích thước, lấy cảm hứng từ mạng học sâu có giám sát, sử dụng phương pháp giám sát trên từng lớp kiến trúc mạng để hướng dẫn quá trình học. Hai đặc điểm này tuy cơ bản nhưng giúp HED dự đoán biên cạnh vừa chính xác vừa tối ưu chi phí thực hiện. Mô hình này được huấn luyện với tập dữ liệu BSD500 [53] và NYU Depth (NYUD)-v2 [68], đây là hai tập dữ liệu chuẩn về phát hiện biên cạnh trên các đối tượng tổng quát. Về kiến trúc mạng CNN để rút trích đặc trưng, mô hình HED thử nghiệm với nhiều biến thể khác nhau dựa trên kiến trúc CNN của mạng VGG. Các kiến trúc biến thể của mạng FCNs [67] như FCN-2s, hay FCN-8s cũng được thử nghiệm cùng với các kết nối tắt (skip-connection) giúp tăng cường khả năng học hiểu các đặc trưng của mô hình. Các thử nghiệm trên giúp tìm ra phiên bản tối ưu cho tác vụ phát hiện biên cạnh của các đối tượng và tăng độ chính xác dự đoán của mô hình đề xuất. Mô



hình HED cuối cùng là kết quả được kết hợp từ các đặc trưng rút trích từ nhiều vị trí của kiến trúc mạng, gồm có các đặc trưng phụ (side output) và các đặc trưng mang tính toàn cục được rút trích tại các lớp cuối của kiến trúc.

**Phương pháp COB – Convolutional Oriented Boundaries [50, 51].**

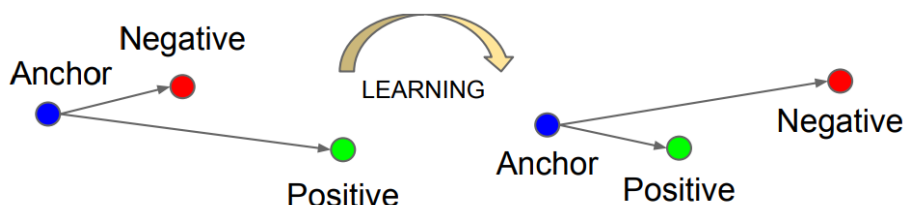


HÌNH 2.15: Kiến trúc mô hình COB [50] phát hiện biên cạnh có hướng.

COB [50, 51] là một kiến trúc mạng CNNs end-to-end thiết kế để học các đường biên cạnh có hướng ở nhiều kích thước khác nhau (multiscale oriented contours), minh họa tại Hình 2.15. COB có thể tận dụng kiến trúc nền tảng từ các mô hình CNNs cho độ chính xác cao để học dự đoán biên cạnh, vì thế, chúng ta có thể áp dụng các kiến trúc khác nhau nhằm tăng độ chính xác của tác vụ phát hiện biên cạnh. Nhóm tác giả cũng đề xuất phương pháp biểu diễn biên cạnh thưa (sparse boundary representation) để xây dựng một cách hiệu quả các vùng phân cấp từ đặc trưng biên cạnh. Về hiệu năng, COB mất 0.8 giây để dự đoán biên cạnh trên một ảnh và đạt độ chính xác cao trên hai tập dữ liệu cho phát hiện biên cạnh là PASCAL [55] và BSDS [52]. Điểm đột phá của COB là khả năng khai thác thông tin về hướng của các đặc trưng trong quá trình học, hơn nữa, các thông tin về hướng của đường biên được học ở nhiều kích thước khác nhau. Việc kết hợp thông tin đặc trưng học sâu và thông tin đặc trưng có hướng giúp xác định đường biên cạnh của các đối tượng một cách hiệu quả.

### 2.3.2 Phương pháp học tương phản

Phương pháp học tương phản là một kỹ thuật học sâu không giám sát với mục tiêu học cách biểu diễn dữ liệu sao cho các thực thể tương đồng được gom lại gần nhau trong không gian đặc trưng, và tách xa các thực thể không tương đồng [8, 9, 27, 66]. Các nghiên cứu được công bố với hướng tiếp cận học tương phản đã chứng minh tính hiệu quả trong nhiều lĩnh vực như thị giác máy tính hay xử lý ngôn ngữ tự nhiên, cụ thể hơn là các tác vụ như truy vấn ảnh (image retrieval), học với không mẫu dữ liệu (zero-shot learning) hay truy vấn đa phương thức (cross-modal retrieval). Trong các tác vụ này, các biểu diễn đặc trưng được học có thể được sử dụng cho các tác vụ con khác như phân loại hay gom cụm.



HÌNH 2.16: Trực quan hóa hướng tiếp cận học tương phản với các mẫu biểu diễn Positive, Negative, và Anchor. Quá trình học tương phản sẽ tìm cách thu hẹp khoảng cách giữa các điểm biểu diễn Positive và Anchor, trong khi đẩy xa khoảng cách giữa các điểm biểu diễn Negative và Anchor.

**Hình 2.16** mô tả trực quan phương pháp học tương phản. Cụ thể, quá trình học tương phản sẽ tìm cách thu hẹp khoảng cách giữa các điểm biểu diễn Positive và Anchor, trong khi đẩy xa khoảng cách giữa các điểm biểu diễn Negative và Anchor. Một số phương pháp điển hình trong hướng tiếp cận học tương phản là Hàm mất mát ba thành phần (Triplet Loss) [66] hay Bộ nhớ lưu trữ (Memory Bank) [8]. Gần đây, học tương phản còn được áp dụng ở cấp độ *token*, cấp độ ngữ nghĩa nhỏ hơn của các biểu diễn đặc trưng, nhằm tăng cường khả năng phân biệt của mô hình với các mẫu biểu diễn nói trên, đơn cử là phương pháp ToCo [64]. Trong đề tài này, chúng tôi dựa trên hướng tiếp cận của hai phương pháp điển hình trong học tương phản để thực hiện đề xuất của chúng tôi trong việc cải thiện tác vụ phân đoạn thực thể nguy trang.

## 2.4 Các tập dữ liệu chuẩn về thực thể nguy trang

Trong phần này, chúng tôi trình bày khảo sát tổng quan về các tập dữ liệu phục vụ nghiên cứu trên đối tượng nguy trang. Các tập dữ liệu được khảo sát từ các công bố từ các hội nghị, tạp chí có uy tín trong ngành. Cụ thể, chúng tôi đã thực hiện khảo sát trên các tập dữ liệu: CamouflagedAnimals [60], MoCA [35], CHAMELEON [69], COD10K [12], NC4K [49], CAMO [38], và CAMO++ [36] Để có cái nhìn bao quát, sau khi phân tích các số liệu, chúng tôi cung cấp thông tin tại **Bảng 2.1**, giúp người đọc nắm được các thông tin theo các thuộc tính cần thiết.

BẢNG 2.1: Thống kê một số tập dữ liệu về thực thể nguy trang (chỉ xét dữ liệu ảnh/video chứa thực thể nguy trang).

Tập dữ liệu	Năm	Hội nghị	Loại	#Gán nhãn nguy trang	#Lớp tổng quát	#Lớp đối tượng	#Thực thể hoặc #Đối tượng trên ảnh	Nhãn khung bao	Nhãn ngữ nghĩa	Nhãn thực thể	Few-shot
CamouflagedAnimals [60]	2016	ECCV	Video	181	-	6	1.238	×	✓	✓	×
MoCA [35]	2020	ACCV	Video	7,617	-	67	1.000	✓	×	×	×
CHAMELEON [69]	2018	-	Ảnh	76	-	-	1.000	×	✓	×	×
CAMO [38]	2019	CVIU	Ảnh	1,250	2	8	1.000	×	✓	×	×
COD [12]	2020	CVPR	Ảnh	5,066	5	69	1.171	✓	✓	✓	×
NC4K [49]	2021	CVPR	Ảnh	4,121	5	69	1.171	✓	✓	✓	×
CAMO++ [36]	2022	TIP	Ảnh	2,695	10	47	1.171	✓	✓	✓	×
CAMO-FS	2023	-	Ảnh	2,858	10	47	1.172	✓	✓	✓	✓

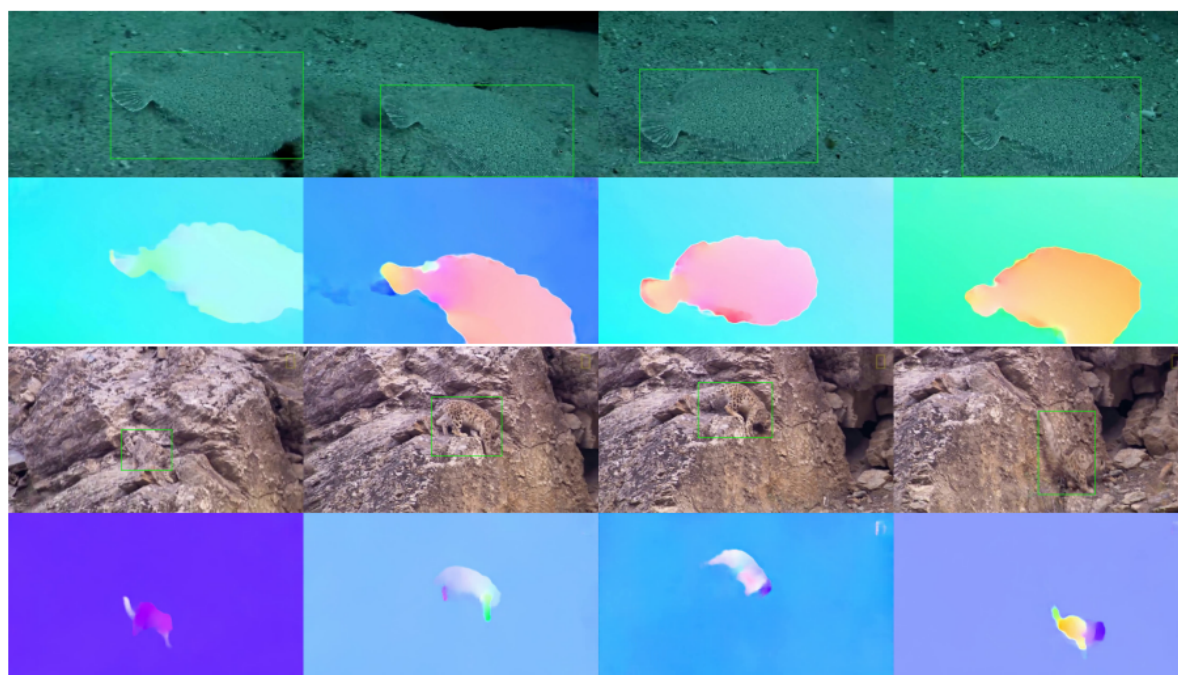
**CamouflagedAnimals [60].** CamouflagedAnimals [60] là một trong những tập dữ liệu đầu tiên được công bố để phục vụ bài toán phân đoạn thực thể nguy trang. Công trình này định nghĩa thực thể nguy trang là các thực thể "khó được phát hiện chỉ với một khung hình", vì thế cần sự hỗ trợ từ các khung hình lân cận để tăng cường cho mô hình. CamouflagedAnimals [60] gồm 09 video ngắn trích từ 04 video<sup>1</sup> đăng tải tại YouTube, mỗi video được trích xuất một số lượng khung hình nhất định, tùy theo vị trí mà đối tượng nguy trang được nhóm tác giả chọn để gán nhãn. Cụ thể, cứ mỗi 5 khung hình, thì ảnh sẽ được gán nhãn phân đoạn. Bên cạnh tác vụ phân đoạn đối tượng nguy trang, tập dữ liệu này đồng thời cũng phục vụ tác vụ phân đoạn hành động (motion segmentation). Đây là đường dẫn đến các video được tập dữ liệu này sử dụng, nhóm tác giả truy cập vào tháng 3 năm 2015.

**MoCA [35].** MoCA [35] được công bố vào năm 2020 là một tập dữ liệu đa dạng về thực thể nguy trang<sup>2</sup>. MoCA [35] gồm có 141 video với tổng cộng 37K khung hình,

<sup>1</sup>Đường dẫn: [youtu.be/{Wc5wMX6IFZ8, yoG1P4newO4, GnEkBJnS\\_FU, adufPBDNCKo}](https://youtu.be/{Wc5wMX6IFZ8, yoG1P4newO4, GnEkBJnS_FU, adufPBDNCKo})

<sup>2</sup><https://www.robots.ox.ac.uk/vgg/data/MoCA/>

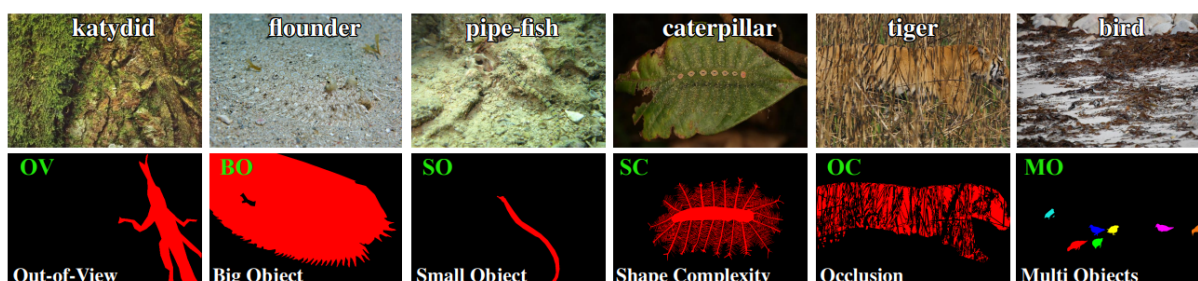
được công bố là tập dữ liệu lớn nhất về video đối tượng ngụy trang tại thời điểm ra mắt. Các đối tượng ngụy trang được chia thành 67 lớp ngữ nghĩa được gán nhãn khung bao và nhãn hành động (optical flow). Hình 2.17 chứa một số hình ảnh trích từ tập dữ liệu MoCA [35] với nhãn khung bao và optical flow. Cũng giống như CamouflagedAnimals [60], các khung hình trích từ video được gán nhãn mỗi 5 khung hình một lần.



HÌNH 2.17: Một số hình ảnh trích từ tập dữ liệu MoCA [35] với nhãn khung bao và optical flow.

**CHAMELEON [69].** CHAMELEON [69] là một trong những tập dữ liệu đầu tiên phục vụ cho nghiên cứu trên đối tượng ngụy trang. Tập dữ liệu này được một nhóm sinh viên tại Đại học Kỹ thuật Silesian, Ba Lan thu thập và thực hiện gán nhãn thủ công. Các hình ảnh của tập dữ liệu này được lấy từ Internet, sử dụng từ khóa "camouflaged animals" và tìm kiếm với công cụ tìm kiếm Google. Các hình ảnh được lựa chọn với mức độ ngụy trang khác nhau, từ có thể nhận biết đến mức độ gần như không thể nhìn thấy bằng mắt thường. Cuối cùng, tập dữ liệu gồm có 76 hình ảnh, đại diện cho các đối tượng ngụy trang, được công bố. Số lượng này là hạn chế khi dùng để huấn luyện các mô hình học sâu. Tuy vậy, đây là một trong những công trình đầu tiên phục vụ đặc thù cho nhánh nghiên cứu này. Nhãn của tập dữ liệu này cũng tương đối thô sơ, được chỉ định với các mức độ như C0 - vùng nền không được gán nhãn, C1 - vùng nền được gán nhãn, C2 -

vùng vật thể được gán nhãn, và C3 - vùng vật thể không được gán nhãn. Cho đến hiện tại, trang chủ của tập dữ liệu này vẫn có thể được truy cập tại đây <sup>3</sup>, tuy nhiên, dữ liệu truy cập không còn được đảm.



HÌNH 2.18: Một số hình ảnh ngụy trang trích từ tập dữ liệu COD10K [12] với nhãn phân đoạn thực thể. COD10K chứng minh tính đa dạng với nhiều điều kiện xuất hiện thực thể ngụy trang khác nhau như bị che khuất (OC), nhiều thực thể (MO), thực thể nhiều kích thước khác nhau (BO)

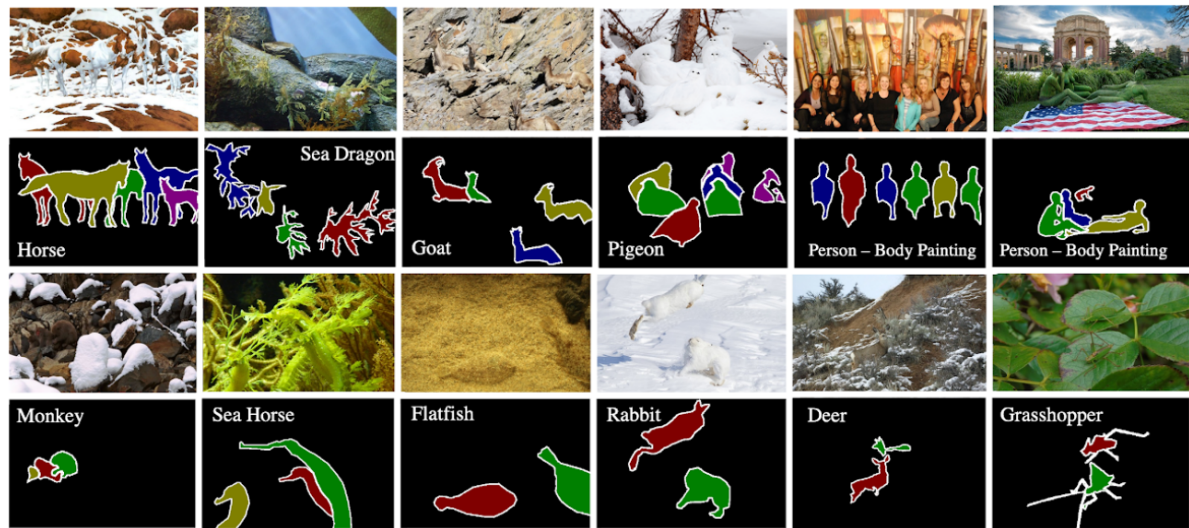
**COD10K [12].** COD10K [12] là một tập dữ liệu lớn với khoảng 10K ảnh ngụy trang và không ngụy trang, được chia thành 5 nhóm ngữ nghĩa lớn. 5 nhóm ngữ nghĩa này bao gồm 69 lớp đối tượng ngụy trang. Tất cả ảnh của COD10K được phân loại theo cấu trúc thứ bậc (hierarchical) và gán nhãn phân loại, khung bao, và nhãn phân đoạn ở cấp độ đối tượng và cấp độ thực thể. Vì thế, COD10K có thể phục vụ tốt cho các tác vụ thị giác máy tính liên quan đến nghiên cứu đối tượng ngụy trang. Theo công bố của nhóm tác giả, trung bình mất khoảng 60 phút để gán nhãn các thực thể ngụy trang cho mỗi ảnh tùy theo mức độ khó. Các nhãn chất lượng tốt này giúp nghiên cứu về thực thể ngụy trang có thêm dữ liệu để thực nghiệm các mô hình và đề xuất mới. Hình 2.18 minh họa một số hình ảnh trích từ tập COD10K, để thấy rằng những hình ảnh này có thể làm khó ngay cả đôi mắt của chúng ta.

**NC4K [49].** Nhóm tác giả của NC4K [49] chỉ ra rằng, các tập dữ liệu lớn như COD10K [12] tuy có nhiều dữ liệu huấn luyện nhưng lại thiếu hụt dữ liệu cho quá trình kiểm tra, cụ thể, COD10K chỉ có khoảng 300 ảnh được sử dụng cho quá trình kiểm tra. Vì thế, NC4K ra đời như một tập dữ liệu phục vụ mục đích kiểm tra. Tập dữ liệu này chứa 4,121 ảnh ngụy trang được thu thập từ Internet và gán nhãn phân đoạn thực thể. Các phân lớp ngữ nghĩa của NC4K phù hợp để sử dụng cùng với COD10K, do đó, các

<sup>3</sup><https://www.polsl.pl/rau6/chameleon-database-animal-camouflage-analysis/>



dữ liệu đã huấn luyện trên COD10K có thể dùng NC4K để tiến hành kiểm tra độ chính xác.



HÌNH 2.19: Một số hình ảnh ngẫu trang trích từ tập dữ liệu CAMO++ [36].

**CAMO [38] và CAMO++ [36].** Le và các cộng sự [38] đề xuất tập dữ liệu CAMO với hơn 1,250 hình ảnh được gán nhãn chi tiết cho tác vụ phân đoạn ngữ nghĩa đối tượng ngẫu trang. Sau đó, CAMO++ [36] được nhóm tác giả công bố, kế thừa từ CAMO nhưng chi tiết hơn và được gán nhãn ở cấp độ thực thể. Sau khi cải tiến CAMO, tập dữ liệu CAMO++ bao gồm 5,500 hình ảnh và có ưu thế hơn các tập dữ liệu khác về số lượng thực thể lên đến 32,756 thực thể cho cả các đối tượng ngẫu trang và không ngẫu trang. Xét về số lớp đối tượng, CAMO++ chứa 93 lớp đối tượng được chia thành 10 lớp ngữ nghĩa lớn. Hình 2.19 cung cấp một số hình ảnh trích từ tập dữ liệu CAMO++ [36].

Trong công trình này, chúng tôi nhận thấy các tập dữ liệu CAMO++ [36], COD10K [12], và NC4K [49] có mức độ đa dạng, độ khó và quy mô phù hợp để huấn luyện và kiểm tra một mô hình học sâu phân đoạn thực thể ngẫu trang. Vì vậy, chúng tôi sử dụng ba tập dữ liệu này để thực nghiệm chứng minh độ chính xác của các mô hình đề xuất.

## 2.5 Tạm kết

Với các nội dung đã trình bày trong phần này, chúng tôi đã mang lại cái nhìn tổng quát về các công trình có liên quan để giúp người đọc tiếp cận bài toán mà chúng tôi

nghiên cứu trong luận văn này. Chúng tôi trình bày các công trình, các kiến thức có liên quan tương ứng với từng đóng góp, đề xuất của chúng tôi cho bài toán phân đoạn thực thể ngẫu trang với hướng tiếp cận khai thác hiệu quả các đặc trưng có tính phân biệt cao. Cụ thể, chúng tôi đã trình bày các mô hình tiêu biểu cho các hướng tiếp cận giải quyết bài toán phân đoạn thực thể ngẫu trang: hướng tiếp cận hai giai đoạn [2, 5, 6, 28, 30, 33, 44, 48], hướng tiếp cận một giai đoạn [1, 23, 59, 71, 78]. Hướng tiếp cận sử dụng ít dữ liệu [11, 21, 25, 45, 56, 65, 72, 74] được trình bày như một phương pháp giúp giải quyết bài toán phân đoạn trong ngữ cảnh thiếu dữ liệu đặc thù của các nghiên cứu trên thực thể ngẫu trang. Các công trình này được dùng làm tham chiếu so sánh độ chính xác của mô hình chúng tôi đề xuất trong các phần tiếp theo. Để khai thác đặc trưng biên cạnh, chúng tôi khảo đã khảo sát các công trình [50, 51, 82] để làm cơ sở đề xuất phương pháp tăng cường đặc trưng biên cạnh của chúng tôi. Cuối cùng, với các thông tin thu được từ việc khảo sát các tập dữ liệu đặc thù [12, 35, 36, 38, 49, 60, 69], chúng tôi chọn ra CAMO++ [36], COD10K [12], và NC4K [49] để đánh giá và so sánh hiệu suất các mô hình phân đoạn thực thể ngẫu trang.

## Chương 3

# Mô hình CE-OST khai thác đặc trưng vùng biên cạnh

### 3.1 Tổng quan

Trong phần này, chúng tôi trình bày chi tiết đề xuất để giải quyết bài toán phân đoạn thực thể nguy trang dựa trên **khai thác đặc trưng phân biệt tại vùng biên cạnh của các thực thể nguy trang**. Cơ chế ẩn mình của các loài động vật nguy trang nhằm hòa nhập kết cấu và màu sắc của chúng với môi trường xung quanh. Do đó, thị giác của con người không thể nhận ra những thực thể đó khi chỉ nhìn thoáng qua. Tuy nhiên, những ranh giới của các loài động vật nguy trang khó có thể biến mất hay hòa lẫn hoàn toàn với môi trường. Trong công trình này, chúng tôi đề xuất một phương pháp để tăng cường đặc trưng biên cạnh của thực thể nguy trang với mục đích hỗ trợ các mô hình phân đoạn để thực hiện tốt tác vụ phân đoạn các thực thể này.

Lấy cảm hứng từ các công trình phát hiện biên cạnh vật thể gần đây [3, 50, 51, 82], chúng tôi đề xuất một phương pháp tăng cường đặc trưng biên cạnh của các thực thể nguy trang trong ảnh để mô hình phân đoạn có thể phân biệt tốt hơn các thực thể này với vùng nền. Về kiến trúc mô hình phân đoạn thực thể, chúng tôi nhận thấy rằng các công trình gần đây trong lĩnh vực thị giác máy tính đã đạt được những thành tựu bùng nổ đáng kể về độ chính xác kể từ sự xuất hiện của công trình [73] về kiến trúc Transformer. Với cơ chế Attention đầy hứa hẹn cho các tác vụ thuộc nhiều lĩnh vực, các công trình [24,



[34] đã áp dụng và đạt được kết quả cao trong lĩnh vực thị giác máy tính. Cụ thể hơn, khi áp dụng kiến trúc Transformer cho các mô hình phân đoạn thực thể, độ chính xác khi dự đoán tác vụ này cũng được cải thiện [16, 39, 59, 87]. Dựa trên hai yếu tố này, chúng tôi tiến hành khảo sát và triển khai các thực nghiệm và cuối cùng đưa ra đề xuất về kiến trúc của mô hình phân đoạn thực thể ngẫu trang với hướng tiếp cận khai thác hiệu quả đặc trưng biên cạnh được trình bày trong phần này.

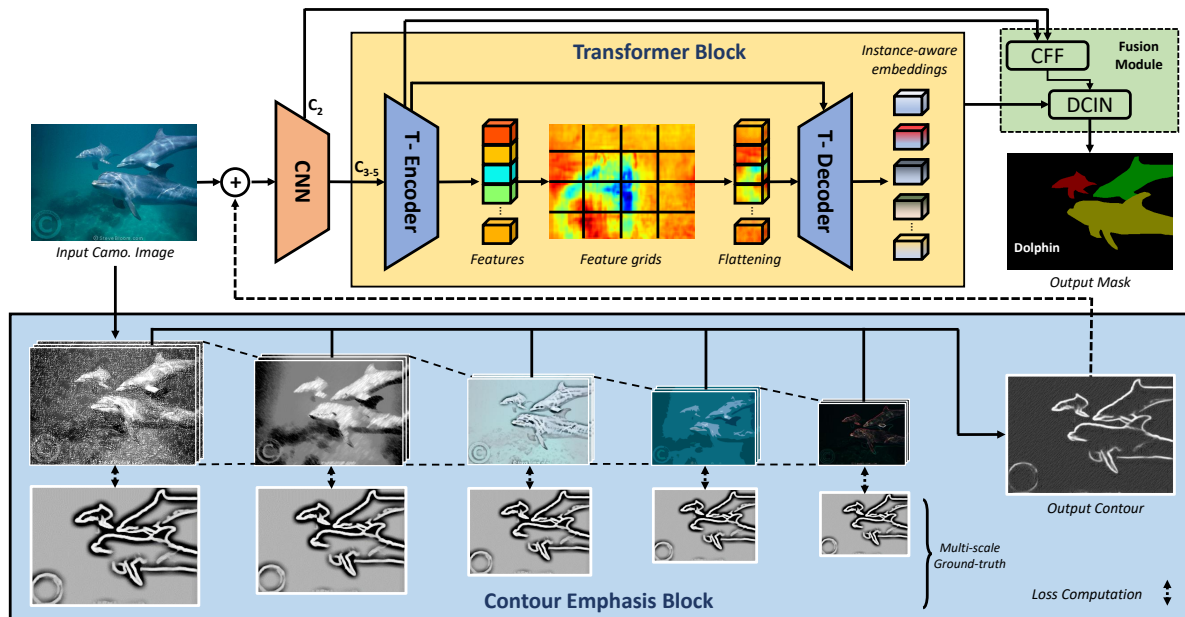
Tổng hợp lại, chúng tôi đề xuất mô hình CE-OST (Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation) với mô-đun tăng cường đặc trưng biên cạnh cho thực thể ngẫu trang. CE-OST là mô hình một giai đoạn dựa trên kiến trúc Transformer [59], vì thế, chúng tôi tận dụng được những khả năng sẵn có của một mô hình Transformer điển hình như độ chính xác cao và tốc độ dự đoán nhanh. Để chứng minh phương pháp của chúng tôi hoạt động tốt, chúng tôi tiến hành các thực nghiệm trên ba tập dữ liệu chuẩn và phổ biến trong lĩnh vực nghiên cứu trên thực thể ngẫu trang, đó là CAMO++ [36], COD10K [12], và NC4K [49]. Kết quả báo cáo cho thấy độ chính xác của phương pháp chúng tôi đề xuất cải thiện hơn so với các phương pháp tiên tiến hiện tại.

## 3.2 Mô hình Transformer một giai đoạn CE-OST

### 3.2.1 Giới thiệu mô hình

Mô hình đề xuất của chúng tôi Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation, viết tắt là CE-OST, được minh họa trong Hình 3.1. CE-OST có hai khối chính: Khối tăng cường đặc trưng biên cạnh (Contour Emphasis Block -CEB) và Khối Transformer (-TB). Một ảnh chứa thực thể ngẫu trang đầu vào cần đi qua hai khối này trước khi gặp mô-đun hợp nhất (Fusion Module) ở phần cuối của mô hình để trả về mặt nạ phân đoạn ở cấp độ thực thể. Khối tăng cường đặc trưng biên cạnh CEB của chúng tôi có thể được thiết kế với cơ chế plug-and-play, do đó, nó có thể dễ dàng được kết hợp với các mô hình khác cần đến đặc trưng tăng cường của biên cạnh. Đầu tiên, ảnh đi qua khối CEB này được tăng cường vùng biên cạnh để các thực thể có

trong ảnh trở nên rõ ràng, làm nổi bật vùng vật thể và vùng nền. Sau đó, ảnh được tăng cường này tiếp tục hành trình của nó để thực hiện quá trình trích xuất đặc trưng thông qua một bộ trích xuất đặc trưng học sâu trước khi đưa vào khối Transformer. Các đặc trưng được trích xuất đi qua khối Transformer để thực hiện các giai đoạn dự đoán mặt nạ phân đoạn thực thể. Chi tiết các mô-đun được giải thích cụ thể trong phần sau.



HÌNH 3.1: Tổng quan mô hình CE-OST (Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation) phân đoạn thực thể ngụy trang dựa trên kiến trúc Transformer một giai đoạn có tăng cường đặc trưng biên cạnh.

### 3.2.2 Khối tăng cường đặc trưng biên cạnh

Biên cạnh đóng vai trò quan trọng trong việc hỗ trợ thị giác của chúng ta nhận biết hình dạng toàn bộ của một thực thể hoặc đối tượng bất kỳ. Hơn nữa, đặc trưng này càng trở nên quan trọng hơn đối với các thực thể ngụy trang, khi mà các thực thể này cố gắng ẩn mình thông qua việc hòa lẫn màu sắc, chất liệu của cơ thể với môi trường xung quanh. Các công trình trước đây về phát hiện biên cạnh dựa trên đặc trưng thủ công như phương pháp Canny Edge Detection đến các công trình áp dụng các phương pháp sâu như Holistically Edge Detection - HED [82], hoặc Convolutional Oriented Boundaries - COB [50, 51] đã đạt được độ chính xác nổi bật về phát hiện biên cạnh. Trong công

trình này, chúng tôi đề xuất một phương pháp tăng cường đặc trưng biên cạnh thực thể để tăng cường các đặc trưng thị giác của các thực thể nguy trang nhằm cải thiện mô hình phân đoạn. Trong **Hình 3.1**, chúng tôi giới thiệu khối tăng cường đặc trưng biên cạnh Contour Emphasis (CE) có nhiệm vụ hợp nhất biên cạnh với ảnh gốc để tăng cường khả năng nhận diện bằng thị giác.

Xuất phát từ ý tưởng của HED [82], chúng tôi sử dụng một mạng tích chập nhiều tầng với các tỉ lệ không gian khác nhau, sử dụng backbone VGG-16 được huấn luyện sẵn để phát hiện biên cạnh của thực thể. Đầu tiên, toàn bộ khối CE được huấn luyện trên một tập dữ liệu phát hiện biên cạnh, gọi là tập dữ liệu BSDS500 [53]. Các hàm mất mát được tính toán ở nhiều tỷ lệ sử dụng Cross-Entropy để tính toán mất mát theo từng điểm ảnh và được tổng hợp thành một hàm mất mát tổng. Để giảm chi phí gán nhãn biên cạnh, chúng tôi sử dụng mô hình đã huấn luyện sẵn để dự đoán biên cạnh của ảnh chứa thực thể nguy trang. Các đường biên cạnh sau đó được thêm vào các ảnh gốc để tăng cường khả năng nhận diện bằng thị giác của chúng.

**Giai đoạn huấn luyện.** Dữ liệu huấn luyện đầu vào được kí hiệu là  $S = \{(X_n, Y_n), n = 1, \dots, N\}$ , với mỗi mẫu  $X_n = \{x_j^{(n)}, j = 1, \dots, |X_n|\}$  là ảnh đầu vào nguyên bản, và  $Y_n = \{y_j^{(n)}, j = 1, \dots, |X_n|\}, y_j^{(n)} \in \{0, 1\}$  là nhãn nhị phân thể hiện bản đồ nhãn biên cạnh của ảnh  $X_n$ . Chúng tôi bỏ qua chỉ số  $n$  để đơn giản hóa các ký hiệu vì mỗi ảnh được xem xét một cách độc lập. Các tham số của mỗi lớp trong mạng được kí hiệu là  $\mathbf{W}$ . Giả sử chúng ta có  $M$  lớp đầu ra phụ trong mạng (side-output layers). Mỗi lớp đầu ra phụ này được gán vào một bộ phân lớp với trọng số của mỗi bộ được kí hiệu là  $\mathbf{w} = (\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(M)})$ . Khi đó, hàm mất mát là **Công thức 3.1** sau đây:

$$\mathcal{L}_{\text{side}}(\mathbf{W}, \mathbf{w}) = \sum_{m=1}^M \alpha_m \ell_{\text{side}}^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}), \quad (3.1)$$

với  $\ell_{\text{side}}$  là hàm mất mát ở cấp độ ảnh của các đầu ra phụ. Trong quá trình huấn luyện, hàm mất mát được tính trên tất cả các điểm ảnh của ảnh đầu vào  $X = (x_j, j = 1, \dots, |X|)$  và ảnh nhãn biên cạnh  $Y = (y_j, j = 1, \dots, |X|), y_j \in \{0, 1\}$ . Với một tấm ảnh tự nhiên, phân bố của các điểm ảnh đại diện cho vùng biên cạnh và không biên cạnh thường chênh lệch lớn: khoảng 90% số điểm ảnh là vùng không thuộc biên cạnh [82]. Một hàm mất

mất đã được đề xuất [31] với tham số riêng được thêm vào để giải quyết vấn đề này. Hàm mất mát này được định nghĩa là "cost-sensitive loss" (nhạy cảm với chi phí).

Thay vì sử dụng hàm mất mát mới, chúng tôi sử dụng một chiến lược đơn giản hơn để cân bằng tự động sự mất mát giữa các lớp biên và không biên cạnh. Chúng tôi sử dụng một trọng số cân bằng lớp (class-balancing weight)  $\beta$  trên mỗi điểm ảnh. Chỉ số  $j$  được kí hiệu trên chiều không gian của ảnh  $X$ . Sau đó, chúng tôi sử dụng trọng số cân bằng lớp  $\beta$  này như một cách đơn giản để bù đắp sự mất cân bằng giữa các lớp biên và không biên cạnh. Cụ thể, chúng tôi định nghĩa hàm mất mát Cross-Entropy cân bằng lớp sau đây được sử dụng trong **Công thức 3.1**.

$$\begin{aligned} \ell_{\text{side}}^{(m)}(\mathbf{W}, \mathbf{w}^{(m)}) = & -\beta \sum_{j \in Y_+} \log \Pr(y_j = 1 | X; \mathbf{W}, \mathbf{w}^{(m)}) \\ & -(1 - \beta) \sum_{j \in Y_-} \log \Pr(y_j = 0 | X; \mathbf{W}, \mathbf{w}^{(m)}) \end{aligned} \quad (3.2)$$

với  $\beta = |Y_-|/|Y|$ , và  $1 - \beta = |Y_+|/|Y|$ .  $|Y_-|$  và  $|Y_+|$  kí hiệu tập nhãn biên cạnh và không biên cạnh.  $\Pr(y_j = 1 | X; \mathbf{W}, \mathbf{w}^{(m)}) = \sigma(a_j^{(m)}) \in [0, 1]$  được tính với công thức Sigmoid  $\sigma(\cdot)$  trên giá trị hàm kích hoạt tại điểm ảnh  $j$ . Tại mỗi lớp đầu ra phụ, chúng tôi thu được dự đoán biên cạnh  $\hat{Y}_{\text{side}}^{(m)} = \sigma(\hat{A}_{\text{side}}^{(m)})$ , với  $\hat{A}_{\text{side}}^{(m)} \equiv \{a_j^{(m)}, j = 1, \dots, |Y|\}$  là giá trị hàm kích hoạt tại lớp đầu ra phụ  $m$ .

Để sử dụng trực tiếp giá trị dự đoán tại các đầu ra phụ, chúng tôi sử dụng thêm một lớp kết hợp trọng số (weighted-fusion) và đồng thời học trọng số kết hợp trong suốt quá trình huấn luyện. Do đó, hàm mất mát cuối cùng tại lớp kết hợp  $\mathcal{L}_{\text{fuse}}$  trở thành **Công thức 3.3**.

$$\mathcal{L}_{\text{fuse}}(\mathbf{W}, \mathbf{w}, \mathbf{h}) = \text{Dist}(Y, \hat{Y}_{\text{fuse}}) \quad (3.3)$$

trong đó,  $\hat{Y}_{\text{fuse}} \equiv \sigma(\sum_{m=1}^M h_m \hat{A}_{\text{side}}^{(m)})$  với  $\mathbf{h} = (h_1, \dots, h_M)$  là trọng số kết hợp.  $\text{Dist}(\cdot, \cdot)$  là khoảng cách giữa bản đồ biên được dự đoán và nhãn, được chúng tôi sử dụng hàm mất mát Cross-Entropy để tính toán. Cuối cùng, mục tiêu của mô-đun khai thác đặc trưng biên cạnh là tối thiểu hóa hàm mất mát tại **Công thức 3.4** thông qua quá trình lan truyền

ngược sử dụng Stochastic Gradient Descent (SGD):

$$(\mathbf{W}, \mathbf{w}, \mathbf{h})^* = \operatorname{argmin}(\mathcal{L}_{\text{side}}(\mathbf{W}, \mathbf{w}) + \mathcal{L}_{\text{fuse}}(\mathbf{W}, \mathbf{w}, \mathbf{h})) \quad (3.4)$$

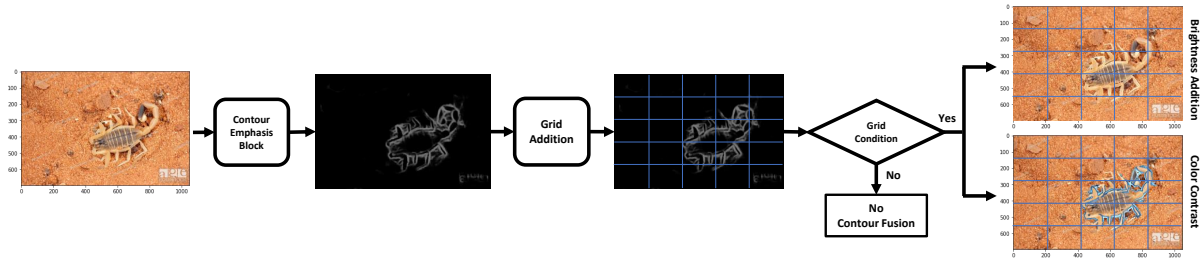
**Giai đoạn kiểm thử.** Trong giai đoạn kiểm thử, cho ảnh  $X$ , chúng tôi thu được bản đồ biên cạnh dự đoán từ các đầu ra phụ và lớp trọng số kết hợp như **Công thức 3.5**:

$$(\hat{Y}_{\text{fuse}}, \hat{Y}_{\text{side}}^{(1)}, \dots, \hat{Y}_{\text{side}}^{(M)}) = \text{CNN}(X, (\mathbf{W}, \mathbf{w}, \mathbf{h})^*), \quad (3.5)$$

với  $\text{CNN}(\cdot)$  là bản đồ biên cạnh dự đoán bởi kiến trúc mạng. Đầu ra biên cạnh cuối cùng thu được sau khi kết hợp các bản đồ biên cạnh dự đoán lại với nhau như trình bày tại **Công thức 3.6**.

$$\hat{Y}_{\text{HED}} = \text{Average}(\hat{Y}_{\text{fuse}}, \hat{Y}_{\text{side}}^{(1)}, \dots, \hat{Y}_{\text{side}}^{(M)}) \quad (3.6)$$

**Lưới điều kiện (Grid-Condition).** Sau khi thu được biên cạnh của ảnh ngụy trang đầu vào, chúng tôi cần kết hợp nó với ảnh đầu vào để cho ra ảnh được tăng cường biên cạnh thực thể ngụy trang. Theo đó, có nhiều cách thức để thực hiện kết hợp biên cạnh vào ảnh gốc. Do đó, chúng tôi lựa chọn theo kinh nghiệm hai phương pháp, là *Tương phản màu sắc (Color Contrast)* (1) và *Cộng độ sáng (Brightness Addition)* (2). Trong phương pháp Cộng độ sáng, chúng tôi trực tiếp cộng kết quả biên cạnh vào ảnh gốc (cộng giá trị theo từng điểm ảnh). Trong phương pháp Tương phản màu sắc, chúng tôi thực hiện cộng trên giá trị phần bù của điểm ảnh tại vị trí từng điểm ảnh tương ứng. Trong **Hình 3.4**, chúng ta có thể so sánh sự khác biệt về mặt thị giác giữa hai phương pháp hợp nhất biên cạnh được đề xuất. Có thể thấy, mô-đun phát hiện biên cạnh có thể thất bại trong các trường hợp ảnh quá phức tạp, khiến kết quả bị nhiễu hơn khi kết hợp với ảnh gốc. Để khắc phục một số trường hợp khó (intense cases) khi kết cấu của ảnh quá phức tạp, chúng tôi đề xuất một kỹ thuật đơn giản gọi là *Lưới điều kiện - Grid Condition* (**Hình 3.2**). Trong trường hợp này, chúng tôi áp dụng một lưới kích thước  $5 \times 5$  cho mỗi ảnh và quyết định không áp dụng hợp nhất biên cạnh với ảnh đó nếu số ô lưới vi phạm điều kiện quá một nửa (tương đương với khoảng 12 ô lưới). Điều kiện để một ô lưới bị loại bỏ nếu số điểm ảnh trong khu vực biên được phát hiện chiếm quá một nửa diện tích của ô lưới



HÌNH 3.2: Sơ đồ minh họa hoạt động của Lưới điều kiện (Grid-Condition) trong mô-đun Tăng cường đặc trưng biên cạnh (Contour Emphasis).

đó. Đề xuất này có thể thích ứng với mọi kích thước ảnh của tập dữ liệu ngựa trang. Nhờ vào Lưới điều kiện, chúng tôi có thể đưa ra quyết định có hoặc không áp dụng biên cạnh để tăng cường nhận biết thực thể ngựa trang. Từ đó, mô hình có thể loại trừ được những trường hợp thất bại trong việc phát hiện biên cạnh, ảnh hưởng tiêu cực đến độ chính xác phân đoạn cuối cùng của mô hình đề xuất.

### 3.2.3 Khối Transformer phân đoạn thực thể ngựa trang

**Bộ trích xuất đặc trưng - Feature Extractor.** Cho trước một ảnh đầu vào  $I \in \mathbb{R}^{H \times W \times 3}$ , chúng tôi sử dụng một mạng nơ-ron học sâu để trích xuất đặc trưng tại các kích thước khác nhau  $C_{2-5}$ . Đầu vào cho khối Transformer được làm phẳng (flatten) với các đặc trưng  $C_{3-5}$  trong khi đặc trưng  $C_2$  được đưa trực tiếp vào mô-đun kết hợp ở giai đoạn tiếp theo nhằm bổ sung đặc trưng như một kết nối tắt (skip connection). Chi tiết về các kiến trúc mô hình cơ sở (backbone) được trình bày thông qua thực nghiệm loại suy ở **Phần 3.3**.

**Mô hình phân đoạn thực thể dựa trên kiến trúc Transformer.** Chúng tôi sử dụng cấu trúc của mô hình Mã hóa-Giải mã [59, 73] dựa trên Transformer vì các công trình trước đó về Transformer đã chứng minh hiệu quả của các lớp Self-Attention trong việc trích xuất thông tin toàn cục của hình ảnh. Trong kiến trúc này, chúng tôi tập trung vào xác định chính xác vị trí của thực thể, đây là một vấn đề quan trọng hàng đầu trong tác vụ phân đoạn thực thể. Đáng chú ý rằng, đầu vào của khối Transformer này trong kiến trúc của chúng tôi là đặc trưng được tổng hợp với nhiều tỷ lệ khác nhau, so với việc sử dụng một tỷ lệ cố định như mô hình DETR [4].

**Kiến trúc Location-sensing Transformer.** Mặc dù kiến trúc Transformer có thể trích xuất tốt đặc trưng toàn cục nhờ vào các lớp self-attention, mô hình vẫn cần một lượng lớn mẫu huấn luyện với chi phí tính toán cao. Tuy nhiên, bài toán nghiên cứu trên thực thể nguy trang thường có số lượng mẫu huấn luyện ít, mục tiêu của chúng tôi là sử dụng một kiến trúc hiệu quả có thể hội tụ nhanh và đạt được độ chính xác cao.

**Bộ mã hóa LST.** Không giống như DETR [4] với một kích thước đặc trưng cấp thấp duy nhất ở bộ mã hóa, kiến trúc bộ mã hóa LST mà chúng tôi sử dụng có thể nhận đa kích thước đặc trưng  $X_m$  để làm giàu thông tin. Học hỏi từ lớp deformable self-attention [89] với khả năng nắm bắt thông tin cục bộ và tăng cường liên kết giữa các cụm (token) lân cận, chúng tôi sử dụng toán tử tích chập kết hợp, gọi là blend-convolution feed-forward network (BC-FFN). Đầu tiên, vector đặc trưng được tái tạo từ chiều không gian (spatial dimension) phụ thuộc vào kích thước  $C_i$ . Sau đó là một lớp tích chập  $3 \times 3$ . Cuối cùng là lớp chuẩn hóa theo nhóm group normalization (GN) và hàm kích hoạt GELU. Sau lớp tích chập  $3 \times 3$ , đặc trưng được duỗi thẳng (flatten). Mạng BC-FFN không chứa các thành phần MLP và kết nối dư so với các kiến trúc phức tạp khác.

Với đặc trưng đầu vào  $X_b$ , quá trình xử lý qua mạng BC-FFN biểu diễn như **Công thức 3.7** sau:

$$X'_b = \text{Conv}^3(\text{GELU}(\text{GN}(\text{Conv}^3(X_b))))), \quad (3.7)$$

với  $\text{Conv}^3$  là tích chập  $3 \times 3$ . Lớp mã hóa LST được mô tả như **Công thức 3.8** sau:

$$X_e = \text{BC-FFN}(\text{LN}((X_m + P_m) + \text{MDAttn}(X_m + P_m))), \quad (3.8)$$

với  $P_m$  là mã hóa vị trí (positional encodings). MDAttn và LN là Multi-head Deformable Self-attention và lớp chuẩn hóa.

**Bộ giải mã LST.** Bộ giải mã LST dùng để giải mã với các đặc trưng toàn cục được tạo ra bởi bộ mã hóa LST và các truy vấn theo vị trí để tạo ra các đặc trưng nhận biết thực thể. Mã hóa vị trí không gian cũng được thêm vào các truy vấn theo vị trí  $Q_L$  và bộ nhớ mã hóa  $X_e$ . Sau đó, chúng được kết hợp bởi lớp deformable cross-attention. Khác với bộ giải mã biến đổi thông thường, chúng tôi sử dụng trực tiếp lớp deformable cross-attention mà



không cần self-attention vì các truy vấn được đề xuất đã chứa các đặc trưng toàn cục có thể học được. BC-FFN cũng được sử dụng sau các phép toán deformable cross-attention, tương tự như bộ mã hóa LST. Cho trước truy vấn theo vị trí (location-guided queries)  $Q_L$ , quá trình giải mã LST được mô tả như **Công thức 3.9** sau:

$$X_d = \text{BC-FFN}(\text{LN}((Q_L + P_s) + \text{MDCAttn}((Q_L + P_s), (X_e + P_m)))), \quad (3.9)$$

với  $P_s$  là mã hóa vị trí dựa trên đặc trưng ô lưới. MDCAttn là toán tử multi-head deformable cross-attention.  $X_d$  là đặc trưng đầu ra biểu diễn theo từng thực thể. Cuối cùng,  $X_d$  được đưa vào mô-đun DCIN để dự đoán mặt nạ phân đoạn.

**Mô-đun tổng hợp đặc trưng.** Trong mô-đun tổng hợp này, chúng tôi học hỏi công trình OSFormer [59] với hai mô-đun chính đó là mô-đun Chuẩn hóa Thực thể Ngụy trang Động (Dynamic Camouflaged Instance Normalization - DCIN) và mô-đun Tổng hợp từ thô-đến-mịn (Coarse-to-Fine Fusion - CFF). Đặc trưng  $C_2$  và các đặc trưng trung gian của T-Encoder được gửi đến mô-đun CFF để tạo ra các đặc trưng toàn cục. Sau đó, các đặc trưng từ lớp cuối của khối Transformer và mô-đun CFF là đầu vào của mô-đun DCIN. Trong mô-đun DCIN, có một lớp kết nối đầy đủ được sử dụng để trả về nhãn vị trí. Đồng thời, một lớp multi-layer perception được sử dụng để tạo ra các tham số nhận biết thực thể (instance-aware parameters). Các tham số này sau đó được sử dụng để thiết lập mặt nạ phân đoạn thực thể. Chi tiết cài đặt có thể được tham chiếu tới công trình [59] này.

Là một mô hình dựa trên kiến trúc transformer bottom-up, mô hình nỗ lực để sử dụng các đặc trưng toàn cục từ nhiều cấp độ từ bộ mã hóa LST để tạo ra một biểu diễn đặc trưng mặt nạ tổng quát. Để kết hợp các thông tin ngữ cảnh đa dạng, chúng tôi cũng kết hợp đặc trưng cấp thấp  $C_2$  từ backbone CNN như một đặc trưng tăng cường để tạo ra một bản đồ đặc trưng có độ phân giải cao thống nhất  $F \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times D}$ . Chúng tôi lấy các đặc trưng nhiều cấp độ  $C_2, T_3, T_4$ , và  $T_5$  làm đầu vào cho kết hợp xếp tầng (cascade fusion). Bắt đầu từ  $T_5$  ở tỷ lệ 1/32 của đầu vào, một lớp tích chập  $3 \times 3$ , GN, và upsampling  $2 \times$  được truyền qua và cộng với đặc trưng có độ phân giải cao hơn ( $T_4$  với tỷ lệ 1/16). Sau khi kết hợp  $C_2$  với tỷ lệ 1/4, đặc trưng tiếp tục qua một lớp tích chập  $1 \times 1$ , GN, và các

phép toán RELU để tạo ra đặc trưng phân đoạn  $F$ . Lưu ý rằng mỗi đặc trưng đầu vào giảm số chiều từ 256 xuống 128 sau tích chập đầu tiên và sau đó được tăng lên 256 chiều ở kết quả cuối cùng.

Nhận thấy rằng các đặc trưng biên cạnh của thực thể nguy trang khó nắm bắt hơn, chúng tôi thiết kế một mô-đun chú ý biên ngược (REA, reverse edge attention) được nhúng trong CFF để giám sát các đặc trưng biên cạnh trong quá trình học. Khác với các công trình trước đây về chú ý biên ngược [7, 13], REA hoạt động trên các đặc trưng biên thay vì các mặt nạ nhị phân dự đoán biên cạnh. Ngoài ra, các nhãn biên được sử dụng để giám sát được thu được bằng cách giảm biên (erosion) các nhãn mặt nạ thực thể mà không cần gán nhãn thủ công. Lấy cảm hứng từ Convolutional Block Attention [79], các đặc trưng đầu vào được thực hiện bởi cả phép gộp trung bình (*AvgPool*) và gộp cực đại (*MaxPool*). Sau đó, chúng tôi nối và chuyển tiếp chúng đến một lớp tích chập  $7 \times 7$  và một hàm sigmoid. Sau đó, chúng tôi đảo ngược trọng số chú ý và áp dụng chúng cho đặc trưng kết hợp  $F_f$  bằng phép nhân theo từng phần tử. Cuối cùng, chúng tôi sử dụng một lớp tích chập  $3 \times 3$  để dự đoán đặc trưng biên. Giả sử rằng đặc trưng đầu vào là  $T_i$ , toàn bộ quá trình của mỗi mô-đun REA có thể được biểu diễn như Công thức 3.10 sau:

$$F_e = \text{Conv}^3(F_f \otimes (1 - \text{Sigmoid}(\text{Conv}^7([\text{AvgPool}(T_i); \text{MaxPool}(T_i)]))))), \quad (3.10)$$

với  $\text{Conv}^7$  là lớp tích chập  $7 \times 7$  và  $[\cdot; \cdot]$  thể hiện kết nối theo chiều sâu (channel axis). Tóm lại, CFF tạo ra đặc trưng phân đoạn  $F$  để đưa qua lớp chuẩn hóa cuối cùng DCIN và dự đoán nhãn mặt nạ phân đoạn thực thể nguy trang.

## 3.3 Thực nghiệm

### 3.3.1 Cấu hình thực nghiệm.

Trong các thí nghiệm của chúng tôi, các tập dữ liệu COD10K [12], NC4K [49] và CAMO++ [36] được sử dụng để đánh giá bài toán phân đoạn thực thể nguy trang. So sánh chi tiết các thuộc tính của những tập dữ liệu nói trên được chúng tôi cung cấp tại

**Bảng 2.1** thuộc **Phần 2**. Đây là những tập dữ liệu chuẩn và phổ biến trong lĩnh vực nghiên cứu trên thực thể ngẫu trang.

Để lựa chọn các mô hình cơ sở (backbone), chúng tôi sử dụng ResNet-50 [26], ResNet-50 [26] (với kích thước đầu vào là  $550 \times 550$  mô phỏng cấu hình thời gian thực), ResNet-101 [26], Pyramid Vision Transformer (PVT) [76], và Swin Transformer (Swin-T) [46] để áp dụng trên phương pháp đề xuất của chúng tôi. CE-OST được xây dựng trên nền tảng Detectron2 [80] và các mô hình khác được cài đặt hoặc tham khảo từ các công bố của chính tác giả của các công trình đó. Mô hình CE-OST của chúng tôi tuân theo các giá trị tham số, siêu tham số gốc của mô hình OSFormer [59]. Cụ thể, chúng tôi sử dụng một GPU GeForce RTX 2080Ti và huấn luyện với Stochastic Gradient Descent. Để khởi tạo các mô hình, chúng tôi sử dụng các trọng số được huấn luyện sẵn trên ImageNet [10]. Quá trình huấn luyện của chúng tôi diễn ra với 90K lần lặp với kích thước batch là 1 và tốc độ học cơ sở là  $lr = 2.5 \times 10^{-4}$  khởi đầu với 1K lần lặp. Chúng tôi cũng sử dụng cơ chế giảm tốc độ học 0.1 sau 60K và 80K lần lặp. Các giá trị *learningrate decay* và *momentum* lần lượt là  $1 \times 10^{-4}$  và  $9 \times 10^{-1}$ .

**Các độ đo đánh giá.** Để trình bày kết quả mô hình, chúng tôi sử dụng độ chính xác trung bình (AP). Chi tiết hơn, chúng tôi sử dụng AP, AP@50, và AP@75. Chi tiết về các độ đo này có thể được tham khảo tại <https://cocodataset.org/#detection-eval>.

### 3.3.2 Kết quả thực nghiệm

**So sánh với các mô hình tiên tiến** Để chứng minh độ chính xác của phương pháp đề xuất của chúng tôi - CE-OST, chúng tôi đã tiến hành các thí nghiệm được thiết lập trong **Bảng 3.1**. Với cấu hình này, chúng tôi tận dụng kết quả trên các tập dữ liệu COD10K và NC4K được công bố để so sánh độ chính xác giữa các mô hình. Chúng tôi không sử dụng tập dữ liệu CAMO++ [36] trong cấu hình này vì các kết quả không được cung cấp bởi các tác giả của những công trình đó. Cụ thể, chúng tôi sử dụng các mô hình thuộc nhánh hai giai đoạn (như [2, 5, 6, 28, 30, 33]) và nhánh một giai đoạn (như [1, 17, 23, 59, 71, 78]). Để so sánh công bằng, chúng tôi sử dụng backbone ResNet-101 [26] làm backbone chung của các mô hình. Kết quả thực nghiệm đã cho thấy sự cải thiện của chúng tôi trên

BẢNG 3.1: So sánh với các mô hình tiên tiến nhất trên tập dữ liệu COD10K [12] và NC4K [49] (cùng sử dụng mô hình cơ sở ResNet-101 [26])

Method		#Params	#GFLOPs	COD10K			NC4K		
				AP	AP50	AP75	AP	AP50	AP75
Two-Stage	Mask R-CNN [28]	62.9M	254.5	28.7	60.1	25.7	36.1	68.9	33.5
	MS R-CNN [30]	79.0M	251.1	33.3	61.0	32.9	35.7	63.4	34.7
	Cascade R-CNN [2]	90.7M	386.7	29.5	61.0	25.9	34.6	66.3	31.5
	HTC [6]	95.9M	384.3	30.9	61.0	28.7	34.2	64.5	31.6
	BlendMask [5]	54.7M	302.8	31.2	60.0	28.9	31.4	61.2	28.8
	Mask Transfuser [33]	63.3M	253.7	31.2	60.7	29.8	34.0	63.1	32.6
One-Stage	YOLACT [1]	-	-	29.0	60.1	25.3	37.8	70.6	35.6
	CondInst [71]	53.1M	269.1	34.3	67.9	31.6	38.0	71.1	35.6
	QueryInst [17]	-	-	32.5	65.1	28.6	38.7	72.1	37.6
	SOTR [23]	82.1M	549.6	32.0	63.6	29.2	34.3	65.7	32.4
	SOLO [78]	65.1M	394.6	35.2	65.7	33.4	37.8	69.2	36.1
	OSFormer [59]	65.5M	398.2	42.0	71.3	42.8	44.4	73.7	45.1
	<b>CE-OST (Ours)</b>	80.2M	523.2	<b>43.2</b>	<b>72.2</b>	<b>44.1</b>	<b>45.1</b>	<b>74.0</b>	<b>46.4</b>

tất cả các độ đo đánh giá AP, AP50 và AP75. Trên COD10K [12], chúng tôi đạt được 43.2%, 72.2%, và 44.1% tương ứng về AP, AP50 và AP75. Trên NC4K [49], ba giá trị tương ứng là 45.1%, 74.0%, và 46.4%. Theo đó, chúng tôi đạt kết quả tiên tiến nhất so với các phương pháp trên cả hai nhánh hai giai đoạn và một giai đoạn. Chúng tôi cũng tiến hành thực nghiệm loại suy trên mô-đun Tăng cường đặc trưng biên cạnh của mình ở phần tiếp theo.

### 3.3.3 Thực nghiệm loại suy.

Trong mô hình đề xuất CE-OST, phương pháp Tăng cường đặc trưng biên cạnh có thể được áp dụng theo hai cách. Trong Bảng 3.2, chúng tôi trình bày hiệu quả của hai phương pháp này, gồm có Tương phản màu sắc (Color Contrast) và Cộng độ sáng (Brightness Addition). Chúng tôi cũng tiến hành các thí nghiệm trên các mô hình cơ sở khác nhau bao gồm 5 phương pháp đã đề cập trong phần Cấu hình thực nghiệm. Nhìn chung, các mô hình dựa trên Transformer như Swin-T hay PVT cho kết quả tốt nhất trong số các phương pháp được thực nghiệm. CAMO++ [36] là tập dữ liệu khó nhất, theo sau là NC4K [49] và COD10K [12]. Các thử nghiệm này cũng chứng minh tính tổng quát của mô hình đề xuất CE-OST trên các mô hình cơ sở khi cải thiện gần như mọi kết quả so với các phương pháp tiên tiến nhất. Đặc biệt đối với tập dữ liệu CAMO++ [36], backbone PVT giữ độ chính xác ổn định tốt nhất. Kết quả có thể được giải thích dựa trên bộ trích

BẢNG 3.2: Thử nghiệm loại suy về các mô hình backbone của CE-OST trên tập dữ liệu COD10K [12], NC4K [49], và CAMO++ [36].

Phương pháp	Mô hình cơ sở	COD10K			NC4K			CAMO++		
		AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
<b>OSFormer</b>	ResNet-50 [26]	41.0	71.1	40.8	42.5	72.5	42.3	19.0	33.8	18.3
	ResNet-50-550 [26]	-	-	-	-	-	-	20.1	36.3	19.3
	ResNet-101 [26]	42.0	71.3	42.8	44.4	73.7	45.1	20.6	34.4	20.2
	PVTv2-B2-Li [76]	47.2	74.9	49.8	-	-	-	27.7	44.7	27.9
	Swin-T [46]	47.7	78.6	49.3	-	-	-	22.3	36.6	21.8
<b>CE-OST (Color Contrast)</b>	ResNet-50 [26]	41.6	70.7	42.3	42.4	71.4	42.6	20.1	34.2	19.6
	ResNet-50-550 [26]	35.9	65.2	34.3	41.1	70.9	41.1	20.6	35.7	20.0
	ResNet-101 [26]	43.2	72.2	44.1	45.1	74.0	46.4	21.7	36.6	21.3
	PVTv2-B2-Li [76]	48.4	75.7	51.3	51.4	77.9	55.0	28.5	45.3	29.9
	Swin-T [46]	49.1	78.0	52.1	50.5	78.9	53.1	22.7	37.6	22.4
<b>CE-OST (Brightness Addition)</b>	ResNet-50 [26]	41.2	69.0	41.6	42.4	71.1	42.9	20.2	34.8	19.5
	ResNet-50-550 [26]	35.9	65.2	34.6	40.8	71.1	40.3	21.0	37.1	20.3
	ResNet-101 [26]	42.4	70.8	43.7	44.2	73.1	45.0	21.1	34.4	20.9
	PVTv2-B2-Li [76]	47.9	74.6	50.5	51.1	77.3	54.9	27.9	45.1	29.2
	Swin-T [46]	49.0	78.5	51.4	50.8	79.3	53.9	22.7	38.4	23.1
<b>CE-OST (Brightness &amp; Contrast)</b>	ResNet-50 [26]	41.8	70.5	42.0	43.4	72.3	44.0	21.0	35.9	20.7
	ResNet-50-550 [26]	36.1	65.8	35.1	41.4	71.8	41.4	20.5	36.6	20.0
	ResNet-101 [26]	42.7	71.2	43.9	45.1	74.0	46.5	21.4	35.4	20.8
	PVTv2-B2-Li [76]	48.3	75.4	51.4	50.8	77.1	54.7	27.6	44.7	28.5
	Swin-T [46]	48.9	78.4	51.3	50.6	79.2	53.3	23.3	38.2	24.1

\*Kết quả tốt nhất, thứ hai và thứ ba lần lượt được kí hiệu với màu đỏ, xanh dương, và xanh lá.

xuất đặc trưng ở nhiều tỷ lệ của PVT có thể xử lý tốt các hình ảnh với nhiều tỷ lệ khác nhau của CAMO++. Trong Hình 3.3, chúng tôi trình bày kết quả tốt nhất của chúng tôi khi sử dụng kiến trúc backbone PVT (bên trái) và một số trường hợp thất bại (bên phải). Các trường hợp xấu xảy ra với hiện tượng phân đoạn quá mức (over-segmentation) hoặc dự đoán nhãn sai (mislabeling).

Các trường hợp mô hình dự đoán sai gây ra bởi kích thước của các thực thể và sự phức tạp của vùng nền trong các ảnh nguy trang. Có thể thấy, khi vùng thực thể chiếm phần quá lớn hoặc quá nhỏ so với diện tích ảnh đầu vào, mô hình chưa nắm bắt tốt thông tin ngữ nghĩa của thực thể và dẫn đến các vấn đề vừa nêu. Bên cạnh đó, việc hình ảnh nguy trang có vùng nền quá phức tạp hoặc tương đồng với vùng thực thể cũng gây khó khăn cho mô hình phân đoạn trong việc phân biệt giữa chúng. Để cải thiện vấn đề này, chúng tôi cần cung cấp thêm thông tin cho mô hình học. Chúng tôi dự kiến thiết kế một kiến trúc hoặc gắn thêm một mô-đun để cung cấp thêm thông tin ngữ nghĩa cho mô hình phân đoạn. Ngoài ra, các phương pháp tăng cường ảnh cũng có thể được xem xét để giải quyết vấn đề này.

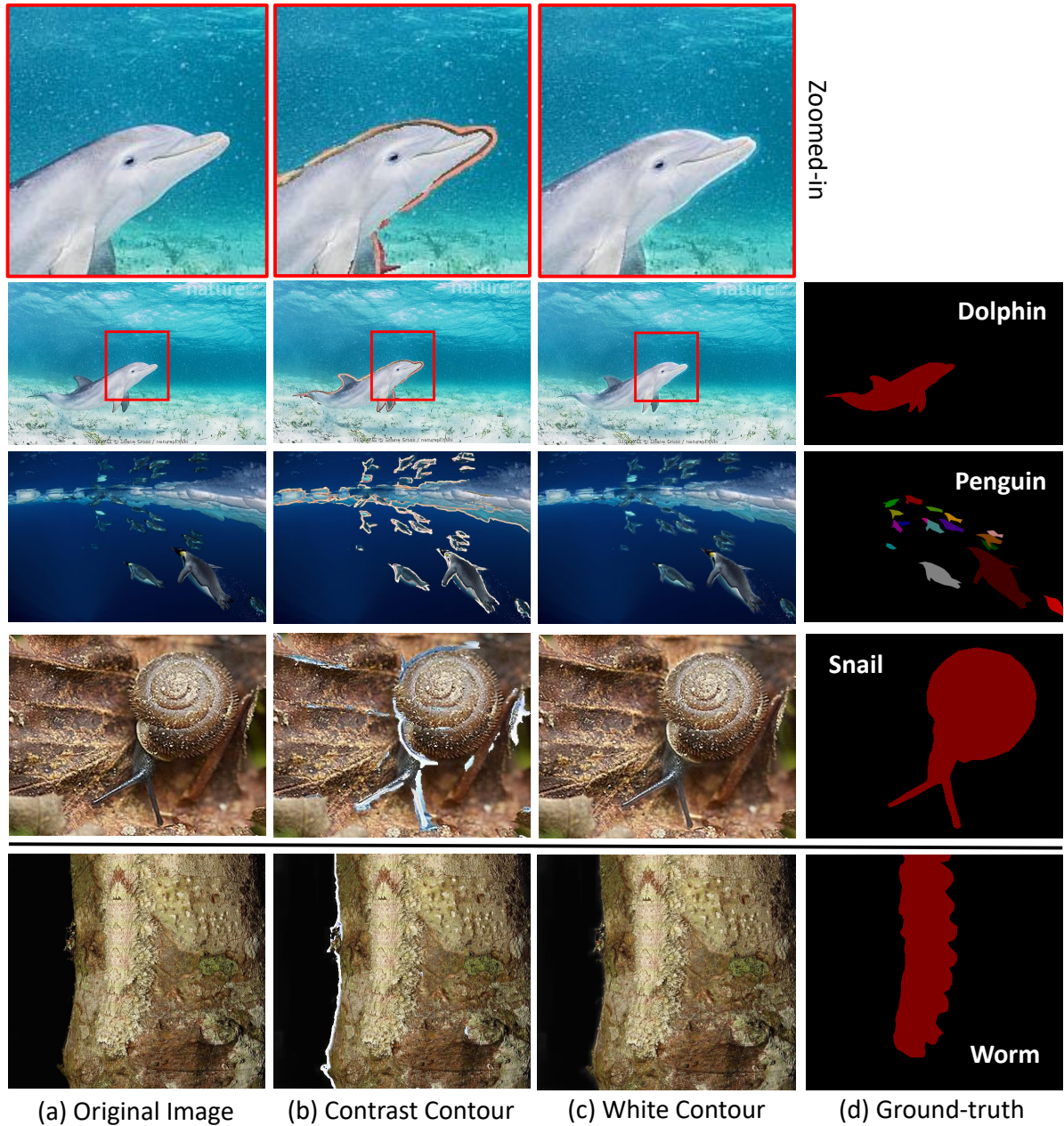




HÌNH 3.3: Trực quan hóa kết quả trên tập dữ liệu CAMO++ [36] với mô hình CE-OST sử dụng backbone PVT. Ngưỡng tin cậy được sử dụng là 0.5.

**Bàn luận.** Trong Hình 3.4, chúng tôi trình bày một số mẫu về các thực thể nguy trang với biên cạnh được tăng cường. Từ trái sang phải, chúng tôi sắp xếp ảnh gốc (a), ảnh với biên cạnh tương phản (b), ảnh với biên cạnh được cộng mức sáng (c) và nhãn phân đoạn (d). Cả hai loại biên cạnh đều được tạo ra bởi phương pháp tăng cường biên cạnh của mô-đun đề xuất CE, và chúng tôi trực quan các đường biên cạnh này dưới hai hình thức như trên. Các đường biên cộng mức sáng là kết quả của việc cộng giá trị mức sáng vào các điểm ảnh thuộc đường biên của các thực thể. Trong khi đó, các đường biên tương phản thay đổi các giá trị màu sang một giá trị bù trong phạm vi tương phản, mang lại một đường biên có thể được phân biệt tốt hơn so với các đường viền màu sáng. Do đó,

chúng ta có thể quan sát kết quả tốt nhất chủ yếu thuộc về phương pháp Tương phản màu sắc. Hàng cuối cùng minh họa một trường hợp khi biên cạnh được phát hiện không thành công, dẫn đến tăng cường sai đối tượng cần phân đoạn. Trong tương lai của chúng tôi sẽ tập trung vào việc cải thiện nhận diện các đường biên cạnh của thực thể nguy trang.



HÌNH 3.4: Một số hình ảnh minh họa cho ảnh đầu vào được tăng cường đặc trưng biên cạnh, dữ liệu từ tập CAMO++ [36]. Dòng đầu tiên thể hiện các vùng được phóng đại.



### 3.4 Tạm kết

Trong phần này, chúng tôi đã đề xuất mô hình CE-OST - một phương pháp Tăng cường đặc trưng biên cạnh cho mô hình Transformer một giai đoạn để giải quyết bài toán phân đoạn thực thể trên các ảnh nguy trang. Chúng tôi đã chứng minh sự cải thiện của phương pháp đề xuất trên ba tập dữ liệu nguy trang chuẩn là COD10K, NC4K và CAMO++. Xây dựng trên nền tảng mô hình OSFormer [59], một trong những kiến trúc một giai đoạn tiên tiến gần đây cho bài toán phân đoạn thực thể nguy trang, chúng tôi đã cải thiện độ chính xác của mô hình phân đoạn nhờ việc khai thác hiệu quả các đặc trưng biên cạnh vật thể nguy trang. Kết quả này chứng minh lập luận của chúng tôi khi nhận định rằng, độ chính xác dự đoán của mô hình phân đoạn có thể được cải thiện khi tăng cường khả năng nhận ra các thực thể nguy trang thông qua tăng cường đặc trưng phân biệt tại vùng biên thực thể. Các đề xuất trong phần này được tổng hợp thành công trình khoa học và đã được công bố tại hội nghị quốc tế MAPR2023 - 2023 International Conference on Multimedia Analysis and Pattern Recognition [CT1]. Trong tương lai, chúng tôi dự định mở rộng ý tưởng của mình sang các lĩnh vực đặc thù khác như ảnh y khoa, nơi các thực thể mang các đặc điểm khó nhận biết của đối tượng nguy trang.

## Chương 4

# Mô hình FS-CDIS học đặc trưng phân biệt với ít mẫu dữ liệu

### 4.1 Tổng quan

Trong phần này, chúng tôi giải quyết thách thức về sự giới hạn trong dữ liệu gán nhãn để huấn luyện các mô hình phân đoạn trên thực thể nguy trang. Chúng tôi trình bày đề xuất phân đoạn thực thể nguy trang với hướng tiếp cận sử dụng ít dữ liệu huấn luyện thông qua một mô hình hai giai đoạn FS-CDIS (Few-shot Camouflaged Detection and Instance Segmentation). Với phương pháp học ít dữ liệu, chúng ta có thể thực hiện các nhiệm vụ học máy với số lượng dữ liệu hạn chế cho trước. Phương pháp học ít dữ liệu yêu cầu hai giai đoạn xử lý: (1) giai đoạn huấn luyện cơ sở (base phase) để mô hình có được kiến thức tổng quát thông qua nhiều dữ liệu huấn luyện, và sau đó (2) thực hiện giai đoạn tinh chỉnh (novel phase) để mô hình tương thích với các tác vụ chi tiết (downstream task). Với bài toán phân đoạn thực thể nguy trang, chúng tôi sử dụng phương pháp học ít dữ liệu áp dụng trên ảnh chứa động vật nguy trang, đây có thể là những loài động vật quý hiếm và khó tìm thấy trong tự nhiên, hoặc đơn giản là những loài động vật có tập tính nguy trang để lẩn trốn kẻ thù. Chính những tập tính lẩn trốn này khiến cho dữ liệu thu thập về động vật nguy trang trong tự nhiên trở nên giới hạn hơn. Thông qua hướng tiếp cận học với ít dữ liệu về đối tượng nguy trang, các mô hình vẫn có thể xử lý tốt các nhiệm vụ cho trước. Mặc dù phát hiện và phân đoạn thực thể nguy trang có nhiều ứng

dụng thực tiễn, các nghiên cứu công bố trước đây vẫn chưa đề cập đến việc thực hiện tác vụ này với ngữ cảnh ít dữ liệu huấn luyện. Trong khi đó, đây là một hướng tiếp cận tiềm năng và phù hợp với tính chất của bài toán đặc thù này. Do đó, chúng tôi muốn giải quyết bài toán phân đoạn và phát hiện động vật nguy trang ở cấp độ thực thể dựa trên ít mẫu dữ liệu huấn luyện.

Các đóng góp của chúng tôi về hướng tiếp cận học ít dữ liệu cho bài toán phát hiện và phân đoạn thực thể nguy trang được công bố trong công trình [CT2, CT3]. Cụ thể, chúng tôi có hai đóng góp chính sau:

- Một là, chúng tôi xây dựng một tập dữ liệu chuẩn, đặt tên là CAMO-FS, dựa trên nền tảng của tập CAMO++ [36], đây là một trong những tập dữ liệu đầu tiên cho ngữ cảnh ít dữ liệu trên thực thể động vật nguy trang.
- Hai là, chúng tôi đề xuất mô hình FS-CDIS, để phân đoạn hiệu quả thực thể động vật nguy trang. Trong đó, hai điểm chính trong mô hình chứa đựng sự cải tiến cho thực thể nguy trang là hàm mất mát ba thành phần ở cấp độ thực thể (Instance Triplet Loss) và bộ nhớ lưu trữ thực thể (Instance Memory Storage).

## 4.2 Bộ dữ liệu đề xuất CAMO-FS

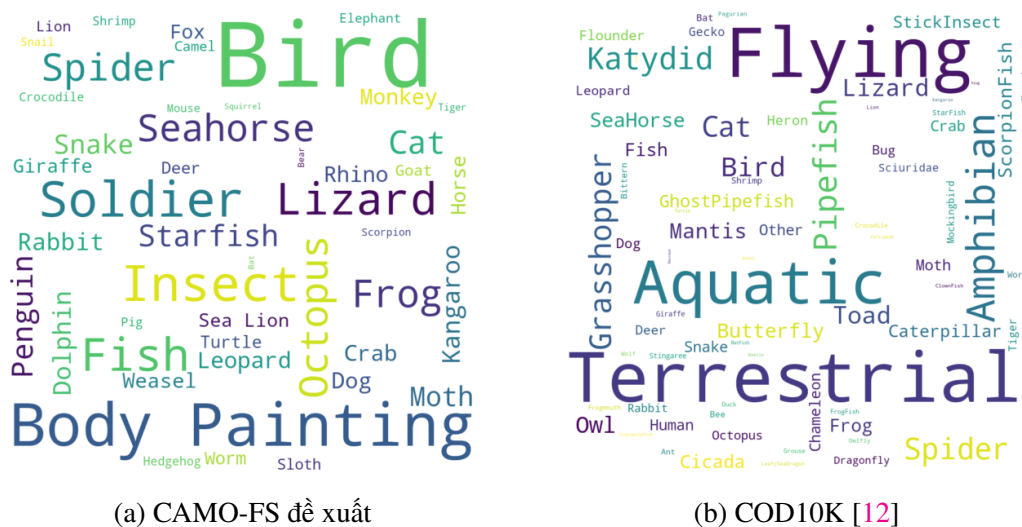
Trong thực tế, việc thu thập dữ liệu hình ảnh về các thực thể nguy trang gặp nhiều khó khăn hơn so với việc thu thập dữ liệu của các đối tượng thông thường khác. Đặc biệt, với bài toán phân đoạn ở cấp độ thực thể, dữ liệu sau khi thu thập cần được gán nhãn chính xác đến cấp độ điểm ảnh, điều này phần nào làm tăng thêm độ khó cho công tác làm dữ liệu. Nguy trang là đặc tính được mô tả ở các loài động vật giúp chúng hòa lẫn vào môi trường xung quanh để tránh kẻ thù nguy hiểm [36]. Thực thể nguy trang là các thực thể có tính chất đặc thù này với màu sắc, chất liệu tương đồng với môi trường khiến chúng khó bị phát hiện. Ranh giới giữa những điểm ảnh thuộc thực thể nguy trang và điểm ảnh thuộc vùng nền hoặc giữa các thực thể nguy trang với nhau, trong nhiều trường hợp bị chồng lấp phần nào. Điều này khiến cho quá trình gán nhãn ở cấp độ điểm ảnh trở nên mơ hồ, việc xác minh các vùng chứa thực thể nguy trang trở nên khó khăn hơn.

Mặt khác, việc huấn luyện mô hình với kỹ thuật học ít dữ liệu cũng yêu cầu một tổ chức dữ liệu khác biệt, cụ thể, dữ liệu cần được chia thành các tập con với số lượng mẫu phù hợp với quá trình học. Vì thế, để dễ dàng chuẩn bị dữ liệu cho tác vụ phân đoạn thực thể nguy trang dựa trên hướng tiếp cận sử dụng ít dữ liệu huấn luyện mà đề tài này nhắm đến, một trong những cách phổ biến nhất để tạo lập dữ liệu chính là kế thừa từ các tập dữ liệu nguy trang chuẩn (benchmark) hiện có. CAMO++ [36] là tập dữ liệu được chúng tôi lựa chọn vì đây là bộ dữ liệu có kích thước lớn, độ đa dạng về số lượng lớp ngữ nghĩa và lớp tổng quát cao, nhãn phân đoạn thực thể đầy đủ và mới được công bố gần đây.

**Đặc trưng của bộ dữ liệu CAMO++ [36].** CAMO++ chứa 5,500 hình ảnh nguy trang và không nguy trang tương ứng với 32,756 nhãn đối tượng [36]. Bộ dữ liệu có 93 lớp chi tiết được gán cho 13 lớp tổng quát. Tuy nhiên, nếu chỉ tính riêng thực thể nguy trang, CAMO++ có 47 lớp chi tiết được thiết kế với cấu trúc phân cấp và được phân thành 10 lớp tổng quát. Nói cách khác, CAMO++ đóng góp 2,695 hình ảnh nguy trang bao gồm 1,250 hình ảnh nguy trang hiện có trong bộ dữ liệu CAMO trước đó cùng với 1,450 hình ảnh nguy trang mới được thu thập thêm cho phiên bản CAMO++. Trong phạm vi luận văn này, chúng tôi không sử dụng 2,800 hình ảnh không nguy trang của CAMO++. Về mức độ tổng quát, CAMO++ cung cấp đầy đủ nhãn cho các tác vụ phân loại, phát hiện đối tượng và phân đoạn thực thể trên các ảnh nguy trang, phù hợp để thực hiện mục tiêu nghiên cứu của chúng tôi trong tác vụ phân đoạn thực thể nguy trang.

**Bộ dữ liệu đề xuất CAMO-FS.** Chúng tôi xây dựng CAMO-FS bằng cách kế thừa cấu trúc phân loại dựa trên đặc trưng sinh học của CAMO++, giản lược quá trình thu thập dữ liệu về thực thể nguy trang. **Bảng 2.1** cung cấp phân tích tổng quan về các công trình trước đây được thực hiện về đối tượng nguy trang và so sánh với CAMO-FS của chúng tôi với một số tiêu chí như: số lượng ảnh, số lượng gán nhãn, số lượng phân lớp và đặc trưng các loại nhãn cho các tác vụ khác nhau. Chúng tôi kế thừa CAMO++ [36] với 10 lớp tổng quát cho đối tượng nguy trang để xây dựng cấu trúc cho bài toán phân đoạn thực thể với ít mẫu dữ liệu. Đặc biệt, về khía cạnh cấu trúc phục vụ bài toán học với ít mẫu dữ liệu, CAMO-FS không chỉ là bộ dữ liệu đầu tiên về đối tượng nguy trang mà còn giữ tỉ lệ thực thể/ảnh ở mức 1.172, cao nhất trong số các bộ dữ liệu được khảo

sát. Lưu ý rằng, các tập dữ liệu khác tuy có số lượng ảnh lớn, nhưng không phải tất cả ảnh đều chứa thực thể nguy trang. Hình 4.1 minh họa sự phân bố theo lớp của bộ dữ liệu CAMO-FS và COD10K [12].



HÌNH 4.1: Phân phối lớp của các mẫu dữ liệu nguy trang dưới dạng word-cloud giữa tập dữ liệu CAMO-FS đề xuất và COD10K [12].

BẢNG 4.1: Số lượng mẫu thu thập thêm cho bộ dữ liệu đề xuất CAMO-FS.

Lớp đối tượng	Bat	Bear	Camel	Dolphin	Elephant	Horse	Kangaroo	Monkey	Penguin	Rhino	Squirrel	Tổng
#Ảnh	12	14	14	13	14	16	22	16	11	14	17	163
#Thực thể	12	14	15	19	14	17	25	20	14	14	17	181

Tuy nhiên, sự mất cân bằng về số lượng ảnh ở một số lớp khi kế thừa CAMO++ gây ra vấn đề khi tạo lập cấu trúc cho hướng tiếp cận học với ít dữ liệu của CAMO-FS. Với bối cảnh học dựa trên ít dữ liệu huấn luyện (few-shot learning), chúng tôi cần có một số lượng mẫu nhất định của từng lớp phục vụ cho quá trình huấn luyện giai đoạn tinh chỉnh. Tuy nhiên, một số lớp đối tượng trong tập dữ liệu này có ít hơn số lượng mẫu cần thiết (5 mẫu thực thể cho mỗi lớp). Cụ thể, có 11 lớp đối tượng gặp phải tình trạng này (đó là: *Camel, Dolphin, Elephant, Horse, Kangaroo, Monkey, Penguin, Bat, Bear, Squirrel, và Rhino*). Vì thế, chúng tôi tiến hành thu thập thêm 163 ảnh tương ứng với 181 thực thể, với trung bình 15 cá thể cho mỗi lớp. Đồng thời, chúng tôi cũng loại bỏ các ảnh có nhãn bị lỗi trong tập dữ liệu ban đầu. Số liệu thống kê về dữ liệu đã thu thập thêm được hiển thị trong Bảng 4.1. Theo đó, bộ dữ liệu CAMO-FS có tổng số 2, 858 hình ảnh tương ứng

với 3,342 nhãn đối tượng. Hình 2.19 trực quan hóa một số hình ảnh mẫu với nhãn mặt nạ từ tập dữ liệu CAMO-FS. Tập dữ liệu CAMO-FS là một trong những tập dữ liệu đầu tiên được đề xuất với cấu trúc phục vụ cho bài toán phân đoạn thực thể nguy trang sử dụng ít dữ liệu huấn luyện.

Trong Bảng 4.2, chúng tôi trình bày số lượng và tỉ lệ của số lượng thực thể trên số lượng ảnh của tập dữ liệu CAMO-FS. Số lượng thực thể trên mỗi ảnh dao động từ 1 đến 25 thực thể và thường rơi vào 1, sau đó là 2 và 3 thực thể trên một ảnh. Có thể thấy rằng, số lượng ảnh chứa từ 1 đến 3 thực thể chiếm một tỷ lệ lớn trong toàn bộ tập dữ liệu. Điều này cũng minh họa cho vấn đề mất cân bằng dữ liệu giữa số lượng thực thể và tỷ lệ ảnh trong tập dữ liệu, phản ánh một vấn đề thực tế về sự xuất hiện của các loài động vật nguy trang. Thật vậy, các loài động vật nguy trang có mục tiêu ẩn mình, cho nên chúng thường sẽ xuất hiện độc lập, riêng lẻ. Ngoài ra, mặc dù được khẳng định trong công trình [36] rằng các đối tượng nguy trang thu thập trong tập CAMO++ xuất hiện trên toàn bộ các vị trí trong ảnh, sau khi loại bỏ các đối tượng không nguy trang và thêm các ảnh nguy trang mới, chúng tôi trực quan hóa phân bố của tâm các thực thể theo tọa độ chuẩn hóa của ảnh trên toàn bộ tập dữ liệu CAMO-FS như trong Hình 4.3-a. Điều này chỉ ra rằng các loài động vật nguy trang có xu hướng nằm ở trung tâm của ảnh chụp. Thực tế, để chụp được những hình ảnh của các loài động vật nguy trang trong tự nhiên, các nhiếp ảnh gia cần tập trung vào các loài vật đó, dẫn đến phân bố của chúng chủ yếu ở bộ cục trung tâm trên các hình ảnh thu thập được.

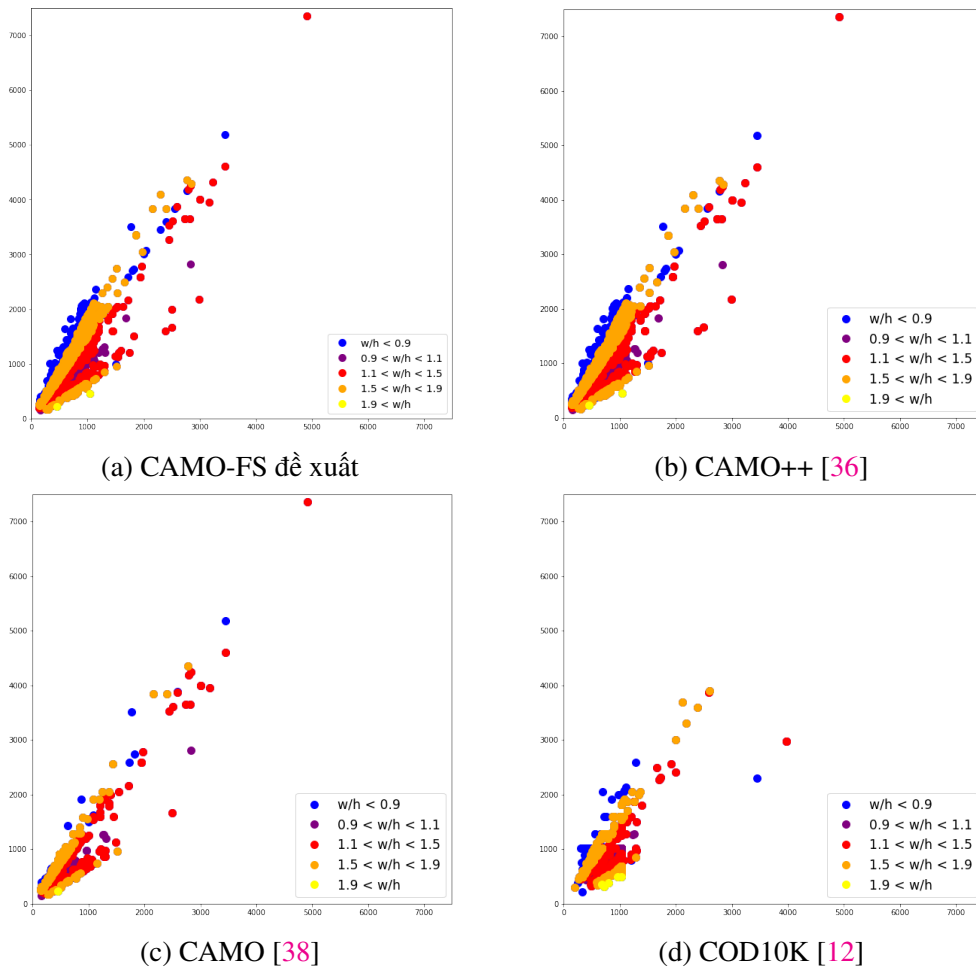
BẢNG 4.2: Tỉ lệ số lượng thực thể nguy trang trung bình trên ảnh của tập dữ liệu CAMO-FS đề xuất.

Số lượng thực thể	Tỉ lệ (%)	Số lượng ảnh
1	90.5	2581
2	1.05	190
3	1.79	51
3+	6.66	30

Cũng trong Hình 4.3, chúng tôi minh họa cho xu hướng tập trung của các thực thể trong hình ảnh nguy trang của các tập dữ liệu khác như CAMO [38] và COD10K [12] để so sánh trực quan hơn. Trong Hình 4.2, chúng tôi trình bày phân bố độ phân giải của



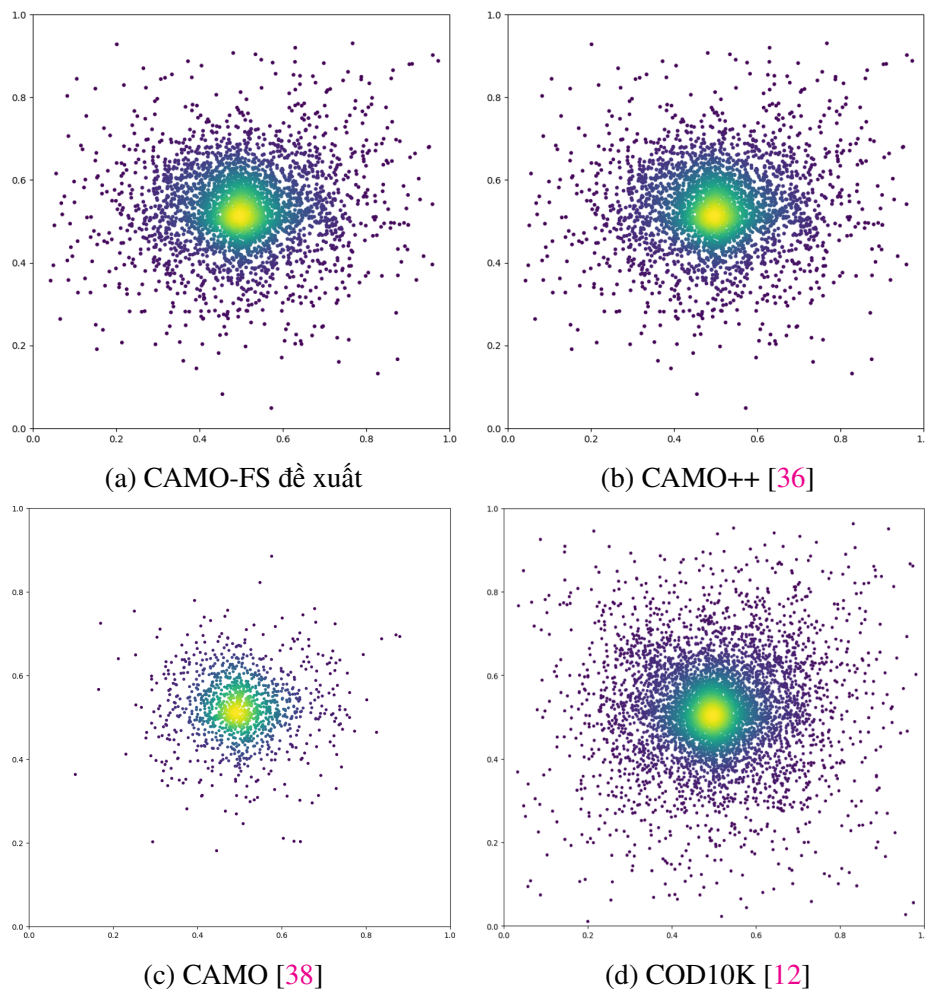
hình ảnh giữa các tập dữ liệu ngẫu trang. Vì chúng tôi chỉ xem xét sử dụng các hình ảnh ngẫu trang của CAMO++ [36] và COD10K [12], mật độ của CAMO-FS của chúng tôi dày đặc hơn CAMO++ do việc thu thập thêm các hình ảnh được trình bày trong Bảng 4.1. So với COD10K [12] và CAMO [38] trước đây, phân bố độ phân giải hình ảnh của CAMO-FS của chúng tôi thoả mãn hơn về sự đa dạng.



HÌNH 4.2: Phân phối độ phân giải ảnh trên các tập dữ liệu về thực thể ngẫu trang: CAMO-FS, CAMO++ [36], CAMO [38], và COD10K [12].

Mặc dù được công bố trong công trình [36] rằng các đối tượng được ngẫu trang trong CAMO++ đã được bản địa hóa trên toàn bộ hình ảnh, nhưng sau khi loại bỏ các đối tượng không ngẫu trang và thêm các hình ảnh được ngẫu trang mới, chúng tôi có sự phân bố các trung tâm đối tượng trong tọa độ hình ảnh chuẩn hóa trên tất cả các hình ảnh trong tập dữ liệu CAMO-FS như trong Hình 4.3-a. Điều này có nghĩa là động vật ngẫu trang có xu hướng nằm ở trung tâm của hình ảnh. Cũng trong Hình 4.3, chúng tôi

minh họa độ lệch tâm của hình ảnh được ngẫu trang trong các bộ dữ liệu CAMO khác [38] và COD10K [12] để so sánh trực quan tốt hơn. Trong Hình 4.2, chúng tôi trình bày độ phân giải hình ảnh trong các bộ dữ liệu ngẫu trang. Vì chúng tôi chỉ xem xét các hình ảnh ngẫu trang của CAMO++ [36] và COD10K [12], nên mật độ CAMO-FS cao hơn so với CAMO++ do chúng tôi có thu thập thêm hình ảnh được trình bày trong Bảng 4.1. So với COD10K [12] và CAMO [38] trước đó, phân phối độ phân giải hình ảnh của CAMO-FS đa dạng hơn.



HÌNH 4.3: Độ lệch tâm thực thể của các bộ dữ liệu ngẫu trang

Để tạo lập dữ liệu cho hướng tiếp cận học ít dữ liệu một cách hiệu quả, chúng tôi lấy  $M$  thực thể từ tập dữ liệu CAMO-FS để tạo các tập huấn luyện (trong thiết lập của chúng tôi,  $M = 5$ ) và sử dụng các thực thể còn lại để làm tập kiểm tra. Với  $M = 5$ , chúng tôi tạo lập cấu trúc cho các số lượng mẫu khác nhau trong tập huấn luyện và đảm bảo mẫu có số lượng lớn hơn sẽ bao hàm thực thể của mẫu có số lượng nhỏ hơn. Cụ thể, chúng

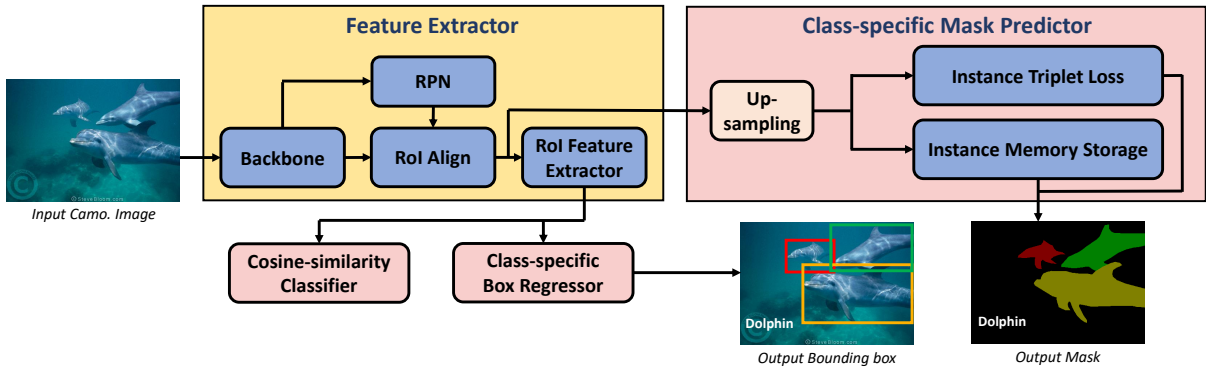
tôi lấy tất cả các mẫu để tạo dữ liệu huấn luyện 5 mẫu (5-shot) và loại bỏ 2 trong 5 thực thể để tạo dữ liệu huấn luyện 3 mẫu (3-shot). Theo cách này, tập huấn luyện 5 mẫu chứa các thực thể của tập dữ liệu 3 mẫu và tập huấn luyện 3 mẫu chứa các thực thể của tập huấn luyện 2 mẫu, tập huấn luyện 2 mẫu chứa thực thể của tập dữ liệu 1 mẫu. Theo khảo sát của chúng tôi trên các công trình được công bố tại các hội nghị, tạp chí uy tín hàng đầu trong ngành, đây là một trong những công trình đầu tiên để giải quyết vấn đề phân đoạn và phát hiện thực thể nguy trang với ít mẫu dữ liệu. Vì thế, việc xây dựng một tập dữ liệu đặc thù để huấn luyện và kiểm thử cho bài toán này là cần thiết và chúng tôi có khả năng đóng góp cho các công trình nghiên cứu sau này của cộng đồng.

### 4.3 Mô hình FS-CDIS phân đoạn thực thể nguy trang với ít mẫu dữ liệu

#### 4.3.1 Giới thiệu mô hình

**Các định nghĩa.** Hướng tiếp cận học ít dữ liệu thường chia dữ liệu huấn luyện thành hai tập: tập thứ nhất chứa các lớp dữ liệu tổng quát gọi là lớp cơ sở  $C_{S_{base}}$  với một lượng lớn dữ liệu huấn luyện sẵn có; tập thứ hai chứa các lớp mới là các đối tượng cần thực hiện tác vụ  $C_{S_{novel}}$  chứa một lượng nhỏ dữ liệu huấn luyện (thường dưới 10 mẫu). Mục tiêu của bài toán là huấn luyện một mô hình để đưa ra các dự đoán tốt trên các lớp mới  $C_{S_{test}} = C_{S_{novel}}$  [70] hoặc trên cả dữ liệu cơ sở và dữ liệu mới  $C_{S_{test}} = C_{S_{base}} \cup C_{S_{novel}}$  [22]. Hướng tiếp cận phân loại ảnh với ít mẫu dữ liệu đã giới thiệu phương pháp học theo tuần tự từng tập dữ liệu. Phương thức này thiết lập một loạt tập con  $E_i = (I_q, S_i)$  trong đó  $S_i$  là tập hỗ trợ chứa  $N$  các lớp từ  $C_{S_{train}} = C_{S_{novel}} \cup C_{S_{base}}$  cùng với  $K$  mẫu cho mỗi lớp (được gọi là  $N$ -way  $K$ -shot). Sau đó, một mạng được huấn luyện để phân loại ảnh đầu vào  $I_q$ , được gọi là ảnh truy vấn. Ý tưởng chính ở đây là việc giải quyết mỗi tác vụ phân loại khác nhau cho mỗi tập sẽ giúp khái quát hóa tốt hơn và mang lại kết quả tốt hơn trên  $C_{S_{novel}}$ . Các bài toán mở rộng của phương pháp này là phát hiện đối tượng ít mẫu - FSOD [32] và phân đoạn thực thể ít mẫu - FSIS [15, 84]. Những công trình này đề xuất xem tất cả các đối tượng trong ảnh là các truy vấn, kế thừa ý tưởng từ bài toán

phân loại ít mẫu. Tuy nhiên, có những thách thức đặc thù trong FSIS bởi đó không chỉ là tác vụ phân loại đối tượng mà còn mục tiêu xác định vị trí và phân đoạn các đối tượng trong ảnh. Vì thế, sử dụng ảnh  $I_q$  để truy vấn, FSIS trả về nhãn  $y_i$ , hộp giới hạn  $b_i$  và mặt nạ phân đoạn  $M_i$  cho tất cả các đối tượng trong  $I_q$  thuộc tập hợp  $C_{Stest}$ .



HÌNH 4.4: Mô hình FS-CDIS đề xuất cho phân đoạn thực thể nguy trang sử dụng ít dữ liệu huấn luyện.

**Tổng quan mô hình FS-CDIS.** Để giải quyết tác vụ phát hiện đối tượng trên ảnh với ít mẫu dữ liệu, mô hình TFA [77] đã được đề xuất dựa trên nền tảng Faster R-CNN [63]. Tiếp sau đó, cùng sử dụng ý tưởng gắn thêm một nhánh chuyên biệt cho tác vụ phân đoạn ảnh như Mask R-CNN [28], MTFA [20] đã ra đời để giải quyết bài toán phân đoạn thực thể với ít dữ liệu huấn luyện. Trong công trình này, chúng tôi dựa trên kiến trúc của mô hình MTFA [20] để đề xuất mô hình FS-CDIS cho bài toán phân đoạn thực thể nguy trang. Các cải tiến của chúng tôi tập trung vào nhánh phân đoạn thực thể với hai đề xuất liên quan đến hàm mất mát ba thành phần và bộ nhớ lưu trữ thực thể. Cũng giống như MTFA, mô hình của chúng tôi được huấn luyện trên 80 lớp cơ sở lấy từ bộ dữ liệu COCO để hoàn thành giai đoạn cơ sở (base phase). Trong giai đoạn tinh chỉnh (novel phase), chúng tôi áp dụng kỹ thuật học với ít mẫu để học các dữ liệu mới về các mẫu dữ liệu nguy trang trong bộ dữ liệu CAMO-FS mà chúng tôi đề xuất.

**Hình 4.4** minh họa các thành phần chính của mô hình đề xuất FS-CDIS. Trong đó, các đóng góp chính tập trung ở nhánh phân đoạn thực thể. Tương tự như Mask R-CNN [28], ảnh đầu vào được đưa vào một bộ rút trích đặc trưng Feature Extractor để rút trích đặc trưng  $F$ . Feature Extractor bao gồm mô hình nền tảng - Backbone  $B$ , mạng đề xuất vùng quan tâm - RPN, mô-đun căn chỉnh vùng quan tâm - RoI Align, và mô-đun rút

trích đặc trưng từ vùng quan tâm - RoI Feature Extractor. Về đầu ra, FS-CDIS có bộ ba đầu ra (heads) tương ứng với ba tác vụ mà mô hình hỗ trợ: đầu ra phân loại  $C$ , đầu ra dự đoán vị trí khung bao  $R$  và đầu ra dự đoán mặt nạ ngữ nghĩa  $M$ .

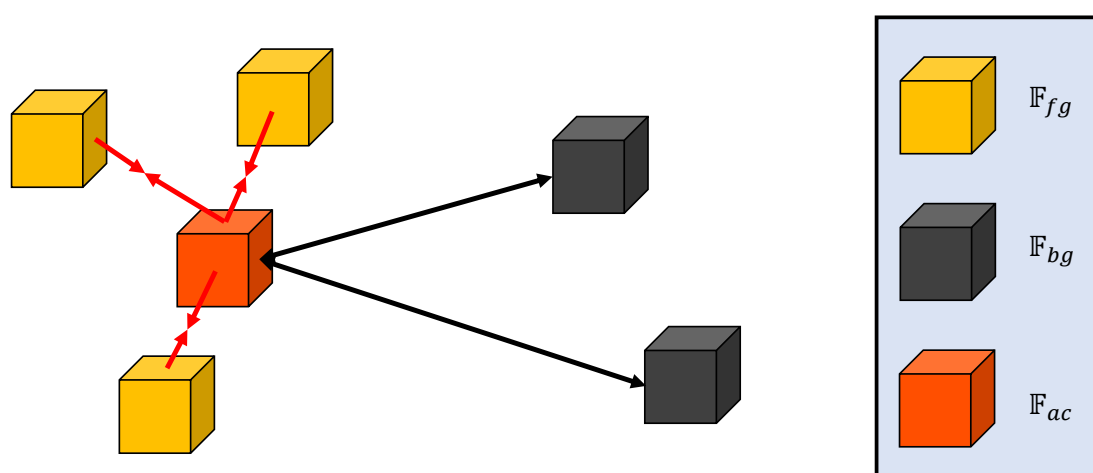
Như đã trình bày ở phần trước, quá trình huấn luyện một mô hình học máy với ít dữ liệu huấn luyện được chia làm hai giai đoạn: giai đoạn cơ sở và giai đoạn tinh chỉnh, cụ thể như sau. Trong giai đoạn đầu tiên, mô hình được huấn luyện trên các lớp cơ sở  $C_{base}$ , trong trường hợp này là 80 lớp cơ sở của tập dữ liệu COCO. Sau đó, trong giai đoạn thứ hai, chúng tôi đóng băng backbone  $B$  của bộ rút trích đặc trưng  $F$  và chỉ thực hiện huấn luyện trên các đầu ra dự đoán. Do đó, đầu ra phân loại  $C$ , đầu ra dự đoán vị trí khung bao  $R$  và đầu ra dự đoán mặt nạ ngữ nghĩa  $M$  được tinh chỉnh với dữ liệu mới trong giai đoạn thứ hai. Để nâng cao hiệu suất của tác vụ phân đoạn thực thể, chúng tôi cải tiến bằng cách áp dụng các ý tưởng về hàm mất mát ba thành phần và bộ nhớ lưu trữ thực thể của kỹ thuật học tương phản, chi tiết được mô tả trong phần tiếp theo. Hai cải tiến này không chỉ bắt nguồn từ tác vụ phân đoạn thực thể nói chung mà còn từ các điểm đặc thù của bài toán phân đoạn thực thể ngẫu trang.

### 4.3.2 Khai thác đặc trưng ngẫu trang với kỹ thuật học tương phản

Một trong những điểm đặc thù của các thực thể ngẫu trang nằm ở chất liệu (texture) của chúng được mô tả trên ảnh màu sẽ tương tự như vùng nền. Điều này làm nên tính chất ngẫu trang, giúp động vật lẫn trốn kẻ thù nguy hiểm, và cũng làm cho quá trình phân định vùng chứa thực thể và vùng nền trở nên khó khăn hơn. Hơn nữa, trong bối cảnh ít dữ liệu huấn luyện, mô hình sẽ phải xoay sở để học được các khái niệm (concept) này dựa vào một số lượng mẫu dữ liệu ít ỏi. Vì thế, câu hỏi đặt ra là làm thế nào để giúp mô hình học được tốt hơn các đặc trưng ngẫu trang trong khi được cung cấp ít dữ liệu?

Để trả lời cho câu hỏi này, đề xuất đầu tiên của chúng tôi giúp mô hình tăng khả năng phân biệt vùng vật thể và vùng nền thông qua một công cụ đó là hàm mất mát. Hàm mất mát chúng tôi đề xuất sử dụng là hàm mất mát ba thành phần [66] (hay còn gọi là Triplet loss). Trong ngữ cảnh của bài toán này, chúng tôi áp dụng hàm mất mát trên từng thực thể vì thế đây được gọi là hàm mất mát ba thành phần ở cấp độ thực thể (Instance

Triplet Loss). Một cách tổng quát, hàm mất mát ba thành phần ở cấp độ thực thể nhận đầu vào là các đặc trưng tiền cảnh - foreground, vùng nền - background, và điểm neo - anchor. Hàm mất mát này có mục tiêu thu hẹp khoảng cách giữa các đặc trưng tiền cảnh với điểm neo, và gia tăng khoảng cách giữa các đặc trưng vùng nền với điểm neo (trực quan hóa tại Hình 4.5). Theo đó, hàm mất mát này sẽ hỗ trợ mô hình tạo ra các đặc trưng phân biệt giữa các vùng nền và vùng thuộc thực thể ngụy trang.



HÌNH 4.5: Mô hình hóa hàm mất mát ba thành phần với các đặc trưng tiền cảnh ( $F_{fg}$ ), vùng nền ( $F_{bg}$ ), và điểm neo ( $F_{ac}$ ). Hàm mất mát ba thành phần có mục tiêu thu hẹp khoảng cách giữa các đặc trưng tiền cảnh với điểm neo, và gia tăng khoảng cách giữa các đặc trưng vùng nền với điểm neo.

Để tính hàm mất mát, chúng tôi sử dụng thông tin từ mặt nạ phân đoạn. Các đặc trưng từ vùng quan tâm (RoI) cung cấp đặc trưng vùng nền  $F_{bg}$  và đặc trưng vùng thực thể  $F_{fg}$  theo vị trí trên mỗi vùng quan tâm. Cả  $F_{bg}$  và  $F_{fg}$  của mỗi vùng quan tâm đều được sử dụng để tính các hàm mất mát đề xuất. Với ý tưởng tăng cường khả năng phân biệt giữa các thực thể ngụy trang và nền xung quanh chúng, chúng tôi coi các điểm ảnh của đối tượng là điểm positive và vùng nền là điểm negative. Theo đó, chúng tôi buộc mô hình học các đặc điểm nổi bật giữa các đại diện của thực thể và vùng nền. Các đặc trưng càng được phân biệt rõ ràng thì mô hình càng có khả năng phát hiện hoặc phân đoạn các trường hợp được ngụy trang tốt hơn. Bằng cách này, chúng tôi làm nổi bật các thực thể ngụy trang có trong ảnh để mô hình có thể phát hiện và phân đoạn các thực thể này dễ dàng, giảm thiểu sai sót.

Cụ thể, đối với mỗi vùng quan tâm trong ảnh, chúng tôi xem xét các đặc trưng trung



bình của thực thể  $\mathbb{F}_{ac} = \mathbb{F}_{avg} = \frac{1}{|\mathbb{F}_{fg}|} \sum \mathbb{F}_{fg}$  như **điểm neo**; các đặc trưng của thực thể nguy trang  $\mathbb{F}_{fg}$  là các **điểm positive**; các đặc trưng vùng nền  $\mathbb{F}_{bg}$  là các **điểm negative** để áp dụng hàm mất mát ba thành phần ở cấp độ thực thể. Bằng cách này, mô hình sẽ cố gắng học cách tối thiểu hóa khoảng cách giữa các đại diện thuộc thực thể và tối đa hóa khoảng cách giữa các đại diện nền như minh họa tại **Hình 4.5**. Để xác định khoảng cách giữa các đặc trưng nói trên, chúng tôi sử dụng độ tương đồng Cosine (Cosine Similarity) thay cho các độ đo như Manhattan hay Euclidean. Chúng tôi sử dụng độ tương đồng Cosine vì sự ưu việt của độ đo này khi tích hợp được các yếu tố liên quan đến góc của các vector trong không gian đặc trưng, yếu tố này giúp cho quá trình xác định khoảng cách được chính xác hơn.

Tổng hợp lại, hàm mất mát ba thành phần ở cấp độ thực thể được định nghĩa như **Công thức 4.1** sau:

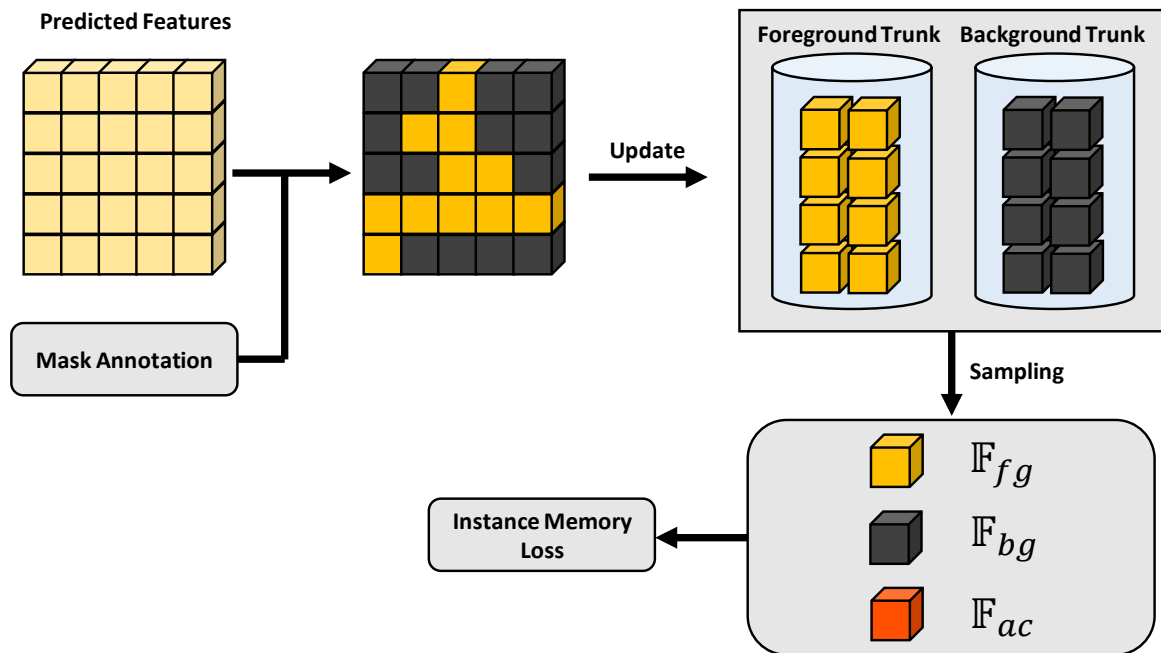
$$\begin{aligned} \mathcal{L}_{triplet} &= \max\{d(\mathbb{F}_{avg}, \mathbb{F}_{fg}) - d(\mathbb{F}_{avg}, \mathbb{F}_{bg}) + margin, 0\} \\ d(x, y) &= 1 - \frac{x \cdot y}{\|x\| \cdot \|y\|}, \end{aligned} \tag{4.1}$$

trong đó tham số *margin* kiểm soát mức độ phân biệt giữa các đặc trưng của vùng thực thể và vùng nền. Trong các thử nghiệm, chúng tôi đặt  $margin = 0.5$ . Các thực nghiệm loại suy sẽ được trình bày trong phần tiếp theo.

### 4.3.3 Cung cố đặc trưng nguy trang với Bộ nhớ lưu trữ thực thể

Quan sát thấy việc cung cấp một số lượng ít mẫu dữ liệu sẽ cản trở khả năng học của mô hình, chúng tôi đề xuất phương pháp giúp mô hình có khả năng ghi nhớ những thông tin đã học được, và sử dụng các thông tin này để cải thiện khả năng đưa ra dự đoán. Lấy cảm hứng từ thực tế, khi đối mặt với một vấn đề mới, chúng ta thường có xu hướng sử dụng những thông tin, kiến thức đã có trong tri thức của mình để xử lý chúng. Với ý tưởng này, chúng tôi tiếp tục đề xuất thêm vào mô hình một bộ nhớ lưu trữ thực thể (Instance Memory Storage) giúp mô hình có khả năng ghi nhớ các mẫu dữ liệu giới hạn trong quá trình học để cải thiện độ chính xác.

Bộ nhớ lưu trữ thực thể được thiết kế để lưu trữ thông tin là các đặc trưng trong một lớp ngữ nghĩa và được cập nhật liên tục trong quá trình huấn luyện với mỗi mẫu dữ liệu. Cơ chế này cho phép mô hình nắm bắt thông tin về lớp ngữ nghĩa đó ở cấp độ tổng quát, tổng hợp được thông tin từ các thực thể khác nhau trong cùng một lớp ngữ nghĩa. Mặt khác, việc lưu trữ và cập nhật các đặc trưng trong bộ nhớ lưu trữ cho mỗi lần lặp trong quá trình huấn luyện cũng tạo ra nhiều biến thể hơn thông qua việc pha trộn đặc trưng sau mỗi lần cập nhật bộ nhớ. Chúng tôi sử dụng bộ nhớ lưu trữ thực thể để chứa các đặc trưng vùng nền và vùng thực thể theo từng lớp ngữ nghĩa và sử dụng các đặc trưng này để tính toán sự phân biệt giữa các vùng có thực thể và vùng không có thực thể trong các vùng đề xuất (Hình 4.6).



HÌNH 4.6: Mô hình hóa bộ nhớ lưu trữ thực thể. Bộ nhớ lưu trữ thực thể lưu trữ thông tin vùng nền và vùng thực thể theo từng lớp ngữ nghĩa và sử dụng chúng để tính toán sự phân biệt cho nhánh phân đoạn thực thể. Bộ nhớ lưu trữ thực thể được cập nhật liên tục trong quá trình huấn luyện với mỗi mẫu dữ liệu mới theo từng lớp ngữ nghĩa.

**Quá trình lưu trữ và cập nhật đặc trưng.** Bộ nhớ lưu trữ thực thể cho mỗi lớp ngữ nghĩa có  $2N$  số lượng đặc trưng, bao gồm  $N$  đặc trưng của thực thể và  $N$  đặc trưng của vùng nền. Khi nhận được các đặc trưng từ những dữ liệu mới, bộ nhớ lưu trữ kết hợp chúng với các đặc trưng cũ hiện có. Trong trường hợp số lượng đặc trưng vượt quá

ngưỡng  $N$  cho trước, các đặc trưng cũ nhất sẽ được bộ nhớ giải phóng. Quá trình này diễn ra đảm bảo cho bộ nhớ luôn được cập nhật những thông tin mới từ các mẫu dữ liệu mới. Quá trình tìm ra dung lượng bộ nhớ ( $N$ ) phù hợp được chúng tôi thực nghiệm loại suy ở phần sau.

**Quá trình lấy mẫu (Sampling).** Tương tự như hàm mất mát ba thành phần, để tính được mất mát cho bộ nhớ lưu trữ với một mẫu dữ liệu mới, bộ nhớ lưu trữ cần cung cấp ba phần tử. Đặc trưng vùng tiền cảnh  $\mathbb{F}_{fg}$ , đặc trưng vùng nền  $\mathbb{F}_{bg}$ , và đặc trưng tổng quát  $\mathbb{F}_{ac}$ .  $\mathbb{F}_{fg}$  và  $\mathbb{F}_{bg}$  là các đặc trưng thực thể và vùng nền bộ nhớ lưu trữ trong hai thùng chứa (Trunk).  $\mathbb{F}_{ac}$  là tổng quát của thực thể tại lớp ngữ nghĩa đó và được tạo ra cho mỗi lớp bằng cách lấy trung bình của tất cả các đặc trưng thực thể  $\mathbb{F}_{fg}$ .

Gọi  $\mathbb{F}_{fg}^i$  là thực thể thứ  $i$  và  $\tau$  là siêu tham số thể hiện mức độ cập nhật (xem thêm tại [81]). Trong các thực nghiệm, chúng tôi sử dụng  $\tau = 1$ . Hàm mất mát cho bộ nhớ lưu trữ thực thể ngẫu nhiên được định nghĩa như **Công thức 4.2** sau:

$$\mathcal{L}_{memory} = -\log \frac{\exp(\mathbb{F}_{ac} \cdot \mathbb{F}_{fg}^i / \tau)}{\sum_{j=0}^{|\mathbb{F}_{bg}|} \exp(\mathbb{F}_{ac} \cdot \mathbb{F}_{bg}^j / \tau) + \exp(\mathbb{F}_{ac} \cdot \mathbb{F}_{fg}^i / \tau)} \quad (4.2)$$

Tổng hợp lại, hàm mất mát cuối cùng của mô hình FS-CDIS được trình bày như **Công thức 4.3** sau:

$$\mathcal{L}_{FS-CDIS} = \mathcal{L}_{mrcnn} + \alpha \mathcal{L}_{triplet} + \beta \mathcal{L}_{memory}, \quad (4.3)$$

trong đó  $\mathcal{L}_{mrcnn}$  là các hàm mất mát của Mask R-CNN [28],  $\alpha$  và  $\beta$  lần lượt là trọng số cân bằng của hai hàm mất mát đề xuất  $\mathcal{L}_{triplet}$  và  $\mathcal{L}_{memory}$ . Thực nghiệm loại suy về tỉ trọng của các hàm mất mát này được đề cập ở phần tiếp theo.

## 4.4 Thực nghiệm

### 4.4.1 Cấu hình thực nghiệm

Như đã đề cập ở phần trước, chúng tôi sử dụng tập dữ liệu đề xuất CAMO-FS chứa các hình ảnh của các loài động vật ngẫu nhiên trong tự nhiên để thiết lập việc đánh giá mô

hình và các cải tiến được đề xuất của chúng tôi. Chúng tôi cùng sử dụng quy trình thực nghiệm được công bố trong các công trình trước đây của bài toán phát hiện đối tượng ít mẫu dữ liệu [32, 77, 84]. Trong giai đoạn cơ sở, chúng tôi huấn luyện mô hình của mình với dữ liệu phong phú từ 80 lớp của tập dữ liệu COCO như được đề xuất trong [32]. Trong giai đoạn tinh chỉnh, chúng tôi đánh giá độ chính xác khi sử dụng  $K = \{1, 2, 3, 5\}$  mẫu cho mỗi lớp mới (novel class).

Để chứng minh hiệu quả của mô hình FS-CDIS trên các tác vụ phát hiện và phân đoạn thực thể nguy trang, chúng tôi sử dụng độ chính xác trung bình (AP - Average Precision) và độ phủ trung bình (AR - Average Recall). Cụ thể, chúng tôi sử dụng AP@50 và AP@75, cùng với AR@10. Ngoài ra, chúng tôi cũng sử dụng AP và AR ở các mức độ nhỏ, trung bình và lớn của kích thước thực thể để hiểu rõ hơn về hiệu suất của mô hình. Thông tin chi tiết về các độ đo có thể được truy cập tại trang chủ của tập dữ liệu COCO tại <https://cocodataset.org/#detection-eval>.

Mô hình cơ sở MTFA [20] của chúng tôi được triển khai sử dụng nền tảng Detectron2 [80]. Chúng tôi sử dụng kiến trúc backbone là Feature Pyramid Network [41] với ResNet-101 [26] (FPN-ResNet101). Các thực nghiệm được tiến hành trên một GPU GeForce RTX 2080Ti với kích thước batch là 2 ảnh. Giai đoạn tinh chỉnh có tốc độ học là  $lr = 1.25 \times 10^{-3}$  được suy ra từ cấu hình của MTFA. Chúng tôi lần lượt thiết lập các tham số cân bằng  $\alpha = 1 \times 10^{-1}$  và  $\beta = 1 \times 10^{-2}$  khi chúng tôi huấn luyện mô hình với hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể. Các tham số khác của mô hình được trình bày tại các công bố TFA [77] hoặc Detectron2 [80].

#### 4.4.2 Kết quả thực nghiệm

##### So sánh với các mô hình tiên tiến khác (State-of-the-art)

Để chứng minh hiệu quả của các phương pháp được đề xuất của chúng tôi, chúng tôi đã tiến hành các thí nghiệm trên tập dữ liệu đề xuất CAMO-FS. Số mẫu  $K$  lần lượt được sử dụng  $K = \{1, 2, 3, 5\}$  mẫu. Vì một số công trình gần đây chưa công bố mã nguồn của họ [25, 74], chúng tôi so sánh hiệu quả các đề xuất của mình với những mô hình đã công bố giải quyết đồng thời hai tác vụ phát hiện và phân đoạn thực thể. Bảng 4.3 trình bày

BẢNG 4.3: So sánh độ chính xác trên tập dữ liệu CAMO-FS giữa các mô hình tiên tiến như MTFA [20], Mask RCNN<sup>†</sup> [28], iFS-RCNN [56], và mô hình đề xuất FS-CDIS với hàm mất mát ba thành phần ở cấp độ thực thể (-ITL) và bộ nhớ lưu trữ thực thể (-IMS). Đề xuất của chúng tôi cải thiện độ chính xác trên các mô hình cơ sở (các dòng tô màu xám).

Mô hình		AP giai đoạn tính chỉnh										FPS	#Params
Phương pháp	Mô hình cơ sở	Phân đoạn thực thể					Phát hiện đối tượng						
		1	2	3	5	Avg.	1	2	3	5	Avg.		
MTFA [20]	COCO-80 ResNet-50	2.48	6.67	5.81	6.40	5.34	1.98	6.47	5.82	6.17	5.11	-	-
M-RCNN <sup>†</sup> [28]		4.08	6.79	6.90	8.29	6.52	2.82	5.09	5.46	6.18	4.89		
iFS-RCNN [56]		4.17	6.26	5.73	6.38	5.64	3.92	6.06	5.47	6.60	5.51		
MTFA [20]	COCO-80	3.66	6.21	6.16	5.95	5.50	2.93	5.90	5.84	5.84	5.13	15.1	
FS-CDIS-ITL	ResNet-101	4.46	5.57	6.41	8.48	<b>6.23</b>	4.04	7.28	7.49	9.76	<b>7.14</b>	14.6	<b>63.9M</b>
FS-CDIS-IMS	MTFA	5.46	6.95	7.36	9.61	<b>7.35</b>	4.50	6.95	7.55	10.36	<b>7.34</b>	15.0	
M-RCNN <sup>†</sup> [28]	COCO-80	4.39	7.69	7.94	10.09	7.53	3.03	5.80	6.20	7.79	5.71	12.2	
FS-CDIS-ITL	ResNet-101	5.73	7.97	8.52	9.92	<b>8.04</b>	5.08	7.56	7.85	9.67	<b>7.34</b>	13.0	<b>63.1M</b>
FS-CDIS-IMS	M-RCNN	5.52	7.84	8.65	9.82	<b>7.96</b>	4.92	7.39	7.96	9.52	<b>7.45</b>	12.9	
iFS-RCNN [56]	COCO-80	4.27	6.55	6.07	7.80	6.17	3.79	6.28	6.01	8.08	6.04	10.8	
FS-CDIS-ITL	ResNet-101	5.35	6.01	7.80	6.23	<b>6.35</b>	4.71	5.66	7.10	6.06	<b>5.88</b>	10.2	<b>77.4M</b>
FS-CDIS-IMS	iFS-RCNN	2.99	6.83	6.14	9.03	<b>6.25</b>	2.74	6.39	5.94	8.44	<b>5.88</b>	11.3	

M-RCNN<sup>†</sup> là mô hình Mask R-CNN [28] với bộ phân loại sigmoid.

đánh giá độ chính xác của các phương pháp đề xuất: hàm mất mát ba thành phần ở cấp độ thực thể (-ITL) và bộ nhớ lưu trữ thực thể (-IMS) trên mô hình cơ sở MTFA [20], mô hình Mask R-CNN [28] với bộ phân loại sigmoid, và phương pháp tiên tiến nhất iFS-RCNN [56] trong hướng tiếp cận của phân đoạn thực thể ít mẫu. Chúng tôi trình bày kết quả các thí nghiệm trên những mô hình nói trên và chọn backbone COCO-80 ResNet-101 làm mô hình cơ sở chung để áp dụng các phương pháp được đề xuất của chúng tôi. Chi tiết về sự lựa chọn backbone được trình bày trong phần thực nghiệm loại suy tiếp theo.

Về tác vụ phân đoạn thực thể, chúng tôi cải thiện độ chính xác tốt hơn so với MTFA [20], Mask RCNN<sup>†</sup> [28], và iFS-RCNN [56] với trung bình AP là 6.23%, 8.04%, 6.35% nhờ vào hàm mất mát ba thành phần ở cấp độ thực thể, và 7.35%, 7.96%, 6.25%, nhờ vào bộ nhớ lưu trữ thực thể. Về tác vụ phát hiện đối tượng, FS-CDIS của chúng tôi đạt được các giá trị trung bình AP là 7.14%, 7.34%, 5.88%, lần lượt với hàm mất mát ba thành phần ở cấp độ thực thể và 7.34%, 7.45%, 5.88% với bộ nhớ lưu trữ thực thể. Với điều kiện lí tưởng, các thực nghiệm cần được thực hiện ngẫu nhiên nhiều lần (khoảng 10 lần) và lấy kết quả trung bình để thể hiện tính khách quan. Tuy nhiên, các thực nghiệm trên đây được chúng tôi thực nghiệm một lần với mỗi cấu hình vì tài nguyên hạn chế.

Độ chính xác chi tiết của các phương pháp của chúng tôi được trình bày trong **Bảng 4.3**. Mặc dù kết quả giới có hạn, chúng tôi đã cải thiện độ chính xác các mô hình tiền nhiệm trên các tác vụ phát hiện và phân đoạn thực thể trên hình ảnh nguy trang. Kết quả phân đoạn thực thể được cải thiện hơn nhờ vào các đề xuất phục vụ nhánh tác vụ này.

### **Đánh giá độ chính xác các thành phần đề xuất**

**Bảng 4.4** trình bày kết quả của mô hình cơ sở MTFA [20] với cấu hình mặc định ban đầu cùng với các cải tiến được đề xuất của chúng tôi, lần lượt là hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể. Trên nền tảng của mô hình cơ sở MTFA [20], chúng tôi thiết lập thực nghiệm các cấu hình tinh chỉnh bằng cách huấn luyện tất cả các đầu ra phân loại, đầu ra dự đoán khung bao và đầu ra dự đoán nhãn mặt nạ trên dữ liệu mới ít mẫu để phân đoạn thực thể nguy trang. Các kết quả được trình bày chứng minh tính hiệu quả của hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể mà chúng tôi giới thiệu.

Nhìn chung, các phương pháp của chúng tôi đạt được kết quả cao hơn so với các mô hình cơ sở. Khi phân đoạn thực thể, phương pháp của chúng tôi tăng lần lượt 1.9%, 3.5%, và 2.3% về AP, AP@50, và AP@75. Kết quả này thể hiện hiệu quả của các phương pháp của chúng tôi trong bối cảnh ít mẫu dữ liệu thực thể nguy trang. Cả hai đề xuất đều tăng khả năng phân biệt giữa các đặc trưng thực thể và vùng nền, giúp mô hình phân đoạn tốt hơn các điểm ảnh thuộc về các loài động vật nguy trang. Kết quả của bộ nhớ lưu trữ thực thể cao hơn kết quả của hàm mất mát ba thành phần ở cấp độ thực thể khoảng 1%. Chúng tôi nhận thấy việc lưu trữ các đại diện cho mỗi lớp là một yếu tố quan trọng trong tác vụ học ít mẫu. Kỹ thuật này không chỉ mở rộng số lượng các biến thể trong quá trình huấn luyện mà còn tăng tính tổng quát cho mỗi lớp ngữ nghĩa, do đó mô hình có thể phân đoạn các đối tượng khó tốt hơn. Theo những cách này, chúng tôi cũng cải thiện các kết quả tương ứng với các độ đo khác trong việc phát hiện đối tượng nguy trang.

Trong **Bảng 4.4**, các cải tiến của chúng tôi giúp mô hình phân đoạn các loài động vật ở những kích thước khác nhau. Cụ thể, tất cả ba chỉ số bao gồm APs, APm và API đều cải thiện so với mô hình cơ sở, cho thấy mô hình của chúng tôi phân đoạn tốt các loài động vật nhỏ, trung bình và lớn. Hiện tượng này cũng xảy ra tương tự trên tác vụ phát



BẢNG 4.4: Độ chính xác của hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể của chúng tôi trên mô hình MTFA [77]. Kết quả tốt nhất được in đậm. # kí hiệu số lượng mẫu, “Memory” là Instance Memory Storage and “Triplet” là Instance Triplet Loss.

#	Phương pháp	AP	AP50	AP75	APs	APm	API	AR1	AR10	ARs	ARm	ARI
<b>Phân đoạn thực thể</b>												
1	Baseline MTFA	3.66	5.37	4.09	22.42	4.35	2.01	11.30	13.58	25.97	<b>12.96</b>	12.53
	MTFA + Triplet	4.46	8.21	4.60	21.33	4.13	<b>4.01</b>	12.36	15.04	23.17	9.49	16.67
	MTFA + Memory	<b>5.46</b>	<b>9.20</b>	<b>6.17</b>	<b>27.79</b>	<b>6.20</b>	<b>4.01</b>	<b>17.08</b>	<b>19.99</b>	<b>29.41</b>	11.45	<b>20.89</b>
2	Baseline MTFA	6.21	8.92	7.28	32.64	<b>7.75</b>	3.50	18.88	21.12	<b>35.82</b>	<b>15.49</b>	20.14
	MTFA + Triplet	5.57	9.45	6.04	25.83	3.01	5.37	15.67	17.33	26.13	7.37	17.50
	MTFA + Memory	<b>6.95</b>	<b>10.72</b>	<b>7.60</b>	<b>33.62</b>	5.73	<b>6.44</b>	<b>20.00</b>	<b>22.15</b>	34.25	13.86	<b>20.92</b>
3	Baseline MTFA	6.16	8.95	6.68	33.74	6.19	5.08	20.25	22.95	36.83	16.31	21.63
	MTFA + Triplet	6.41	10.67	6.72	30.39	5.17	5.30	20.69	22.98	31.90	15.69	22.53
	MTFA + Memory	<b>7.36</b>	<b>11.23</b>	<b>8.49</b>	<b>37.03</b>	<b>6.24</b>	<b>5.64</b>	<b>24.40</b>	<b>27.69</b>	<b>38.44</b>	<b>17.02</b>	<b>26.71</b>
5	Baseline MTFA	5.95	8.67	6.94	34.71	<b>6.25</b>	4.85	21.29	24.42	36.86	<b>14.51</b>	24.83
	MTFA + Triplet	8.48	13.43	9.80	36.66	5.75	8.04	23.83	26.66	37.03	11.62	25.91
	MTFA + Memory	<b>9.61</b>	<b>14.61</b>	<b>11.73</b>	<b>38.60</b>	5.79	<b>10.40</b>	<b>26.65</b>	<b>30.37</b>	<b>39.21</b>	12.26	<b>30.02</b>
<b>Phát hiện đối tượng</b>												
1	Baseline MTFA	2.93	5.86	2.20	20.95	4.18	2.03	9.25	10.84	21.74	<b>11.49</b>	8.77
	MTFA + Triplet	4.04	8.65	2.98	20.50	4.90	<b>4.22</b>	12.89	<b>15.53</b>	20.73	11.45	17.46
	MTFA + Memory	<b>4.50</b>	<b>9.14</b>	<b>3.45</b>	<b>22.88</b>	<b>5.61</b>	3.54	<b>13.14</b>	15.22	<b>23.14</b>	8.78	<b>16.33</b>
2	Baseline MTFA	5.90	8.87	6.83	33.04	9.74	3.10	17.26	19.25	34.04	<b>15.74</b>	19.61
	MTFA + Triplet	<b>7.28</b>	<b>11.22</b>	<b>8.25</b>	32.31	<b>10.72</b>	<b>6.83</b>	<b>20.52</b>	<b>22.69</b>	32.34	14.88	<b>23.52</b>
	MTFA + Memory	6.95	10.88	7.75	<b>33.93</b>	7.49	6.81	19.84	22.01	<b>34.10</b>	15.04	21.47
3	Baseline MTFA	5.84	8.98	6.29	34.56	7.78	4.31	19.13	21.83	35.80	15.93	21.09
	MTFA + Triplet	7.49	<b>11.51</b>	8.23	<b>38.45</b>	8.61	<b>6.38</b>	<b>24.88</b>	<b>27.52</b>	<b>38.55</b>	17.66	27.44
	MTFA + Memory	<b>7.55</b>	11.45	<b>8.50</b>	38.07	<b>9.21</b>	5.70	24.20	27.29	38.50	<b>18.10</b>	<b>27.56</b>
5	Baseline MTFA	5.84	9.13	6.04	35.44	8.17	4.22	19.67	22.96	35.94	<b>14.16</b>	22.58
	MTFA + Triplet	9.76	14.37	11.12	<b>40.05</b>	<b>8.82</b>	9.89	25.93	29.28	<b>40.05</b>	12.53	30.32
	MTFA + Memory	<b>10.36</b>	<b>16.27</b>	<b>11.79</b>	39.32	8.08	<b>11.36</b>	<b>26.34</b>	<b>30.30</b>	39.35	12.37	<b>30.91</b>

hiện đối tượng. Khi dữ liệu rất khan hiếm như trong trường hợp 1 mẫu hoặc 2 mẫu, hàm mất mát ba thành phần ở cấp độ thực thể có kết quả tương đương với bộ nhớ lưu trữ thực thể. Tuy nhiên, trong bối cảnh huấn luyện trên 3 mẫu hoặc 5 mẫu (nhiều dữ liệu hơn), bộ nhớ lưu trữ thực thể thể hiện hiệu quả xuất sắc hơn nhờ vào việc lưu trữ và cập nhật bộ nhớ qua các lần lặp để tạo ra các đặc trưng phân biệt ở cấp độ toàn cục. Hình 4.7 minh họa sự so sánh trực quan giữa các kết quả của cấu hình huấn luyện với 5 mẫu của mô hình cơ sở MTFA [20] và các phương pháp được đề xuất của chúng tôi. Chúng tôi chọn để trực quan hóa các hình ảnh này với ngưỡng tin cậy khi dự đoán là 0.5, do đó số lượng lớn các dự đoán có độ tin cậy thấp từ các mô hình được loại bỏ. Hai hàng cuối là những trường hợp thách thức mà cả hai đề xuất của chúng tôi đều chưa thể xử lý tốt các thực thể ngụy trang này.

### 4.4.3 Thực nghiệm loại suy

BẢNG 4.5: Thực nghiệm loại suy trên các mô hình cơ sở với 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*. “Memory” là Instance Memory Storage và “Triplet” là Instance Triplet Loss.

Phương pháp	Mô hình cơ sở	Phân đoạn thực thể			Phát hiện đối tượng		
		AP	AP50	AP75	AP	AP50	AP75
Triplet	COCO-80 R-101	<b>4.46</b>	<b>8.21</b>	<b>4.60</b>	<b>4.04</b>	<b>8.65</b>	<b>2.98</b>
	COCO-80 R-50	3.68	<i>6.79</i>	3.81	2.85	<i>6.67</i>	1.65
	COCO-60 R-101	<i>3.87</i>	6.26	<i>3.90</i>	<i>3.37</i>	6.51	<i>2.69</i>
	COCO-60 R-50	2.56	4.25	2.79	2.28	4.13	2.26
Memory	COCO-80 R-101	<b>5.46</b>	<b>9.20</b>	<b>6.17</b>	<b>4.50</b>	<b>9.14</b>	<b>3.45</b>
	COCO-80 R-50	<i>3.87</i>	<i>6.81</i>	<i>3.91</i>	<i>3.40</i>	<i>6.94</i>	2.76
	COCO-60 R-101	2.89	4.50	3.26	2.76	4.66	<i>2.81</i>
	COCO-60 R-50	2.63	4.50	3.02	2.25	4.50	1.65

#### Thực nghiệm loại suy chọn các mô hình cơ sở (backbone)

Chúng tôi cũng tiến hành các thí nghiệm để chọn ra mô hình cơ sở phù hợp được huấn luyện sẵn trên tập COCO. Cụ thể, chúng tôi thực nghiệm độ chính xác của phương pháp được đề xuất của chúng tôi về hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể trên bốn mô hình cơ sở khác nhau. Các mô hình cơ sở được xem xét là ResNet-50 và ResNet-101 [26]. Hai tập dữ liệu cơ sở là MS-COCO với 80 lớp và 60 lớp. Do đó, sự kết hợp của chúng tạo ra bốn mô hình cơ sở khác nhau (tức là COCO-80 R-101, COCO-80 R-50, COCO-60 R-101 và COCO-60 R-50). Có thể thấy từ **Bảng 4.5**, độ chính xác của việc áp dụng mô hình cơ sở COCO-80 R-101 cho kết quả tốt hơn so với các mô hình khác được đánh giá qua AP, AP@50 và AP@75 trong cả hai tác vụ phân đoạn thực thể và phát hiện đối tượng. Trong cả hai trường hợp với hai cải tiến được đề xuất, kết quả thực nghiệm cho thấy việc lựa chọn COCO-80 R-101 là phù hợp nhất trong số các mô hình cơ sở được kiểm thử ở giai đoạn cơ sở. Đối với tác vụ phân đoạn thực thể, chúng tôi đạt được 4.46% và 5.46% AP lần lượt cho hàm mất mát ba thành phần và bộ nhớ lưu trữ thực thể. Đối với tác vụ phát hiện đối tượng, chúng tôi đạt được 4.04% và 4.50% AP lần lượt cho hai đề xuất. Tóm lại, mô hình cơ sở được chọn cho hiệu suất cao hơn khoảng 1% đến 2% với các chỉ số đánh giá như trong bảng số liệu. Có thể giải thích rằng, mô hình cơ sở với dữ liệu từ COCO-80 chứa nhiều khái niệm ngữ nghĩa hơn so với

mô hình cơ sở với dữ liệu COCO-60 (nhiều hơn 20 lớp dữ liệu), dẫn đến hiệu suất cao hơn. Lưu ý rằng tất cả các kết quả trong phần thực nghiệm loại suy này được thực hiện cho cấu hình 1 mẫu dữ liệu huấn luyện (1-shot).

BẢNG 4.6: Thực nghiệm loại suy trên trọng số  $\alpha$  và tham số *margin* của hàm mất mát ba thành phần ở cấp độ thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*.

AP		Phân đoạn thực thể					Phát hiện đối tượng				
$\alpha$	Margin	0	0.25	0.50	0.75	1.00	0	0.25	0.50	0.75	1.00
		1	3.89	4.50	3.92	<b>5.16</b>	4.43	3.34	3.65	3.22	<b>4.22</b>
	$1 \times 10^{-1}$	<b>4.82</b>	<i>4.74</i>	4.46	<i>4.58</i>	<b>4.57</b>	<b>4.36</b>	<b>4.27</b>	<i>4.04</i>	<i>4.16</i>	<b>3.79</b>
	$1 \times 10^{-2}$	<i>4.29</i>	<b>4.74</b>	<b>4.69</b>	4.46	<i>4.39</i>	<i>4.02</i>	<i>3.97</i>	<b>4.24</b>	4.06	3.71

### Thực nghiệm loại suy trên hàm mất mát ba thành phần ở cấp độ thực thể

Về hàm mất mát ba thành phần ở cấp độ thực thể được mô tả trong Công thức 4.1, chúng tôi thiết lập các thí nghiệm loại suy để đánh giá độ chính xác của mô hình với các cấu hình khác nhau của tham số *margin* và giá trị trọng số  $\alpha$ . Để làm điều này, chúng tôi thiết lập *margin* thay đổi từ 0 đến 1, với bước nhảy là 0.25. Đối với tỷ lệ  $\alpha$  của hàm mất mát (Công thức 4.3), chúng tôi kiểm thử  $\alpha = \{1, 1 \times 10^{-1}, 1 \times 10^{-2}\}$ . Về ý nghĩa, giá trị *margin* chỉ ra sự phân biệt giữa các đặc trưng vùng nền và thực thể. Trong khi đó, trọng số  $\alpha$  điều khiển mức độ ảnh hưởng của hàm mất mát ba thành phần ở cấp độ thực thể đối với hàm mất mát tổng của mô hình FS-CDIS. Bảng 4.6 trình bày đánh giá cả hai tác vụ phát hiện và phân đoạn thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Vậy, ảnh hưởng của trọng số  $\alpha$  quyết định *margin* nào nên được chọn cho hàm mất mát ba thành phần ở cấp độ thực thể. Với  $\alpha = 1$  có nghĩa là chúng tôi giữ nguyên tỷ lệ của hàm mất mát, kết quả phân đoạn trong cấu hình 1 mẫu dữ liệu cho hiệu suất cao nhất là 5.16% mAP với giá trị *margin* là 0.75. Trong khi đó, kết quả phát hiện đối tượng có hiệu suất cao nhất là 4.36% với  $\alpha = 1 \times 10^{-1}$  và không có *margin*. Bảng này cung cấp phân tích cụ thể về tác động của trọng số  $\alpha$  và tham số *margin* đối với hiệu suất tổng thể của mô hình.

### Thực nghiệm loại suy trên bộ nhớ lưu trữ

Như đã giới thiệu trong Công thức 4.2 và Công thức 4.3, bộ nhớ lưu trữ thực thể có một số tham số cần được phân tích, như là dung lượng của bộ nhớ lưu trữ *capacity* và

BẢNG 4.7: Thục nghiệm loại suy trên tham số sức chứa *capacity* của bộ nhớ lưu trữ thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*.

Capacity	Phân đoạn thực thể			Phát hiện đối tượng		
	AP	AP50	AP75	AP	AP50	AP75
32	4.56	7.30	5.02	3.85	7.72	2.91
64	4.51	7.67	4.49	3.94	8.37	2.84
128	4.53	7.55	4.87	4.13	7.98	3.62
256	4.56	7.50	5.02	4.01	8.22	3.39
512	<b>4.76</b>	7.57	<b>5.37</b>	<b>4.48</b>	8.25	<b>4.44</b>
1024	4.72	<b>8.06</b>	5.20	4.14	<b>8.45</b>	3.79

trọng số  $\beta$  điều khiển mức độ ảnh hưởng của hàm mất mát bộ nhớ lưu trữ trong hàm mất mát tổng. **Bảng 4.7** và **Bảng 4.8** trình bày kết quả thực nghiệm loại suy của những yếu tố nói trên. Về dung lượng của bộ nhớ lưu trữ, chúng tôi thiết lập các thực nghiệm trên một phạm vi dung lượng bộ nhớ là  $2^i$  với  $i = \{5, 6, 7, 8, 9, 10\}$ . Các kết quả được báo cáo cho thấy độ chính xác trên cả hai tác vụ phân đoạn thực thể và phát hiện đối tượng đều tăng lên với dung lượng bộ nhớ lớn hơn. Cụ thể, với dung lượng 512, chỉ số mAP đạt được giá trị cao nhất trong số các cấu hình, tức là 4.76% và 4.48% cho phân đoạn thực thể và phát hiện đối tượng. Những kết quả này biểu hiện sự hiệu quả của các phương pháp của chúng tôi đề xuất trong bối cảnh ít mẫu thực thể nguy trang được dùng cho quá trình huấn luyện. Cả hai hàm mất mát đều tăng khả năng phân biệt giữa các đặc trưng nền và thực thể, giúp mô hình phân đoạn các điểm ảnh thuộc về các loài động vật nguy trang tốt hơn. Đối với bộ nhớ lưu trữ thực thể và hàm mất mát ba thành phần ở cấp độ thực thể, kết quả của bộ nhớ lưu trữ cao hơn kết quả của hàm mất mát ba thành phần khoảng 1%. Chúng tôi nhận ra rằng việc lưu trữ các đại diện cho mỗi lớp là một yếu tố quan trọng trong hướng tiếp cận học ít dữ liệu. Kỹ thuật này không chỉ mở rộng các biến thể trong quá trình huấn luyện mà còn tăng tính nhất quán cho mỗi lớp ngữ nghĩa, do đó mô hình có thể phân đoạn các thực thể nguy trang tốt hơn. Ngoài ra, **Bảng 4.8** biểu thị sự hiệu quả của bộ nhớ lưu trữ đối với hàm mất mát tổng. Có thể suy ra, trọng số  $\beta = 1 \times 10^{-4}$  cho kết quả hiệu suất cao nhất được đánh giá trên độ đo mAP, AP50 và AP75 trong số tất cả các cấu hình được thử nghiệm.

BẢNG 4.8: Kết quả thực nghiệm loại suy trên trọng số  $\beta$  của bộ nhớ lưu trữ thực thể trên cấu hình 1 mẫu dữ liệu huấn luyện. Kết quả tốt thứ nhất và thứ hai được tô màu **đỏ**, và *xanh*.

$\beta$	Phân đoạn thực thể			Phát hiện đối tượng		
	AP	AP50	AP75	AP	AP50	AP75
$1 \times 10^{-1}$	3.36	6.58	2.91	3.69	8.02	2.90
$1 \times 10^{-2}$	<i>4.57</i>	<i>8.02</i>	<i>4.74</i>	3.73	7.78	2.76
$1 \times 10^{-3}$	4.51	7.15	4.67	3.87	7.16	3.51
$1 \times 10^{-4}$	<b>5.12</b>	<b>8.71</b>	<b>5.54</b>	<b>4.58</b>	<b>9.23</b>	<b>3.69</b>
$1 \times 10^{-5}$	4.44	7.58	3.89	<i>4.06</i>	<i>7.99</i>	<i>3.63</i>

## 4.5 Tạm kết

Trong phần này, chúng tôi trình bày đề xuất để khai thác hiệu quả đặc trưng mang tính phân biệt cao của thực thể nguy trang trong bối cảnh sử dụng ít mẫu dữ liệu huấn luyện. Với mô hình FS-CDIS, chúng tôi giải quyết bài toán phân đoạn thực thể nguy trang với hướng tiếp cận sử dụng ít dữ liệu huấn luyện. Cụ thể, chúng tôi đã đề xuất hàm mất mát ba thành phần ở cấp độ thực thể và bộ nhớ lưu trữ thực thể để tăng cường khả năng nắm bắt thông tin về thực thể nguy trang của mô hình với ít dữ liệu huấn luyện. Đồng thời, chúng tôi cũng công bố tập dữ liệu CAMO-FS, một trong những tập dữ liệu đầu tiên cho bài toán phân đoạn thực thể nguy trang, được tạo lập cấu trúc cho hướng tiếp cận sử dụng ít dữ liệu huấn luyện. Các thực nghiệm chính và thực nghiệm loại suy chúng tôi thực hiện đã chứng minh hiệu suất các phương pháp đề xuất, và đưa ra một cái nhìn chi tiết để chứng minh cho các lý thuyết hay các lập luận trong đề xuất của chúng tôi. Các đề xuất trong phần này được được tổng hợp thành công trình khoa học và bản tóm tắt đã được công bố tại hội thảo CV4Animals - Computer Vision for Animal Behavior Tracking and Modeling (CVPRW2022) [CT2], toàn văn công trình đã nộp và đang chờ phê duyệt tại tạp chí SIViP - Signal, Image and Video Processing [CT3]. Trong tương lai, với hướng tiếp cận này, chúng tôi mong muốn có thể mở rộng với nhiều mẫu huấn luyện hơn hoặc tăng cường đặc trưng về ngữ cảnh để cải thiện độ chính xác phân đoạn thực thể nguy trang.





HÌNH 4.7: Kết quả so sánh định tính giữa mô hình baseline MTFA [20] và các đề xuất của chúng tôi. Kết quả được lấy từ cấu hình 5 mẫu huấn luyện. “Memory” là Instance Memory Storage và “Triplet” là Instance Triplet Loss. Các thực thể được dự đoán với ngưỡng tin cậy là 0.5, số lượng dự đoán không đáng tin cậy phần lớn đã được loại bỏ. Hai hàng cuối cùng thể hiện các trường hợp mà hàm mất mát ba thành phần hoặc bộ nhớ lưu trữ thực thể chưa giải quyết được.

## Chương 5

# KẾT LUẬN

### 5.1 Kết quả đạt được

Như đã giới thiệu ở phần mở đầu, bài toán phân đoạn thực thể nguy trang là một bài toán có tiềm năng ứng dụng thực tiễn để phục vụ cuộc sống của con người. Bằng việc khai thác các đặc trưng nguy trang một cách hiệu quả, chúng tôi cải thiện độ chính xác của các mô hình học sâu khi thực hiện tác vụ phân đoạn thực thể nguy trang, phần nào giải quyết được tác vụ khó khăn này. Chúng tôi mong rằng các kết quả nghiên cứu này đóng góp một bước phát triển của nhánh nghiên cứu đặc thù trên đối tượng nguy trang nói chung và góp phần rút ngắn khoảng cách từ lý thuyết đến ứng dụng thực tiễn của bài toán này.

Tổng kết lại, trong đề tài này, chúng tôi nghiên cứu giải quyết bài toán phân đoạn thực thể nguy trang với hướng tiếp cận khai thác hiệu quả các đặc trưng có tính phân biệt cao. Ở chương Giới thiệu đề tài, chúng tôi đã trình bày tổng quan về bài toán với bối cảnh thực tiễn, phát biểu bài toán cùng với mục tiêu, lý do thực hiện đề tài và nêu quan điểm, lập luận của chúng tôi khi giải quyết bài toán này với hướng tiếp cận khai thác đặc trưng phân biệt. Kế đến, chúng tôi trình bày Các công trình liên quan với tổng quan về các nghiên cứu trên thực thể nguy trang, các kiến trúc mô hình có liên quan đến bài toán nghiên cứu và các tập dữ liệu đặc thù được công bố trước đây. Chương 3 và Chương 4 trình bày chi tiết lý thuyết và thực nghiệm của các phương pháp do chúng tôi nghiên cứu, đề xuất, gồm có mô hình Transformer một giai đoạn CE-OST khai thác hiệu quả đặc trưng biên cạnh, mô hình FS-CDIS đặc thù cho ngữ cảnh học với ít dữ liệu huấn luyện,



và tập dữ liệu CAMO-FS tiên phong cho nghiên cứu bài toán phân đoạn thực thể nguy trang sử dụng ít dữ liệu huấn luyện. Bên cạnh các thực nghiệm chính, với mỗi phương pháp đề xuất, chúng tôi cũng thực hiện các thực nghiệm loại suy để chứng minh tính hiệu quả của các yếu tố cấu thành phương pháp của chúng tôi.

Qua các đóng góp đã trình bày trong luận văn này, chúng tôi công bố các công trình khoa học rộng rãi trong cộng đồng nghiên cứu quốc tế với các điểm chính sau:

- Một là, nghiên cứu, đề xuất mô hình CE-OST [CT1] dựa trên kiến trúc Transformer tăng cường đặc trưng biên cạnh (contour emphasis) trên thực thể nguy trang.
- Hai là, nghiên cứu, đề xuất mô hình FS-CDIS [CT2, CT3] đặc thù cho ngữ cảnh ít dữ liệu để giải quyết bài toán phân đoạn thực thể nguy trang dựa trên kỹ thuật học tương phản (contrastive learning) với hàm loss ba thành phần (triplet loss) và sử dụng bộ nhớ lưu trữ (memory storage).
- Ba là, đề xuất tập dữ liệu ảnh CAMO-FS [CT2, CT3] cho bài toán phát hiện và phân đoạn thực thể nguy trang, được tinh chỉnh và tạo lập cấu trúc cho hướng tiếp cận học ít dữ liệu.

Tuy đạt được một số cải tiến nhất định, hướng nghiên cứu chúng tôi theo đuổi còn tồn tại nhiều thách thức cần được giải quyết. Đó là các thách thức liên quan đến dữ liệu và thách thức liên quan đến mô hình học. Về vấn đề thiếu dữ liệu, các tác vụ trên thực thể nguy trang gặp khó khăn khi thu thập dữ liệu thực tế từ các loài động vật nguy trang và tiêu tốn nhiều chi phí để gán nhãn chi tiết phục vụ cho bài toán đặc thù này. Bên cạnh đó, chúng tôi cũng gặp phải thách thức liên quan đến các mô hình phân đoạn thực thể hiện nay, tuy thể hiện tốt trên miền dữ liệu tổng quát, chúng chưa được tinh chỉnh để cho độ chính xác tốt khi dự đoán trên đối tượng nguy trang. Hay nói cách khác, các mô hình chưa khai thác tốt yếu tố nguy trang đặc thù của các thực thể này.

## 5.2 Hướng phát triển

Dựa trên những kết quả và kinh nghiệm từ nghiên cứu này, chúng tôi đề xuất một số hướng phát triển đề tài sau đây:

### 5.2.1 Cải tiến các đặc trưng có tính phân biệt cao

Trong luận văn này, chúng tôi đề xuất mô hình CE-OST tập trung vào khai thác hiệu quả các đặc trưng ở vùng biên cạnh vật thể. Việc tăng cường các đặc trưng ở vùng biên cạnh giúp tách biệt rõ ràng vùng nền và vùng thực thể, từ đó giúp mô hình học có khả năng đưa ra dự đoán tốt hơn. Tuy nhiên, các đặc trưng có tính phân biệt cao không chỉ giới hạn ở các đặc trưng ở vùng biên cạnh. Vậy, việc khai thác các loại đặc trưng có tính phân biệt cao khác cũng có tiềm năng giúp cải thiện hiệu suất cho bài toán phân đoạn thực thể nguy trang. Các đặc trưng có thể xem xét như đặc trưng đặc thù của mỗi thực thể thuộc cùng một lớp ngữ nghĩa, hay đặc trưng đặc thù của vùng nền. Yếu tố ngữ cảnh có thể được xem xét để bổ sung đặc trưng cho quá trình phân đoạn. Ngoài ra, việc thiết kế kiến trúc mô hình mạng đặc thù cho việc rút trích đặc trưng nguy trang cũng có tiềm năng để phát triển trong tương lai.

### 5.2.2 Áp dụng hướng tiếp cận cho bài toán trên ảnh y khoa

Các đặc trưng nguy trang là các đặc trưng giúp cho các thực thể nguy trang ẩn mình vào môi trường xung quanh, ngăn cản sự phát hiện của thị giác hay các mô hình xử lý ảnh. Nhận thấy các đặc trưng này không chỉ tồn tại ở các loài động vật nguy trang, các thực thể trong ảnh y khoa (ảnh RGB) như các bộ phận nội tạng, các vùng bệnh hay các khối u cũng chứa các loại đặc trưng tương tự khi có màu sắc hay hình dáng tương đồng với các bộ phận, tế bào khác. Vì thế, chúng tôi kỳ vọng hướng tiếp cận sử dụng đặc trưng phân biệt có thể được áp dụng để tăng cường đặc trưng của các thực thể cần phân đoạn trên ảnh y khoa.

# CÔNG BỐ KHOA HỌC

- [CT1] Thanh-Danh Nguyen, Duc-Tuan Luu, Vinh-Tiep Nguyen, and Thanh Duc Ngo. “CE-OST: Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation”. In: *2023 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*. IEEE. 2023, pp. 1–6.
- [CT2] Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Nhat-Duy Nguyen, Vinh-Tiep Nguyen, Thanh Duc Ngo, Thanh-Toan Do, Minh-Triet Tran, and Tam V Nguyen. “Few-shot Camouflaged Animal Detection and Segmentation”. In: *CV4Animals: Computer Vision for Animal Behavior Tracking and Modeling, in conjunction with Computer Vision and Pattern Recognition 2022* (2022).
- [CT3] Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Nhat-Duy Nguyen, Vinh-Tiep Nguyen, Thanh Duc Ngo, Thanh-Toan Do, Minh-Triet Tran, and Tam V Nguyen. “The Art of Camouflage: Few-shot Learning for Animal Detection and Segmentation”. In: *Signal, Image and Video Processing (Under Review)* (2023).

# Tài liệu tham khảo

- [1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. “YOLACT: Real-time Instance Segmentation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
- [2] Zhaowei Cai and Nuno Vasconcelos. “Cascade R-CNN: Delving into High Quality Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.
- [3] Sema Candemir and Sameer Antani. “A review on lung boundary detection in chest X-rays”. In: *International Journal of Computer Assisted Radiology and Surgery* 14 (2019), pp. 563–576.
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. “End-to-end object detection with transformers”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2020, pp. 213–229.
- [5] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. “Blendmask: Top-down meets bottom-up for instance segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8573–8581.
- [6] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. “Hybrid task cascade for instance segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4974–4983.

- [7] Shuhan Chen, Xiuli Tan, Ben Wang, and Xuelong Hu. “Reverse attention for salient object detection”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 234–250.
- [8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A simple framework for contrastive learning of visual representations”. In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2020, pp. 1597–1607.
- [9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. “Improved baselines with momentum contrastive learning”. In: *arXiv preprint arXiv: 2003.04297* (2020).
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE. 2009.
- [11] Nanqing Dong and Eric P Xing. “Few-shot semantic segmentation with prototype learning.” In: *Proceedings of the British Machine Vision Conference*. Vol. 3. 4. 2018.
- [12] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. “Camouflaged object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2777–2787.
- [13] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. “PRANET: Parallel reverse attention network for polyp segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2020, pp. 263–273.
- [14] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. “Few-shot object detection with attention-RPN and multi-relation detector”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

- [15] Zhibo Fan, Jin-Gang Yu, Zhihao Liang, Jiarong Ou, Changxin Gao, Gui-Song Xia, and Yuanqing Li. “FGN: Fully guided network for few-shot instance segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9172–9181.
- [16] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. “EVA: Exploring the limits of masked visual representation learning at scale”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Oct. 2023.
- [17] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. “Instances as queries”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 6910–6919.
- [18] J. Gallego and P. Bertolino. “Foreground object segmentation for moving camera sequences based on foreground-background probabilistic models and prior probability maps”. In: *Proceedings of the IEEE International Conference on Image Processing*. Oct. 2014, pp. 3312–3316.
- [19] M. Galun, E. Sharon, R. Basri, and A. Brandt. “Texture segmentation by multi-scale aggregation of filter responses and shape elements”. In: *Proceedings of the IEEE International Conference on Computer Vision*. Oct. 2003, pp. 716–723.
- [20] Dan Andrei Ganea, Bas Boom, and Ronald Poppe. “Incremental Few-Shot Instance Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2021, pp. 1185–1194.
- [21] Bin-Bin Gao, Xiaochen Chen, Zhongyi Huang, Congchong Nie, Jun Liu, Jinxiang Lai, Guannan Jiang, Xi Wang, and Chengjie Wang. “Decoupling Classifier for Boosting Few-shot Object Detection and Instance Segmentation”. In: *Advances in Neural Information Processing Systems*. 2022.
- [22] Spyros Gidaris and Nikos Komodakis. “Dynamic few-shot visual learning without forgetting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4367–4375.

- [23] Ruohao Guo, Dantong Niu, Liao Qu, and Zhenbo Li. “SOTR: Segmenting objects with transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 7157–7166.
- [24] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. “A survey on vision transformer”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2022), pp. 87–110.
- [25] Yue Han, Jiangning Zhang, Zhucun Xue, Chao Xu, Xintian Shen, Yabiao Wang, Chengjie Wang, Yong Liu, and Xiangtai Li. “Reference Twice: A Simple and Unified Baseline for Few-Shot Instance Segmentation”. In: *arXiv preprint arXiv: 2301.01156* (2023).
- [26] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.
- [27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. “Momentum contrast for unsupervised visual representation learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9729–9738.
- [28] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969.
- [29] Jianqin Yin Yanbin Han Wendi Hou and Jinping Li. “Detection of the mobile object with camouflage color under dynamic background based on optical flow”. In: *Procedia Engineering* 15 (2011), pp. 2201–2205.
- [30] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. “Mask Scoring R-CNN”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.



- [31] Jyh-Jing Hwang and Tyng-Luh Liu. “Pixel-wise deep learning for contour detection”. In: *Proceedings of the International Conference on Learning Representations* (2015).
- [32] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. “Few-shot object detection via feature reweighting”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
- [33] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. “Mask transfiner for high-quality instance segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
- [34] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. “Transformers in vision: A survey”. In: *ACM Computing Surveys (CSUR)* 54.10s (2022), pp. 1–41.
- [35] Hala Lamdouar, Charig Yang, Weidi Xie, and Andrew Zisserman. “Betrayed by Motion: Camouflaged Object Discovery via Motion Segmentation”. In: *Proceedings of the Asian Conference on Computer Vision*. Nov. 2020.
- [36] Trung-Nghia Le, Yubo Cao, Tan-Cong Nguyen, Minh-Quan Le, Khanh-Duy Nguyen, Thanh-Toan Do, Minh-Triet Tran, and Tam V Nguyen. “Camouflaged Instance Segmentation In-the-Wild: Dataset, Method, and Benchmark Suite”. In: *IEEE Transactions on Image Processing* 31 (2022), pp. 287–300.
- [37] Trung-Nghia Le, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. “Open-Forensics: Large-Scale Challenging Dataset For Multi-Face Forgery Detection And Segmentation In-The-Wild”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2021.
- [38] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. “Anabranh Network for Camouflaged Object Segmentation”. In: *Computer Vision and Image Understanding* 184 (2019), pp. 45–56.

- [39] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. “Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Oct. 2023.
- [40] Shuai Li, Dinei Florencio, Yaqin Zhao, Chris Cook, and Wanqing Li. “Foreground detection in camouflaged scenes”. In: *Proceedings of the IEEE International Conference on Image Processing*. IEEE. 2017, pp. 4247–4251.
- [41] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2017.
- [42] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. “Focal loss for dense object detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2980–2988.
- [43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. “Microsoft COCO: Common Objects in Context”. In: *Proceedings of the European Conference on Computer Vision*. 2014, pp. 740–755.
- [44] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. “Path Aggregation Network for Instance Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.
- [45] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. “CRNet: Cross-Reference Networks for Few-Shot Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2020.
- [46] Z Liu, Y Lin, Y Cao, H Han, Y Wei, Z Zhang, S Lin, and B Guo. “Hierarchical ViT using shifted windows”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.

- [47] Z. Liu, K. Huang, and T. Tan. “Foreground Object Detection Using Top-Down Information Based on EM Framework”. In: *IEEE Transactions on Image Processing* 21.9 (Sept. 2012), pp. 4204–4217.
- [48] Naisong Luo, Yuwen Pan, Rui Sun, Tianzhu Zhang, Zhiwei Xiong, and Feng Wu. “Camouflaged Instance Segmentation via Explicit De-Camouflaging”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17918–17927.
- [49] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. “Simultaneously localize, segment and rank the camouflaged objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11591–11601.
- [50] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. “Convolutional Oriented Boundaries”. In: *Proceedings of the European Conference on Computer Vision*. 2016.
- [51] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. “Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 819–833.
- [52] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics”. In: *Proceedings of the IEEE International Conference on Computer Vision*. Vol. 2. IEEE. 2001, pp. 416–423.
- [53] David R Martin, Charless C Fowlkes, and Jitendra Malik. “Learning to detect natural image boundaries using local brightness, color, and texture cues”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.5 (2004), pp. 530–549.

- [54] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. “Camouflaged object segmentation with distraction mining”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- [55] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. “The role of context for object detection and semantic segmentation in the wild”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2014, pp. 891–898.
- [56] Khoi Nguyen and Sinisa Todorovic. “iFS-RCNN: An Incremental Few-shot Instance Segmenter”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 7010–7019.
- [57] A. Wahi P. Sengottuvelan and A. Shanmugam. “Performance of Decamouflaging Through Exploratory Image Analysis”. In: *Proceedings of the International Conference on Emerging Trends in Engineering and Technology*. 2008.
- [58] Yuxin Pan, Yiwang Chen, Qiang Fu, Ping Zhang, and Xin Xu. “Study on the camouflaged target detection method based on 3D convexity”. In: *Modern Applied Science* 5.4 (2011), p. 152.
- [59] Jialun Pei, Tianyang Cheng, Deng-Ping Fan, He Tang, Chuanbo Chen, and Luc Van Gool. “OSFormer: One-Stage Camouflaged Instance Segmentation with Transformers”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2022.
- [60] Erik Learned-Miller Pia Bideau. “It’s Moving! A Probabilistic Model for Causal Motion Segmentation in Moving Camera Videos”. In: *Proceedings of the European Conference on Computer Vision*. 2016.
- [61] Thomas W Pike. “Quantifying camouflage and conspicuousness using visual salience”. In: *Methods in Ecology and Evolution* 9.8 (2018).

- [62] Joseph Redmon and Ali Farhadi. “YOLO9000: better, faster, stronger”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7263–7271.
- [63] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In: *Advances in Neural Information Processing Systems* 28 (2015).
- [64] Lixiang Ru, Heliang Zheng, Yibing Zhan, and Bo Du. “Token contrast for weakly-supervised semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3093–3102.
- [65] Oindrila Saha, Zezhou Cheng, and Subhansu Maji. “GanOrCon: Are Generative Models Useful for Few-Shot Segmentation?” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2022, pp. 9991–10000.
- [66] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2015, pp. 815–823.
- [67] E. Shelhamer, J. Long, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016).
- [68] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. “Indoor segmentation and support inference from rgb-d images”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2012, pp. 746–760.
- [69] P. Skurowski, H. Abdulameer, J. Baszczyk, T. Depta, A. Kornacki, and P. Kozie. “Animal camouflage analysis: Chameleon database”. In: (2018).
- [70] Jake Snell, Kevin Swersky, and Richard Zemel. “Prototypical networks for few-shot learning”. In: *Advances in Neural Information Processing Systems* 30 (2017).

- [71] Zhi Tian, Chunhua Shen, and Hao Chen. “Conditional Convolutions for Instance Segmentation”. In: *Proceedings of the European Conference on Computer Vision*. 2020.
- [72] Zhuotao Tian, Xin Lai, Li Jiang, Shu Liu, Michelle Shu, Hengshuang Zhao, and Jiaya Jia. “Generalized Few-Shot Semantic Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2022, pp. 11563–11572.
- [73] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* 30 (2017).
- [74] Haochen Wang, Jie Liu, Yongtuo Liu, Subhansu Maji, Jan-Jakob Sonke, and Efstratios Gavves. “Dynamic Transformer for Few-shot Instance Segmentation”. In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 2969–2977.
- [75] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. “PANet: Few-Shot Image Semantic Segmentation With Prototype Alignment”. In: *Proceedings of the IEEE International Conference on Computer Vision*. Oct. 2019.
- [76] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. “PVT v2: Improved baselines with pyramid vision transformer”. In: *Computational Visual Media* 8.3 (2022), pp. 415–424.
- [77] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. “Frustratingly Simple Few-Shot Object Detection”. In: *Proceedings of the International Conference on Machine Learning*. 2020.
- [78] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. “SOLO: Segmenting Objects by Locations”. In: *Proceedings of the European Conference on Computer Vision*. 2020.

- [79] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. “Cbam: Convolutional block attention module”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 3–19.
- [80] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. *Detectron2*. <https://github.com/facebookresearch/detectron2>. 2019.
- [81] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. “Unsupervised feature learning via non-parametric instance discrimination”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3733–3742.
- [82] Saining Xie and Zhuowen Tu. “Holistically-nested edge detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1395–1403.
- [83] Feng Xue, Chengxi Yong, Shan Xu, Hao Dong, Yuetong Luo, and Wei Jia. “Camouflage performance analysis and evaluation framework based on features fusion”. In: *Multimedia Tools and Applications* 75 (2016), pp. 4065–4082.
- [84] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. “Meta R-CNN: Towards general solver for instance-level low-shot learning”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019.
- [85] Jianqin Yin, Yanbin Han, Wendi Hou, and Jinping Li. “Detection of the Mobile Object with Camouflage Color Under Dynamic Background Based on Optical Flow”. In: *Procedia Engineering* 15 (2011), pp. 2201–2205.
- [86] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. “Mutual Graph Learning for Camouflaged Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2021.



- 
- [87] Hao Zhang, Feng Li, Huaizhe Xu, Shijia Huang, Shilong Liu, Lionel M Ni, and Lei Zhang. “MP-Former: Mask-piloted transformer for image segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 18074–18083.
- [88] Jinchao Zhu, Xiaoyu Zhang, Shuo Zhang, and Junnan Liu. “Inferring Camouflage Objects by Texture-Aware Interactive Guidance Network”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021.
- [89] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. “Deformable DETR: Deformable transformers for end-to-end object detection”. In: *Proceedings of the International Conference on Learning Representations* (2020).

# CE-OST: Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation

Thanh-Danh Nguyen<sup>1,2</sup>, Duc-Tuan Luu<sup>1,2</sup>, Vinh-Tiep Nguyen<sup>†1,2</sup>, and Thanh Duc Ngo<sup>1,2</sup>

<sup>1</sup>University of Information Technology, Ho Chi Minh City, Vietnam

<sup>2</sup>Vietnam National University, Ho Chi Minh City, Vietnam

{*danhnt, tuanld, tiepvt, thanhnd*}@uit.edu.vn, <sup>†</sup>*corresponding author*

**Abstract**—Understanding camouflage images at instance level is such a challenging task in computer vision. Since the camouflage instances have their colors and textures similar to the background, the key to distinguish them in the images should rely on their contours. The contours separate the instance from the background, thus recognizing these contours should break their camouflage mechanism. To this end, we address the problem of camouflage instance segmentation via the Contour Emphasis approach. We improve the ability of the segmentation models by enhancing the contours of the camouflaged instances. We propose the CE-OST framework which employs the well-known architecture of Transformer-based models in a one-stage manner to boost the performance of camouflaged instance segmentation. The extensive experiments prove our contributions over the state-of-the-art baselines on different benchmarks, i.e. CAMO++, COD10K and NC4K.

**Index Terms**—Camouflage Image Understanding, One-Stage Instance Segmentation, Transformer-based model

## I. INTRODUCTION

In nature, animals conceal themselves from their enemies by blending in with their environment [1]. That is the so-called camouflaged animals or camouflaged instances. Various applications can be considered via autonomously distinguishing those instances e.g., search-and-rescue missions, wild species discovery, preservation campaign [2] and media forensics (manipulated image/video detection and segmentation [3]); or even in the medical domain where the tumors, polyps or cells seems to be similar to other cells [4]–[6]. Despite the long development history of image segmentation methods, general segmentation models cannot deal well with camouflaged instances [7]–[10]. At first, traditional approaches [11]–[18] utilized low-level handcrafted features for simple camouflaged images. Recent models with a learning-based approach rely on large-scale datasets [2], [19] to address camouflaged object segmentation.

The concealing mechanism of camouflage animals aims to merge their texture and color with the surroundings. Therefore, human vision fails to recognize those instances at a glance. However, such boundaries of the camouflaged animals can hardly be disappeared. In this work, based on the biological assumption, we propose a method to enhance the boundary of the camouflaged instance with the purpose of supporting the segmentation models to segment the instances. Inspired by the work on boundary detection [20]–[23], we enhance the boundary of the camouflaged instances in the images so that

the segmentation model can better differentiate the instances from the background.

Furthermore, recent work on computer vision has achieved a significant explosion in performance since the work [24] on Transformer architecture. Lately, the methods are applied commonly to the computer vision domain [25], [26]. Specifically, the performance of instance segmentation models has also improved thanks to transformer architecture [27]–[30]. In this work, we follow OSFormer [30] which is the first one-stage transformer-based framework designed for this task.

To summarize, in this work, we propose a simple yet effective boundary enhancement module in a plug-and-play manner. We focus on addressing camouflaged image instance segmentation via a one-stage transformer-based model. To this end, we propose the Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation, dubbed CE-OST framework. To prove our method, we conduct experiments on the three well-known benchmarks in this camouflage domain, i.e. CAMO++ [31], COD10K [19], and NC4K [32]. The reported results demonstrate the performance of our proposed method over state-of-the-art baselines.

The rest of this paper is organized as follows. Section II reviews related work on camouflage research, Section III presents our proposed method - CE-OST. In Section IV, extensive experiments and ablation studies prove the effectiveness of our proposal. Finally, Section V concludes our work.

## II. RELATED WORK

### A. Camouflaged Research

Early approaches mainly exploit low-level handcrafted features, including color features, edge, texture, and brightness [33]–[35]. Recent studies have taken advantage of the large capacity of deep networks to recognize more intricate characteristics of camouflage, enhancing the performance of detecting camouflaged objects. Zhai *et al.* [36] utilized a mutual graph learning technique to interactively train the boundaries and regions of camouflaged objects. PFNet [37] was designed to simulate the natural process of predation. Le *et al.* [2] introduced Anabran network that combines both object classification and segmentation. SINet [19] tried to mimic the predators’ hunting behavior by containing two main modules for locating and identifying the camouflaged objects. Lyu *et al.* [32] designed a network that ranks concealed

TABLE I  
COMPARISON AMONG CAMOUFLAGE DATASETS  
(WITHOUT NON-CAMOUFLAGED IMAGES).

Dataset	#Annot. Camo. Img.	#Meta- Cat.	#Obj. Cat.	Bbox. GT	Obj. Mask GT	Ins. Mask GT
CAMO [2]	1,250	2	8	×	✓	×
COD10K [19]	5,066	5	69	✓	✓	✓
NC4K [32]	4,121	5	69	✓	✓	✓
CAMO++ [31]	2,695	10	47	✓	✓	✓

objects while simultaneously localizing and segmenting them to enhance prediction accuracy. A dual-stream MirrorNet [38] was proposed to capture various scene layouts while TINet [39] interactively refined texture and segmentation features at multiple levels. Le *et al.* [31] presented CFL approach which integrates various models via learning image contexts.

### B. Instance Segmentation

Instance segmentation in computer vision involves identifying individual objects and generating their corresponding masks. Existing methods can be divided into two categories: **Two-stage** and **One-stage approach**. Methods in the first group employ a traditional detect-then-segment scheme that initially identifies Regions of Interest (ROIs) using bounding boxes and subsequently generates local pixel-level instance segmentation [40]. Mask RCNN [41], built upon Faster RCNN [42], is a well-known approach that incorporates an additional mask-prediction branch at the instance level. Mask Scoring RCNN [43] includes a MaskIOU head on top of Mask RCNN to evaluate the quality of the predicted instance masks. Cascade Mask RCNN [44] is a multi-stage architecture including a series of detectors trained with increasing IOU thresholds to filter out false positives more effectively. PANet [45] was proposed to shorten information flow and enhance the feature extractor by designing a bottom-up path augmentation. Blend-Mask [46] initially generates dense yet shallow positional sensitive instance features for each pixel. Then, a blender module will merge the features for each instance to produce an attention map. Moreover, Chen *et al.* [47] presented the HTC to combine both detection and segmentation features for joint processing. Recently, DCNet was introduced [48] with a de-camouflaging mechanism to extract the camouflage characteristics.

In the second category, single-stage methods adopt the anchor-free object detection approach. YOLACT [49] is the first method that attempts real-time instance segmentation by combining the results of two parallel tasks: generating a set of non-local prototype masks and predicting per-instance mask coefficients. SOLO [50], [51] redefines instance segmentation as predicting categories then generating masks. This method utilizes semantic categories to locate the center of the instances and separates mask prediction into dynamic kernel feature learning. Consequently, the output masks are generated without the need to compute bounding boxes. CondInst [40] can solve instance segmentation with fully convolutional networks while eliminating the ROI cropping and feature alignment.

### C. Camouflaged Datasets

CHAMELEON [52] and Camouflaged Animals [53] are the first two camouflage datasets providing mask annotations.

However, the sizes of the test datasets are less than 300, which is insufficient for deep learning methods. Regarding object detection task, MoCA dataset [54] was introduced which contains only bounding box ground-truth. In terms of suitable instance segmentation datasets, i.e. [2], [19], [31], [55], we carefully describe in Section IV. Table I provides a comprehensive comparison on our chosen datasets.

## III. PROPOSED METHOD

### A. Overview our CE-OST framework

Our proposed framework of Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation, dubbed **CE-OST**, is illustrated in Figure 1. There are two main blocks: Contour Emphasis Block and Transformer Block. A camouflaged input image should go through the two blocks before meeting the Fusion Module at the end of the framework to return the segmentation mask. Our proposed Contour Emphasis Block can be considered a portable plug-and-play component. The image passing through this block has its boundary enhanced. Then, the enhanced image continues its journey to perform feature extraction via a CNN model. The extracted features join the One-Stage Transformer Block to conduct instance segmentation masks. The details are explained in the following.

### B. Contour Emphasis Method

Boundary plays an important role in supporting our vision to recognize the whole shape of an arbitrary instance or object. Since the work on handcrafted features like Canny Edge Detection to work applied deep methods like HED [22], or COB [20], [21] set the very outstanding performance on edge detection. In this work, we propose a Contour Emphasis approach to enhance the visual features of camouflaged instances to improve the segmentation model. In Figure 1, we present the Contour Emphasis (CE) Block that takes the responsibility of fusing the boundary to the original image.

Originating from HED [22], we adopt a multi-scale convolutional network with a pre-trained VGG-16 backbone [56] to detect the instance boundary. First, the whole CE Block is trained on an edge detection dataset, dubbed BSD500 dataset [57]. The losses are computed at multi-scale with pixel-wise cross-entropy loss and fused into a total loss in the end. To reduce the edge annotation cost, we utilize the trained model to predict the camouflaged image boundary. The contours are then added to the original images to enhance their appearances.

Accordingly, there are several ways of contour combination. Therefore, we empirically choose the two methods, i.e. *color contrast* (1) and *brightness addition* (2). In the brightness addition, we straight forward add the boundary result to the image (pixel-wise value addition). In the color contrast method, we perform addition on the compensation value of the pixel at the corresponding pixel-wise location. In Figure 3, readers can compare the visual differences between the two methods of our contour fusion. To overcome some intensive cases where the texture is too complex, we conduct a simple *grid-condition* procedure (Figure 2). The edge detector may

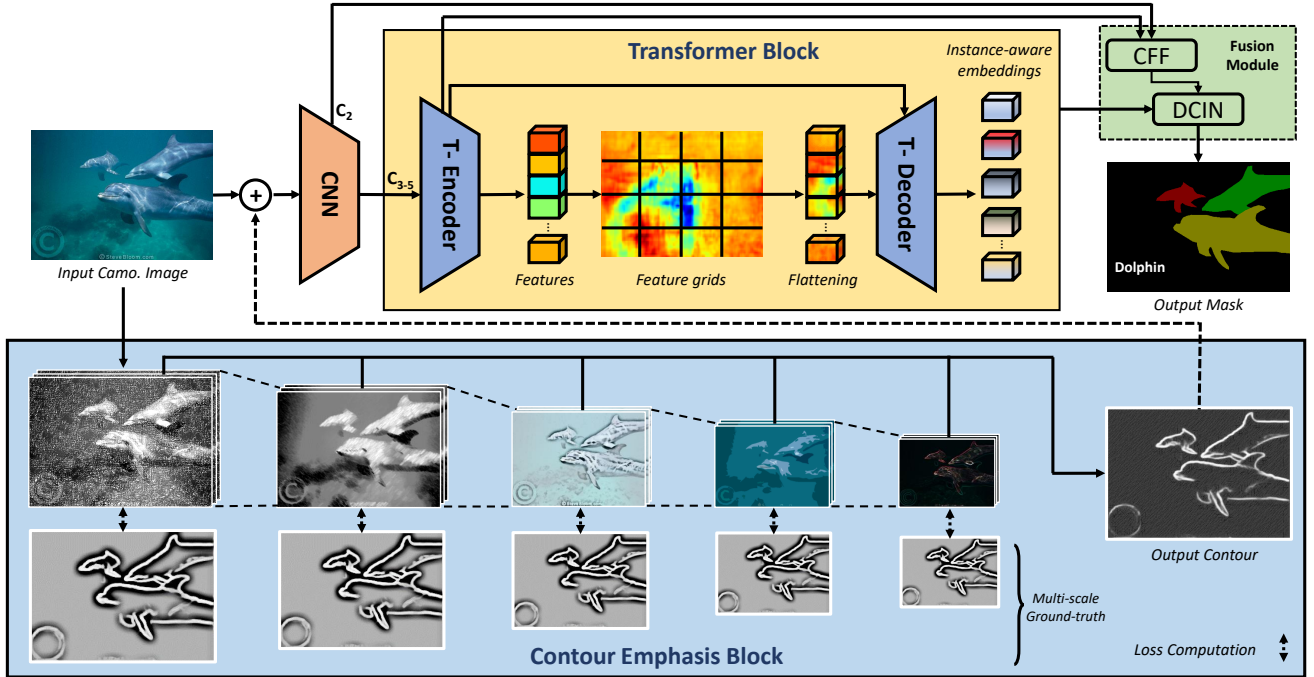


Fig. 1. Overall our CE-OST framework: Contour Emphasis for One-Stage Transformer-based Camouflage Instance Segmentation.

fail, whose results are further noisy than enhancing the camouflaged instances. In this case, we apply a grid  $5 \times 5$  to each image and decide not to apply boundary fusion if the number of eliminated cells is over half (which is equal to approx. 12). A cell is eliminated if the number of pixels in the detected boundary area covers over a half area of the cell. This proposal is adaptive to every single image size of the camouflage dataset.

### C. One-Stage Transformer-based Camouflage Instance Segmentation Model

**Feature Extractor.** Given an input image  $I \in \mathbb{R}^{H \times W \times 3}$ , we adopt a CNN to release multi-scale features  $C_{2-5}$ . The input for the Transformer Block is the flattened features from  $C_{3-5}$  while  $C_2$  is fed into the Fusion Module for a low-level feature enhancement. The details of the backbones are in Section IV.

**Transformer-based Instance Segmentation Model.** We employ the structure of an Encoder-Decoder model [24], [30] since previous work done on Transformers proved the effectiveness of the self-attention layers in extracting global information from the image. In this architecture, the network focuses on the precise location, which is crucial for instance segmentation. Notably, the input of this block is multi-scale, compared to the limited single-scale of DETR [58].

**Fusion Module.** In this Fusion Module, we follow [30] while having Dynamic Camouflaged Instance Normalization (DCIN) and Coarse-to-Fine Fusion (CFF). The CNN feature  $C_2$  and middle features of the T-Encoder are sent to the CFF to create comprehensive features. Then, features from the final layer of the Transformer Block and CFF are inputs of the DCIN. In DCIN, there is a fully-connected layer used to gain the location label. At the same time, a multi-layer perceptron

is employed to gain the instance-aware parameters. These parameters are then used for establishing the segmentation mask. Please visit [30] for more implementation details.

## IV. EXPERIMENTS

### A. Dataset and Settings

In our experiments, we utilize COD10K [19], NC4K [32] and CAMO++ [31] for camouflage instance segmentation. Please see Table I for a more detailed comparison among these datasets. The following paragraphs are their brief reviews.

**COD10K Dataset.** [19] comprises around 10,000 images divided into 5 meta-categories. However, in terms of camouflage, the training set of COD10K contains approximately 3,040 and the test set includes 2,026 images.

**NC4K Dataset.** The NC4K is a testing dataset [32] with 4,121 images of camouflaged instances collected from online resources. We utilize this benchmark to evaluate our proposed method applied to the one-stage transformer-based model.

**CAMO++ Dataset.** The original CAMO++ dataset contains both images of camouflaged and non-camouflaged instances with a total of 5,500 images corresponding to 32,756 instances [31]. There are 47 fine-grained camouflaged classes designed with a hierarchical structure and assigned into 10 coarse-grained classes. CAMO++ contributes 2,695 camouflage images including 1,250 existing images in CAMO [2] and 1,450 newly collected images.

**Experimental Settings.** To select the base models, we employed ResNet-50 [59], ResNet-50 (with input size of  $550 \times 550$  for real-time manner) [59], ResNet-101 [59], Pyramid Vision Transformer (PVT) [60], and Swin Transformer (Swin-T) [61] to apply our proposed method. The framework was built on top of Detectron2 [62] and other models originated from their own publications. Our Contour Emphasis



Fig. 2. Our Grid-Condition to control the Contour Emphasis.

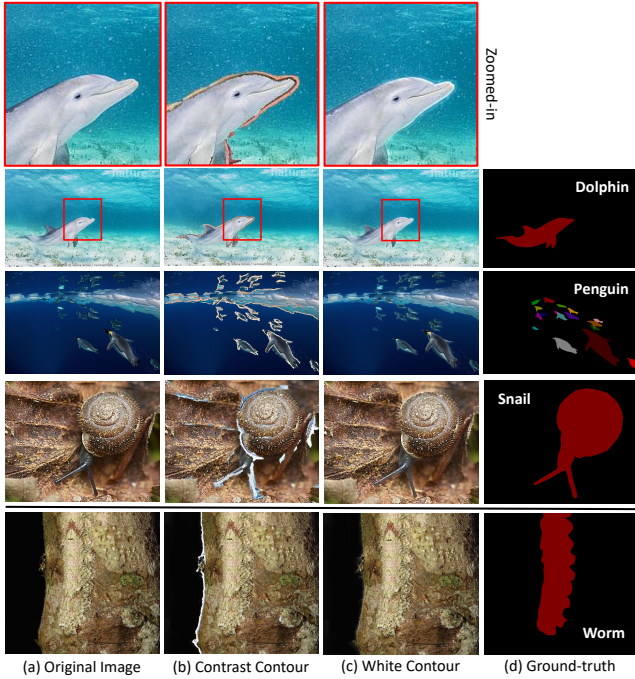


Fig. 3. Exemplary contour emphasized images on CAMO++ [31]. The first rows are the zoomed-in regions. Best viewed online with colors and zoom-in.

framework follows the original configurations [30]. In detail, we adopted a single GeForce RTX 2080Ti GPU and trained with Stochastic Gradient Descent. To initialize the models, we utilized the well-known pre-trained weights of ImageNet [63]. Our training process occurred with 90K iterations with a batch size of 1 and base learning rate of  $2.5e^{-4}$  heading by 1K iterations of warming-up. We also use a learning rate reduction of 0.1 after 60K and 80K iterations. The weight decay and the momentum values are  $10^{-4}$  and 0.9, respectively.

**Evaluation metrics.** To report our results, we use average precision (AP). To be detailed, we report AP, AP@50, and AP@75. Readers can reach this site <https://cocodataset.org/#detection-eval> for more details on the evaluation metrics.

### B. State-of-the-art Comparison

To prove the performance of our proposed method - the Contour Emphasis approach, we reported the established experiments in Table II. With this setting, we utilize the results on COD10K and NC4K datasets to compare the performance

TABLE II  
STATE-OF-THE-ART COMPARISON ON COD10K [19] AND NC4K [32] DATASET. THE CHOSEN BACKBONE IS THE COMMON RESNET-101 [59].

Method		COD10K			NC4K		
		AP	AP50	AP75	AP	AP50	AP75
Two-Stage	Mask R-CNN [41]	28.7	60.1	25.7	36.1	68.9	33.5
	MS R-CNN [43]	33.3	61.0	32.9	35.7	63.4	34.7
	Cascade R-CNN [44]	29.5	61.0	25.9	34.6	66.3	31.5
	HTC [47]	30.9	61.0	28.7	34.2	64.5	31.6
	BlendMask [46]	31.2	60.0	28.9	31.4	61.2	28.8
	Mask Transfuser [64]	31.2	60.7	29.8	34.0	63.1	32.6
One-Stage	YOLOACT [49]	29.0	60.1	25.3	37.8	70.6	35.6
	CondInst [65]	34.3	67.9	31.6	38.0	71.1	35.6
	QueryInst [66]	32.5	65.1	28.6	38.7	72.1	37.6
	SOTR [67]	32.0	63.6	29.2	34.3	65.7	32.4
	SOLOv2 [51]	35.2	65.7	33.4	37.8	69.2	36.1
	OSFormer [30]	42.0	71.3	42.8	44.4	73.7	45.1
	<b>CE-OST (Ours)</b>	<b>43.2</b>	<b>72.2</b>	<b>44.1</b>	<b>45.1</b>	<b>74.0</b>	<b>46.4</b>

among models. To this end, we employed models from two-stage (i.e. [41], [43], [44], [46], [47], [64]) and one-stage approaches (i.e. [30], [49], [51], [65]–[67]). For a fair comparison, ResNet-101 [59], which is the popular backbone utilized by other publications, is utilized. The reported results recognized our improvement on all AP, AP50, and AP75 evaluation metrics. On COD10K [19], we achieved 43.2%, 72.2%, and 44.1% on AP, AP50, and AP75, respectively. On NC4K [32], the three respective values were 45.1%, 74.0%, and 46.4%. To this end, we present the state-of-the-art results over the baseline methods on both one-stage and two-stage approaches. Please find the next ablation section for more empirical details of our Contour Emphasis method.

### C. Ablation Study

In our CE-OST framework, the Contour Emphasis can be applied in two ways. In Table III, we present the effectiveness of color contrasting and brightness addition. We also conducted experiments on different base models including the 5 aforementioned methods in the Experimental Settings section. In general, Transformer-based models such as Swin-T or PVT give the best results among methods. The CAMO++ [31] is the most intensive dataset following by NC4K [32] and COD10K [19]. This ablation study also proves the generality of our CE-OST over the base models when improving almost every result in comparison with the state-of-the-art baselines. Especially to CAMO++ [31] dataset, the PVT backbone stably holds the best performance. The results can be explained as the multi-scale feature extractor of PVT can well handle the various scales of CAMO++. In Figure 4, we present our best results on the PVT backbone (left) and some failure cases (right) of over-segmentation or mislabeling. The instance scale and too complex background cause this phenomenon.



TABLE III  
ABLATION STUDY ON DIFFERENT BASE MODELS ON COD10K [19], NC4K [32], AND CAMO++ [31].

Method	Base-Model	COD10K			NC4K			CAMO++		
		AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
OSFormer	ResNet-50 [59]	41.0	71.1	40.8	42.5	72.5	42.3	19.0	33.8	18.3
	ResNet-50-550 [59]	-	-	-	-	-	-	20.1	36.3	19.3
	ResNet-101 [59]	42.0	71.3	42.8	44.4	73.7	45.1	20.6	34.4	20.2
	PVTv2-B2-Li [60]	47.2	74.9	49.8	-	-	-	27.7	44.7	27.9
	Swin-T [61]	47.7	78.6	49.3	-	-	-	22.3	36.6	21.8
CE-OST (Color Contrast)	ResNet-50 [59]	41.6	70.7	42.3	42.4	71.4	42.6	20.1	34.2	19.6
	ResNet-50-550 [59]	35.9	65.2	34.3	41.1	70.9	41.1	20.6	35.7	20.0
	ResNet-101 [59]	43.2	72.2	44.1	45.1	74.0	46.4	21.7	36.6	21.3
	PVTv2-B2-Li [60]	48.4	75.7	51.3	51.4	77.9	55.0	28.5	45.3	29.9
	Swin-T [61]	49.1	78.0	52.1	50.5	78.9	53.1	22.7	37.6	22.4
CE-OST (Brightness Addition)	ResNet-50 [59]	41.2	69.0	41.6	42.4	71.1	42.9	20.2	34.8	19.5
	ResNet-50-550 [59]	35.9	65.2	34.6	40.8	71.1	40.3	21.0	37.1	20.3
	ResNet-101 [59]	42.4	70.8	43.7	44.2	73.1	45.0	21.1	34.4	20.9
	PVTv2-B2-Li [60]	47.9	74.6	50.5	51.1	77.3	54.9	27.9	45.1	29.2
	Swin-T [61]	49.0	78.5	51.4	50.8	79.3	53.9	22.7	38.4	23.1

\*The first, second, and third best results are marked in red, blue, and green, respectively.

#### D. Discussion

In Figure 3, we present several examples of camouflaged instances with their enhanced boundaries. From left to right, we show the original (a), contrast contour (b), bright contour (c), and ground-truth (d) images, respectively. Both kinds of contours are generated by the Contour Emphasis, and we present these contours under two appearances. The bright contours are the brightness addition to the boundary lines of the instances. While, the contrast contours shift the color values to another value in the contrast range, bringing a better-distinguished contour view compared to bright color contours. Thus, we can observe the first best results major in the Color Contrast approach. The last row illustrates a case where the recognized boundary fails to enhance the visual appearance of the considered instance, i.e. worm. Our future work should focus on camouflage boundary recognition.

#### V. CONCLUSION

In this work, we proposed the CE-OST framework - a Contour Emphasis approach for One-Stage Transformer-based model to address the instance segmentation task on camouflaged images. We have demonstrated the improvement of our proposed method over the three well-known camouflage benchmarks of COD10K, NC4K, and CAMO++. In the future, we plan to extend our idea to other specific domains of medical imaging where the instances carry camouflaged features.

#### VI. ACKNOWLEDGEMENT

This research is funded by University of Information Technology - Vietnam National University Ho Chi Minh City under grant number D1-2023-20. Thanh-Danh Nguyen was funded by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), code VINIF.2022.ThS.104. We also would like to acknowledge Tam V. Nguyen, Assoc. Prof. for his inspiration for camouflage research.

#### REFERENCES

- [1] S. Singh, C. Dhawale, and S. Misra, "Survey of object detection methods in camouflaged image," *IERI Procedia*, vol. 4, pp. 351 – 357, 2013.
- [2] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabran network for camouflaged object segmentation," *CVIU*, vol. 184, pp. 45–56, 2019.
- [3] T.-N. Le, H. H. Nguyen, J. Yamagishi, and I. Echizen, "Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild," in *ICCV*, 2021.
- [4] Q. Zhangli, J. Yi, D. Liu, X. He, Z. Xia, Q. Chang, L. Han, Y. Gao, S. Wen, H. Tang *et al.*, "Region proposal rectification towards robust instance segmentation of biological images," in *MICCAI*. Springer, 2022, pp. 129–139.
- [5] X. Liu, B. Hu, W. Huang, Y. Zhang, and Z. Xiong, "Efficient biomedical instance segmentation via knowledge distillation," in *MICCAI*. Springer, 2022, pp. 14–24.
- [6] C. Li, D. Liu, H. Li, Z. Zhang, G. Lu, X. Chang, and W. Cai, "Domain adaptive nuclei instance segmentation and classification via category-aware feature alignment and pseudo-labelling," in *MICCAI*. Springer, 2022, pp. 715–724.
- [7] C. Kervrann and F. Heitz, "A markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics," *IEEE TIP*, vol. 4, no. 6, pp. 856–862, 1995.
- [8] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *IJCV*, vol. 70, no. 2, pp. 109–131, 2006.
- [9] X. Li and H. Sahbi, "Superpixel-based object class segmentation using conditional random fields," in *ICASSP*, 2011, pp. 1101–1104.
- [10] L. Sulimowicz, I. Ahmad, and A. Aved, "Superpixel-enhanced pairwise conditional random field for semantic segmentation," in *ICIP*, 2018.
- [11] M. Galun, E. Sharon, R. Basri, and A. Brandt, "Texture segmentation by multiscale aggregation of filter responses and shape elements," in *ICCV*, Oct 2003, pp. 716–723.
- [12] L. Song and W. Geng, "A new camouflage texture evaluation method based on wssim and nature image features," in *ICMT*, Oct 2010.
- [13] F. Xue, C. Yong, S. Xu, H. Dong, Y. Luo, and W. Jia, "Camouflage performance analysis and evaluation framework based on features fusion," *MTAP*, vol. 75, pp. 4065–4082, 2016.
- [14] Y. Pan, Y. Chen, Q. Fu, P. Zhang, and X. Xu, "Study on the camouflaged target detection method based on 3d convexity," *Modern Applied Science*, vol. 5, no. 4, p. 152, 2011.
- [15] Z. Liu, K. Huang, and T. Tan, "Foreground object detection using top-down information based on em framework," *IEEE TIP*, vol. 21, no. 9, pp. 4204–4217, Sept 2012.
- [16] A. W. P. Sengottuvelan and A. Shanmugam, "Performance of decamouflaging through exploratory image analysis," in *ICETET*, 2008.
- [17] J. Yin, Y. Han, W. Hou, and J. Li, "Detection of the mobile object with camouflage color under dynamic background based on optical flow," *Procedia Engineering*, vol. 15, pp. 2201 – 2205, 2011.
- [18] J. Gallego and P. Bertolino, "Foreground object segmentation for moving camera sequences based on foreground-background probabilistic models and prior probability maps," in *ICIP*, Oct 2014, pp. 3312–3316.
- [19] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *CVPR*, 2020, pp. 2777–2787.
- [20] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool, "Convolutional oriented boundaries," in *ECCV*, 2016.
- [21] —, "Convolutional oriented boundaries: From image segmentation to high-level tasks," *IEEE TPAMI*, vol. 40, no. 4, pp. 819 – 833, 2018.

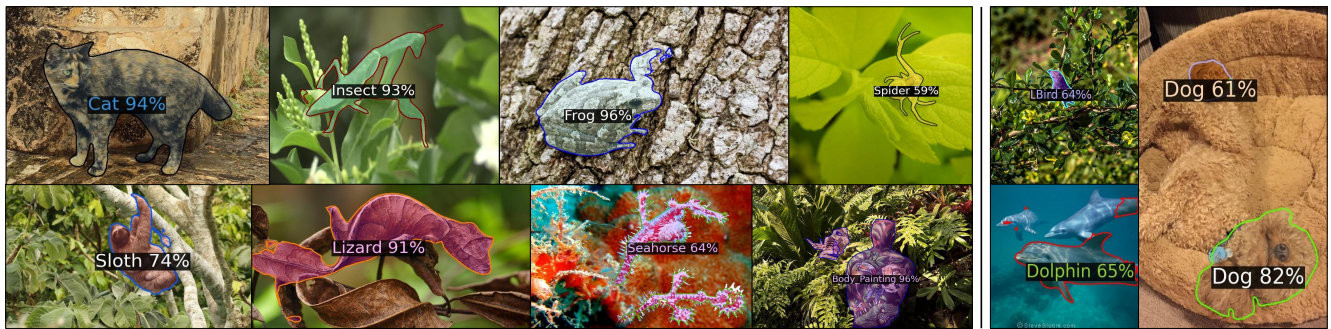


Fig. 4. Quality visualization results on the CAMO++ [31] testing set on our CE-OST-PVT. The confidence threshold is 0.5.

- [22] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015, pp. 1395–1403.
- [23] S. Candemir and S. Antani, "A review on lung boundary detection in chest x-rays," *IJCARS*, vol. 14, pp. 563–576, 2019.
- [24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, vol. 30, 2017.
- [25] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.
- [26] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu *et al.*, "A survey on vision transformer," *IEEE TPAMI*, vol. 45, no. 1, pp. 87–110, 2022.
- [27] Y. Fang, W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao, "Eva: Exploring the limits of masked visual representation learning at scale," in *CVPR*, Oct 2023.
- [28] H. Zhang, F. Li, H. Xu, S. Huang, S. Liu, L. M. Ni, and L. Zhang, "Mp-former: Mask-piloted transformer for image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18074–18083.
- [29] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask dino: Towards a unified transformer-based framework for object detection and segmentation," in *CVPR*, Oct 2023.
- [30] J. Pei, T. Cheng, D.-P. Fan, H. Tang, C. Chen, and L. Van Gool, "Osformer: One-stage camouflaged instance segmentation with transformers," in *ECCV*. Springer, 2022.
- [31] T.-N. Le, Y. Cao, T.-C. Nguyen, M.-Q. Le, K.-D. Nguyen, T.-T. Do, M.-T. Tran, and T. V. Nguyen, "Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite," *IEEE TIP*, vol. 31, pp. 287–300, 2022.
- [32] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *CVPR*, 2021, pp. 11591–11601.
- [33] F. Xue, C. Yong, S. Xu, H. Dong, Y. Luo, and W. Jia, "Camouflage performance analysis and evaluation framework based on features fusion," *MTAP*, vol. 75, no. 7, pp. 4065–4082, 2016.
- [34] S. Li, D. Florencio, Y. Zhao, C. Cook, and W. Li, "Foreground detection in camouflaged scenes," in *ICIP*. IEEE, 2017, pp. 4247–4251.
- [35] T. W. Pike, "Quantifying camouflage and conspicuousness using visual salience," *Methods in Ecology and Evolution*, vol. 9, no. 8, 2018.
- [36] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, and D.-P. Fan, "Mutual graph learning for camouflaged object detection," in *CVPR*, Jun 2021.
- [37] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *CVPR*, 2021.
- [38] J. Yan, T.-N. Le, K.-D. Nguyen, M.-T. Tran, T.-T. Do, and T. V. Nguyen, "Mirronet: Bio-inspired camouflaged object segmentation," *IEEE Access*, vol. 9, pp. 43290–43300, 2021.
- [39] J. Zhu, X. Zhang, S. Zhang, and J. Liu, "Inferring camouflage objects by texture-aware interactive guidance network," in *AAAI*, 2021.
- [40] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *ECCV*, 2020.
- [41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2980–2988.
- [42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *IEEE TPAMI*, 2016.
- [43] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *CVPR*, 2019.
- [44] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, 2018.
- [45] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *CVPR*, 2018.
- [46] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *CVPR*, 2020, pp. 8573–8581.
- [47] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *CVPR*, 2019, pp. 4974–4983.
- [48] N. Luo, Y. Pan, R. Sun, T. Zhang, Z. Xiong, and F. Wu, "Camouflaged instance segmentation via explicit de-camouflaging," in *CVPR*, 2023, pp. 17918–17927.
- [49] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *ICCV*, 2019.
- [50] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," in *ECCV*, 2020.
- [51] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic, faster and stronger," in *NeurIPS*, 2020.
- [52] P. Skurowski, H. Abdulameer, J. Baszczyk, T. Depta, A. Kornacki, and P. Kozie, "Animal camouflage analysis: Chameleon database," 2018.
- [53] E. L.-M. Pia Bideau, "It's moving! a probabilistic model for causal motion segmentation in moving camera videos," in *ECCV*, 2016.
- [54] H. Lamdouar, C. Yang, W. Xie, and A. Zisserman, "Betrayed by motion: Camouflaged object discovery via motion segmentation," in *ACCV*, Nov 2020.
- [55] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, "Simultaneously localize, segment and rank the camouflaged objects," in *CVPR*, Jun 2021.
- [56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [57] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE TPAMI*, vol. 26, no. 5, pp. 530–549, 2004.
- [58] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*. Springer, 2020, pp. 213–229.
- [59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, Jun 2016, pp. 770–778.
- [60] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.
- [61] Z. Liu, Y. Lin, Y. Cao, H. Han, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Hierarchical vit using shifted windows," in *CVPR*, 2021.
- [62] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.
- [63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. IEEE, 2009.
- [64] L. Ke, M. Danelljan, X. Li, Y.-W. Tai, C.-K. Tang, and F. Yu, "Mask transfiner for high-quality instance segmentation," in *CVPR*, 2022.
- [65] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *ECCV*. Springer, 2020, pp. 282–298.
- [66] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, and W. Liu, "Instances as queries," in *ICCV*, 2021, pp. 6910–6919.
- [67] R. Guo, D. Niu, L. Qu, and Z. Li, "Sotr: Segmenting objects with transformers," in *ICCV*, 2021, pp. 7157–7166.



# Few-shot Camouflaged Animal Detection and Segmentation

Thanh-Danh Nguyen<sup>1,4†</sup>, Anh-Khoa Nguyen Vu<sup>1,4†</sup>, Nhat-Duy Nguyen<sup>1,4†</sup>, Vinh-Tiep Nguyen<sup>1,4</sup>, Thanh Duc Ngo<sup>1,4</sup>, Thanh-Toan Do<sup>5</sup>, Minh-Triet Tran<sup>2,3,4</sup> and Tam V. Nguyen<sup>6\*</sup>

<sup>1</sup>University of Information Technology, Ho Chi Minh City, Vietnam. <sup>2</sup>University of Science, Ho Chi Minh City, Vietnam. <sup>3</sup>John von Neumann Institute, VNU-HCM, Vietnam. <sup>4</sup>Vietnam National University, Ho Chi Minh City, Vietnam. <sup>5</sup>Monash University, Clayton, VIC 3800, Australia. <sup>6</sup>University of Dayton, Dayton, OH 45469, United States. (<sup>†</sup>equal contribution, <sup>\*</sup>corresponding author)

CV4Animals Workshop



## Introduction

**Overview:** Camouflage is a defense mechanism that animals use to conceal their appearance by blending in with their environment. Automatically detecting camouflaged animals is helpful in various fields of computer vision: search-and-rescue mission; wild species discovery and preservation activities; media forensics, etc.

### Motivation:

- Research on camouflaged animals suffers from the lack of data
- Camouflaged instances have their texture similar to the background

### Main contributions:

1. A novel benchmark **CAMO-FS** for few-shot detection and segmentation on camouflaged animals
2. A **framework** to efficiently detect and segment camouflaged instances given a small number of training data for novel classes, thanks to **Instance Triplet Loss** and **Instance Memory Storage**

## CAMO-FS Benchmark Dataset

- Our **CAMO-FS** benchmark contains:
  - **2,852** images/ **3,342** camouflaged instances with fine-grained annotations for detection and instance segmentation
  - **enhanced 163** images compared with CAMO++ dataset [1])
  - **47** semantic classes, customized for few-shot learning with 1, 2, 3, and 5-shot

This dataset is among the firsts to address few-shot camouflaged detection and instance segmentation

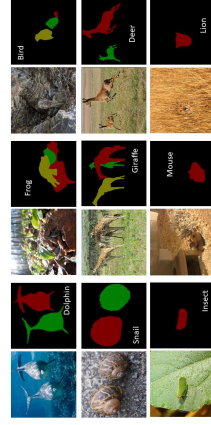


Fig. 1 Exemplary camouflaged instances in CAMO-FS

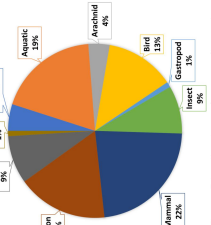


Fig. 2 Distribution of camouflaged course-grained classes in CAMO-FS

## General Framework

Originated from MTTFA [2] model which is a two-stage training and fine-tuning mechanism, we introduce a framework for few-shot camouflaged detection and instance segmentation as below:

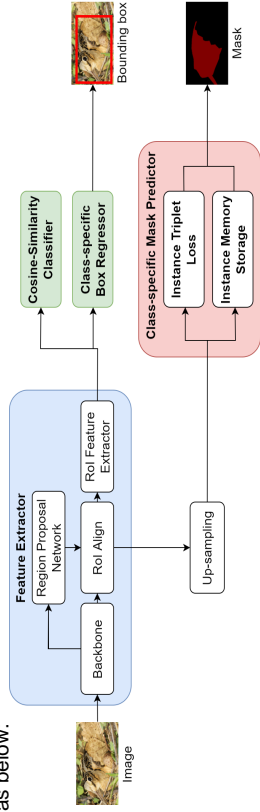


Fig. 3 General framework for camouflaged animal detection and instance segmentation

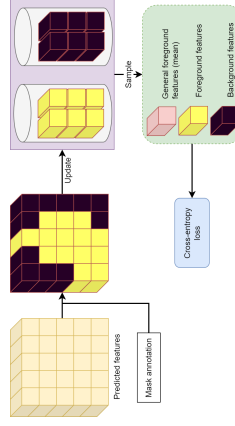
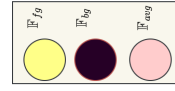


Fig. 4 Instance Memory Storage

### Instance Memory Storage

Contains the **background** and **foreground features** per each class, and calculates the discrimination between areas of objects and no-objects in region proposals



**Instance Triplet Loss** minimizes/ maximizes the distances among foreground/ background representatives

Fig. 5 Instance Triplet Loss

## Acknowledgement

This research was supported by the VNUHCM-University of Information Technology's Scientific Research Support Fund

## Experimental Results

Tab. 1 Our Improvement of Instance Triplet Loss and Memory Storage on MTTFA model

#	Method	AP	AP50	AP75	AP <sub>FB</sub>	AP <sub>FM</sub>	AP <sub>FA</sub>	ARI	ARI0	ARI5	ARI <sub>0</sub>	ARI <sub>5</sub>	ARI <sub>0</sub>	ARI <sub>5</sub>
<b>Instance Segmentation</b>														
1	Baseline	3.66	5.37	4.06	21.33	4.13	4.01	12.36	13.68	35.97	12.96	12.93	16.07	16.07
	+ Memory	4.46	6.21	4.69	21.33	4.13	4.01	12.36	15.04	35.97	23.17	14.49	20.89	20.89
2	Baseline	5.75	8.58	6.91	27.79	6.29	6.01	17.08	19.09	29.41	11.45	11.45	15.09	15.09
	+ Memory	5.77	9.05	6.91	27.83	6.01	5.87	15.67	17.33	30.15	13.37	17.00	17.00	17.00
3	Baseline	6.95	10.72	7.60	32.62	5.73	6.44	20.00	22.15	31.25	13.86	20.92	20.92	20.92
	+ Memory	6.16	8.95	6.68	33.74	6.19	5.08	20.25	22.95	36.83	16.31	21.63	21.63	21.63
5	Baseline	7.36	11.23	8.49	37.03	6.24	5.64	24.40	27.69	38.44	17.02	26.71	26.71	26.71
	+ Memory	5.95	8.67	6.94	34.71	6.25	4.85	21.29	24.42	36.86	14.54	21.83	21.83	21.83
<b>Detection</b>														
1	Baseline	2.93	5.80	2.20	20.36	4.18	2.03	9.25	10.84	21.74	11.49	8.77	11.49	8.77
	+ Memory	3.44	6.44	2.64	20.36	4.18	2.03	9.25	10.84	21.74	11.49	8.77	11.49	8.77
2	Baseline	4.50	9.14	3.45	22.89	5.01	3.74	13.14	15.22	23.14	8.75	16.33	16.33	16.33
	+ Memory	5.90	8.87	6.83	33.04	9.74	3.10	17.26	19.25	34.04	15.74	19.61	19.61	19.61
3	Baseline	7.28	11.22	8.25	32.31	10.72	6.83	20.52	22.69	31.01	18.88	23.52	23.52	23.52
	+ Memory	5.83	8.98	6.29	34.56	7.78	4.31	19.13	21.83	35.80	15.93	21.09	21.09	21.09
5	Baseline	7.09	11.51	8.23	38.45	8.61	6.38	24.88	27.52	38.55	17.66	27.44	27.44	27.44
	+ Memory	5.81	9.13	6.00	35.47	8.17	4.29	19.07	22.96	35.94	14.16	22.58	22.58	22.58
5	Baseline	9.76	14.37	11.12	40.05	8.82	8.89	25.93	29.28	40.05	12.53	30.32	30.32	30.32
	+ Memory	10.36	16.27	11.79	39.32	8.08	11.86	26.34	30.39	39.35	12.37	30.91	30.91	30.91

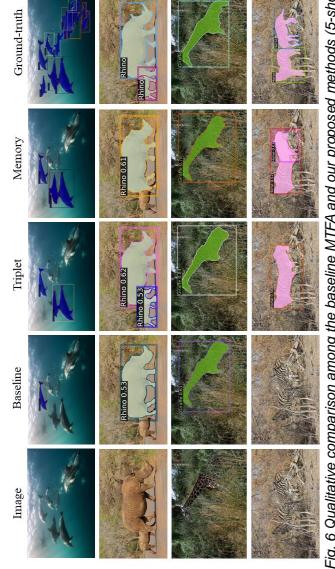


Fig. 6 Qualitative comparison among the baseline MTTFA and our proposed methods (5-shot)

## References

- [1] Le, T.-N. et al.: Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite. IEEE Transactions on Image Processing 31, 287-300 (2022)
- [2] Ganeva, D.A. et al.: Incremental few-shot instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 1185-1194 (2021)

# The Art of Camouflage: Few-shot Learning for Animal Detection and Segmentation

Thanh-Danh Nguyen<sup>1,4†</sup>, Anh-Khoa Nguyen Vu<sup>1,4†</sup>,  
Nhat-Duy Nguyen<sup>1,4†</sup>, Vinh-Tiep Nguyen<sup>1,4</sup>, Thanh Duc Ngo<sup>1,4</sup>,  
Thanh-Toan Do<sup>5</sup>, Minh-Triet Tran<sup>2,3,4</sup>, Tam V. Nguyen<sup>6\*</sup>

<sup>1</sup>University of Information Technology, Ho Chi Minh City, Vietnam.

<sup>2</sup>University of Science, Ho Chi Minh City, Vietnam.

<sup>3</sup>John von Neumann Institute, VNU-HCM, Vietnam.

<sup>4</sup>Vietnam National University, Ho Chi Minh City, Vietnam.

<sup>5</sup>Monash University, Clayton, VIC 3800, Australia.

<sup>6\*</sup>University of Dayton, Dayton, OH 45469, United States.

\*Corresponding author(s). E-mail(s): [tamnguyen@udayton.edu](mailto:tamnguyen@udayton.edu);

Contributing authors: [danhnt@uit.edu.vn](mailto:danhnt@uit.edu.vn); [khoanva@uit.edu.vn](mailto:khoanva@uit.edu.vn);

[duymn@uit.edu.vn](mailto:duymn@uit.edu.vn); [tiepnt@uit.edu.vn](mailto:tiepnt@uit.edu.vn); [thanhd@uit.edu.vn](mailto:thanhd@uit.edu.vn);

[toan.do@monash.edu](mailto:toan.do@monash.edu); [tmtriet@fit.hcmus.edu.vn](mailto:tmtriet@fit.hcmus.edu.vn);

†These authors contributed equally to this work.

## Abstract

Camouflaged object detection and segmentation is a new and challenging research topic in computer vision. There is a serious issue of lacking data of camouflaged objects such as camouflaged animals in natural scenes. In this paper, we address the problem of few-shot learning for camouflaged object detection and segmentation. To this end, we first collect a new dataset, CAMO-FS, for the benchmark. We then propose a novel method to efficiently detect and segment the camouflaged objects in the images. In particular, we introduce the instance triplet loss and the instance memory storage. The extensive experiments demonstrated that our proposed method achieves state-of-the-art performance on the newly collected dataset.

**Keywords:** Camouflaged Instance, Camouflaged Animal, Few-shot Learning, Object Detection, Instance Segmentation

# 1 Introduction

Camouflage is a defense mechanism that animals use to conceal their appearance by blending in with their environment [1]. Autonomously detecting camouflaged animals is helpful in various applications, e.g., search-and-rescue missions [2]; wild species discovery and preservation activities [2]; and media forensics (manipulated image/video detection and segmentation [3]). Although image segmentation methods have been proposed for a long time, general detectors cannot deal with camouflaged animals [4–7]. The detectors initially developed for camouflage detection [8–15], which use handcrafted low-level features, are effective only for images with a simple and uniform background. More recently developed deep learning-based detectors [2, 16] for camouflaged object segmentation rely on large-scale data.

With the approach of few-shot learning, we can perform machine learning tasks with given limited data. A few-shot method requires two stages of processing: (1) one base phase training for the model to gain concept knowledge of general domains with abundant data, and then (2) performing a novel phase which can do the specific task on few-shot data. In the case of the camouflaged object detection and segmentation task, we leverage few-shot learning in the concept of camouflaged animals which is rare and hard to find in the wild. Thus, with limited data on camouflaged objects, the models can still well handle the given tasks. However, none of the aforementioned publications targets few-shot camouflaged object detection and segmentation despite its practical applications. The task of segmentation supports better to identify camouflaged objects in terms of specifying which pixels in the images conceal the objects in comparison to classification and detection. In fact, the research on camouflaged animals suffers due to the lack of data. There are not many object classes with rare instances captured in photos. Therefore, in this paper, we would like to address few-shot learning with camouflaged animals.

Our contributions in this work are two-fold:

- First, we build a new benchmark dataset, CAMO-FS, which is among the first datasets to support few-shot detection and instance segmentation on camouflaged instances in nature.
- Second, we propose FS-CDIS, a framework to efficiently detect and segment camouflaged instances given a small shot of training data for novel classes utilizing an instance triplet loss and memory storage.

The remainder of this paper is organized as follows. [Section 2](#) summarizes related work. Next, [Section 3](#) introduces the newly constructed CAMO-FS dataset and presents our proposed framework for few-shot camouflaged object detection and segmentation. [Section 4](#) presents the results of our evaluation of baselines on the newly constructed dataset. Finally, [Section 5](#) summarizes the key points and mentions future work.

## 2 Related Work

### 2.1 Camouflage Research

Given any region (i.e. bounding boxes or polygon masks) presented for an object of interest (i.e. animals or artificial objects) in an image and then they tend to be classified as background, contents in that region can be qualified as camouflaged objects. Thus, a camouflaged object is defined as a set of bounding boxes or camouflaged pixels in an image without any further detailed information such as the number of objects or the semantic meaning [2]. Although tasks related to camouflaged animals are performed in a wide range of applications, this research field has not been well explored in the literature, especially few-shot learning which is practically suitable to the context of scarce data as camouflaged animals.

**Binary camouflage segmentation.** Prior to the advancement of deep neural networks, most of the work exploits identical regions between camouflaged regions and the background by handcrafted or low-level features, specifically based on external characteristics (e.g., color, shape, orientation, and brightness). Particularly, early camouflage detection works had attention on the foreground region even when some of its texture was similar to the background [8–10, 17]. The foreground was distinguishable from the background via simple features, such as color, intensity, shape, orientation, and edge [17–21]. A few methods [11–15, 22] based on handcrafted low-level features have been proposed for tackling the problem of camouflage detection. However, they are effective only for images with a simple and uniform background. Thus, their performances are unsatisfactory in camouflaged object segmentation due to the substantial similarity between the foreground and the background.

Until now, the convention of binary prefers binary ground truth camouflaged object datasets [2, 16, 23]. Existing methods for camouflaged objects [2, 16, 24–29] based on binary ground truth are considered as the binary camouflage segmentation. For example, Le *et al.* [2] proposed an end-to-end Anabranched Network, dubbed ANet which includes two streams of classification and segmentation. The outputs of both streams are fused to improve the segmentation performance of camouflaged objects. This proposed network was also flexibly applied to any fully convolutional networks. Similarly, motivated by the way of hunting strategies of predators, Fan *et al.* [16] designed Search Identification Network (SINet) with two main modules to simulate this hunting behavior, namely a search module searching for targets and an identification module identifying the existence of targets then catching them. Yan *et al.* [27] recently introduced MirrorNet, a dual-stream network comprising a mainstream and a mirror stream. This mirror stream aimed to capture instinct information by horizontally flipping camouflaged objects to break their camouflaged nature and make them more distinguishable. Zhu *et al.* [28] presented the TINet, which interactively refines multi-level texture and segmentation features and thereby gradually enhances the segmentation of camouflaged objects. Lv *et al.* [29] simultaneously worked on ranking and localization to well-present camouflaged objects. As a result, they formed a triplet task with localizing, segmenting, and ranking the camouflaged objects. Besides, the authors also introduced the NC4K dataset for camouflaged segmentation. Such methods reveal the presence of the camouflaged objects with the high level of bounding

boxes and contain corresponding pixel-wise ground truth belonging to camouflage. Further understanding of the camouflage level may help us to give comparative analyses, finding evidence for links between camouflage and other defensive strategies with aspects of habitat and life-history [30].

**Camouflage instance segmentation.** Although several works have been proposed, there is still a difficulty in efficiently exploring the information of camouflage animals, especially at the instance level with more challenging detailed masks. Therefore, for ease of training methods with the challenging task of camouflaged instance segmentation, Le *et al.* [31] introduced a framework with several state-of-the-art methods and proposed a tool with user interactive cues to tune the segmentation mask on a website. Realizing that the semantic level is not detailed enough, Le *et al.* [32] introduced a camouflage fusion learning (CFL) to utilize the strength of different instance segmentation methods by fusing various models via learning image contexts.

**Camouflage datasets.** CamouflagedAnimals [33] and CHAMELEON [23] were the first two camouflage datasets with mask annotations. The two datasets do not contain enough images to train deep learning methods. Le *et al.* [2] created the CAMO dataset, the first camouflage dataset with more than 1,000 annotated images. It contains 1,250 annotated images, which is a limited number of samples to train and evaluate deep learning methods. Then, Fan *et al.* [16] collected the COD dataset, which comprises 10,000 images (both camouflage and non-camouflage) divided into 5 meta-categories. However, they annotated only 5,066 camouflage images. Lamdouar *et al.* [34] recently developed the MoCA dataset for the camouflage object detection task; it contains only bounding box ground truths. Hence, these datasets limit their annotations at binary ground truth datasets which have a shortage of intensive annotations for multi-task camouflage problems. CAMO++ [32] is different from the above-mentioned dataset, CAMO++ provides a benchmark for camouflaged instance segmentation with more comprehensive annotations and diverse meta-categories of 10. The dataset comprises 5,500 images with superiority over other datasets on instances including 32,756 instances for both camo and non-camo objects.

## 2.2 Few-shot Learning

**Few-shot object detection (FSOD).** When having some available samples of given classes with their corresponding bounding boxes, FSOD aims to learn from these limited data in order to help models adapt to the new classes. To date, several works [35–38] have been proposed to deal with FSOD. Early works [36, 38] mainly prefer to overcome the difficulties of the data scarcity of FSOD via meta-learning approaches by combining supportive information from meta-based streams with their main streams. Particularly, Bingyi [36] proposed a Feature Reweighting framework that leverages the free-proposal approach of a well-known one-stage framework such as YOLO [39] to boost FSOD performance. The network integrated a meta-model that aims to generate reweighting vectors from support samples for highlighting the attention to features from the YOLO network. Conversely, Meta RCNN [38] based on the two-stage proposal approach as Mask RCNN [40] and fed available annotations such as bounding boxes and segmented masks to train a meta-network called Predictor-head Remodeling Network for inferring attention features. Fan *et al.* [35] recently proposed to take

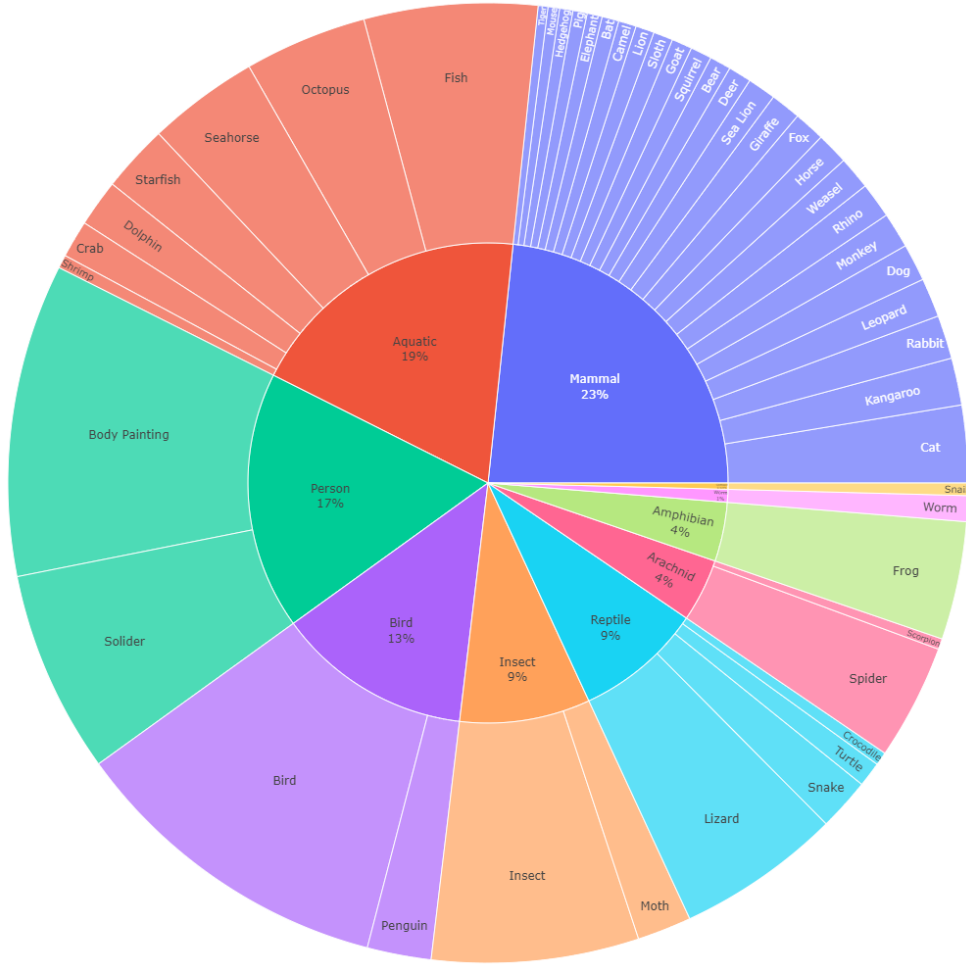


**Table 1:** Statistics of camouflage datasets (without non-camo images).

Dataset	Year	Publication	Type	#Annot. Camo. Img.	#Meta- Cat.	#Obj. Cat.	#Ins. or #Obj. per Img.	Bbox. GT	Obj. Mask GT	Ins. Mask GT	Few-shot
CamouflagedAnimals [33]	2016	ECCV	Video	181	-	6	1.238	×	✓	✓	×
MoCA [34]	2020	ACCV	Video	7,617	-	67	1.000	✓	×	×	×
CHAMELEON [23]	2018	-	Image	76	-	-	1.000	×	✓	×	×
CAMO [2]	2019	CVIU	Image	1,250	2	8	1.000	×	✓	×	×
COD [16]	2020	CVPR	Image	5,066	5	69	1.171	✓	✓	✓	×
CAMO++ [32]	2022	TIP	Image	2,695	10	47	1.171	✓	✓	✓	×
CAMO-FS (Ours)	2023	-	Image	2,858	10	47	1.172	✓	✓	✓	✓

advantage of support images from a massive FSOD dataset to generate significant results combined with their proposed network called Attention-RPN, Multi-Relation Detectors. The Attention-RPN directed the trained model to look at the image for the task of object detection. Differently, Wang *et al.* [37] simply adopted Faster RCNN with two-stage finetuning to transfer massive knowledge from abundant data in the base model to fine-tune the novel one by freezing the whole network except for the fully connected layer for object classification. Through this simple straightforward mechanism, this model significantly improved few-shot performance without a complex pipeline of training the model. Further, such works [41–45] presented advanced methods by applying class max-margin, multiple scale proposals, or feature alignment in FSOD. Other ones were based on transformed inputs [46, 47], transformer approaches [48, 49], contrastive method [50], or kernels design [51]. Other methods [37, 52–54] relied only on query images to deal with FSOD via extra text data [54], unlabeled image [55], generated samples [53], gradient scaling [52].

**Few-shot object segmentation (FSOS).** Recently, the field of few-shot segmentation gained attention from the community. As mentioned above, the first work Meta RCNN originated from Mask RCNN, therefore, Meta RCNN simultaneously performed detection and segmentation. Liu *et al.* [56] utilized a cross-reference network for generic image segmentation. The authors proposed a cross-reference mechanism and a mask refinement module to specifically support the task of segmentation. Before, Dong *et al.* [57] proposed a prototype learning component in a framework of semantic segmentation that learned to take discriminative information from features to help segment objects better. Also, Wang *et al.* [58] introduced a prototype align method that learns class-specific prototype representations from a few image samples to perform segmentation over the query images. Lately, Liu *et al.* proposed a dynamic prototype convolution network to address few-shot semantic segmentation. The work of [59] proposed context-aware prototype learning. [60] introduced generative models approach for this task. Recently, Nguyen *et al.* [61] came up with iFS-RCNN, an instance segmenter via an incremental approach. Gao *et al.* [62] proposed the DCFS framework, an effective decoupling classifier that boosted the performance of object detection and segmentation heads. Han *et al.* [63] suggested a reference twice transformer-based framework (RefT) to enhance features in segmentation tasks. Also in the transformer approach, Wang *et al.* [64] introduced DTN to directly segment the target object instances from arbitrary categories given reference images. In common, these aforementioned methods focus on generic objects.



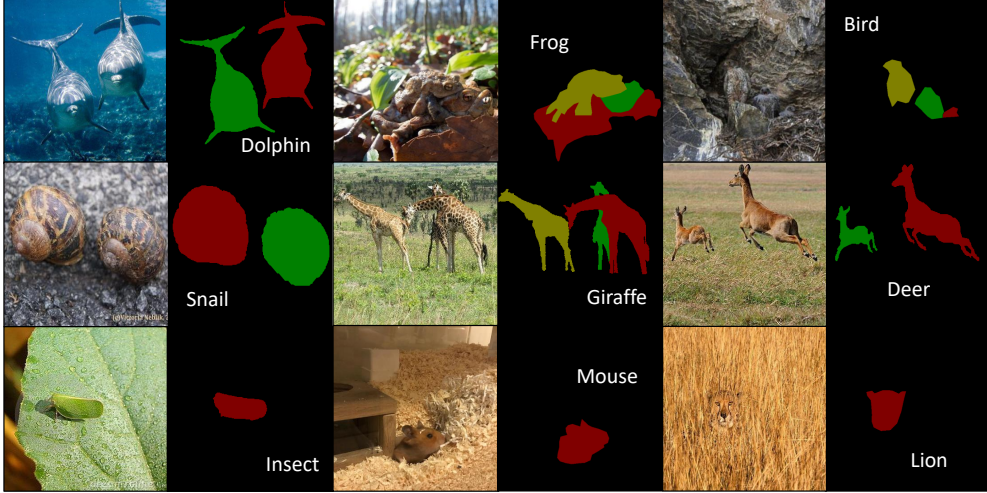
**Fig. 1:** Hierarchical taxonomic structure of our CAMO-FS dataset and the corresponding distribution of camouflaged coarse-grained classes. Best viewed online in color and zoomed in.

## 3 Proposed Method

### 3.1 CAMO-FS Benchmark Dataset

Camouflaged data tends to be more difficult to collect in the real world rather than non-camouflaged ones. Generating intensive annotations with multi-task or hierarchical labels for camouflaged objects is also costly and complicated, especially with the pixel level as polygon masks. Particularly, the visual characteristics of a camouflaged object are extremely identical to the background. The external appearances (i.g. the intensity, color, and textures) are close to their surrounding environment, the boundary



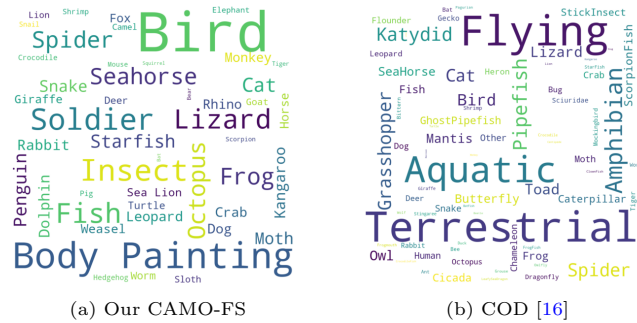


**Fig. 2:** Exemplary images with instance-level mask annotations from our proposed CAMO-FS dataset.

between camouflaged objects and the background or other identical-type camouflaged objects in case of being nearly or partly overlapped. Thus, it is really tough to provide the concurrence between annotators due to ambiguity in verifying camouflaged regions blended in surroundings. For ease of data preparation such as collections and annotations, one of the most common way is that inherits existing camouflaged datasets and CAMO++ [32] is our selected dataset since it is a high-diversity dataset with a variety of camouflaged object categories. Furthermore, the key to few-shot learning lies in the generalization ability of the pertinent model when presented with a few available samples. The context of camouflaged objects inherently matches this understanding because the number of camouflaged images is often scarce in practice.

**CAMO++ Dataset.** CAMO++ generally contains camouflaged and non-camouflaged images with a total of 5,500 images corresponding to 32,756 instances [32]. The dataset contains 93 fine-grained classes assigned to 13 coarse-grained classes. However, in the case of camouflaged objects, there are 47 fine-grained classes designed with a hierarchical structure and assigned into 10 coarse-grained classes. In detail, CAMO++ contributes 2,695 camouflage images including 1,250 existing camouflage images in the previous CAMO dataset with 1,450 newly collected camouflage images for CAMO++. In this scope of our paper, 2,800 remaining non-camouflage images are ignored. CAMO++ especially provides common ground truths such as bounding boxes, object masks, and instance masks which are suitable for many tasks of camouflage research.

**CAMO-FS Dataset.** We leverage the available CAMO++ to build our CAMO-FS dataset. In this way, we inherit the biology taxonomic and vision taxonomic structure of CAMO++ which helps us to reduce the burden of data collection. Table 1 provides an overview of previous works done on camouflage, which is mentioned in the related work, and our proposed CAMO-FS in terms of main characteristics. We exploit the



**Fig. 3:** Word cloud of class-distribution of camouflaged instances.

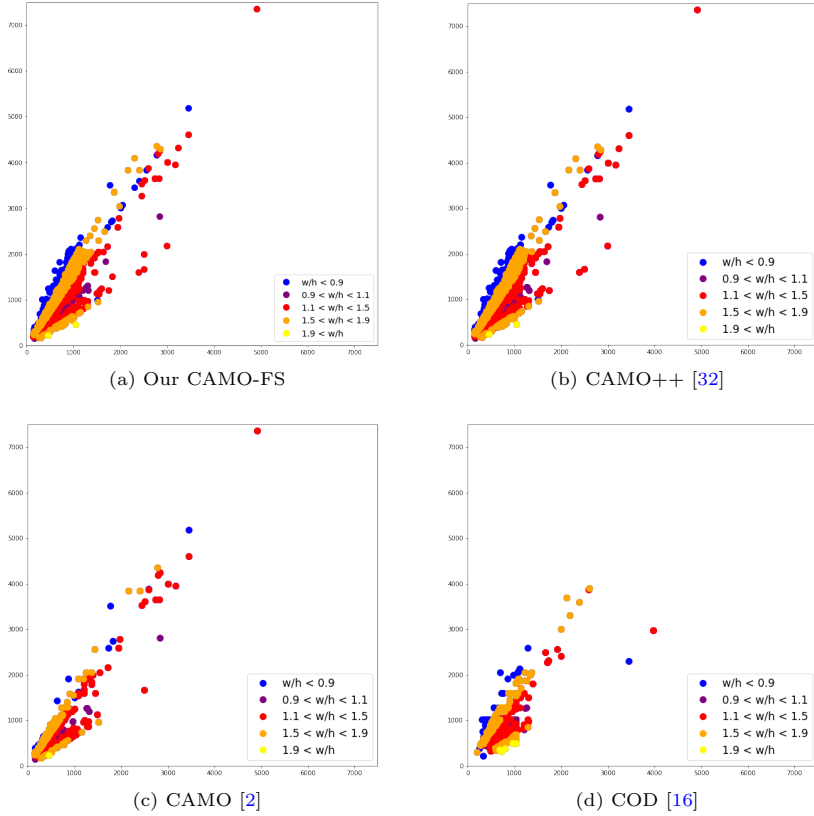
**Table 2:** Extra collected number of images and instances in CAMO-FS dataset.

Classes	Bat	Bear	Camel	Dolphin	Elephant	Horse	Kangaroo	Monkey	Penguin	Rhino	Squirrel	Total
#images	12	14	14	13	14	16	22	16	11	14	17	163
#instances	12	14	15	19	14	17	25	20	14	14	17	181

diversity of CAMO++ by its 10 meta-categories to build up the few-shot concept for instance segmentation. To this end, our CAMO-FS not only keeps a good ratio of instances per image of 1.172 but also contributes as the very first dataset specific for few-shot research on camouflaged animals. Note that the large amount of images in some datasets does not mean they are all camouflaged images. **Figure 3** illustrates the class-distribution of our CAMO-FS dataset and COD [16]. Our CAMO-FS shares similar categorical diversity with the original CAMO++ [32].

However, imbalanced data and a shortage of the number of images of some classes inherently exist in CAMO++ posing problems of evaluation for few-shot tasks. Particularly, there are 11 classes (*e.g. Camel, Dolphin, Elephant, Horse, Kangaroo, Monkey, Penguin, Bat, Bear, Squirrel and Rhino*) even having a shortage of images that are needed to train a few-shot model. Hence, we hardly perform training or testing on these classes. As a result, we collect more data for these classes with 163 total images corresponding to 181 instances (an average of 15-16 instances per class). We also remove images with mistakes in the original dataset. The statistics of collected data are shown in **Table 2**. By gathering more camouflaged animals and combining them with the CAMO++ dataset, we conduct our CAMO-FS dataset for few-shot camouflaged animal detection and segmentation with 2,858 total images corresponding to 3,342 instances. **Figure 1** shows the vision taxonomic structure of coarse-grained and corresponding fine-grained classes and illustrates the ratios of 10 coarse-grained classes in our proposed CAMO-FS dataset. **Figure 2** visualizes the exemplary images with mask annotations from our proposed CAMO-FS.

In **Table 3**, we report the aggregated number of instances per image. The number of instances per image ranges from 1 to 25 and commonly falls into 1, then 2 and 3 while the remaining is beyond 3 instances. As can be seen, the number of images that

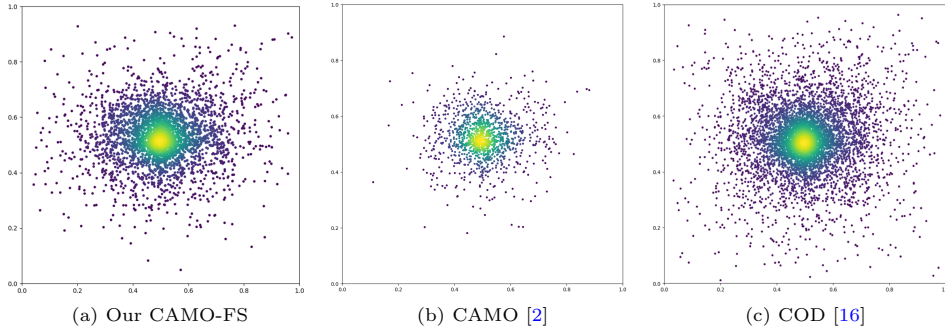


**Fig. 4:** Distribution of camouflage image resolution. Best viewed online in color and zoomed in.

No. of instances	Ratio (%)	#Images
1	90.5	2581
2	1.05	190
3	1.79	51
3+	6.66	30

**Table 3:** CAMO-FS dataset instances per image distribution.

contain 1 to 3 instances takes up a large proportion of the entire dataset. This also illustrates the problem of data imbalance between the number of instances and the ratio of images in the dataset, which reflects a problem that the presence of camouflaged animals captured in photos is often limited, i.e. mostly one animal per image. Additionally, although being claimed in [32] that camouflaged objects in CAMO++ were localized over the entire image, after removing non-camouflage objects and adding new camouflaged images, we have the distributions of object centers in normalized image coordinates over all images in the CAMO-FS dataset as in Figure 5-a. This means



**Fig. 5:** Instance center bias camouflaged datasets. Best viewed online with color and zoomed in.

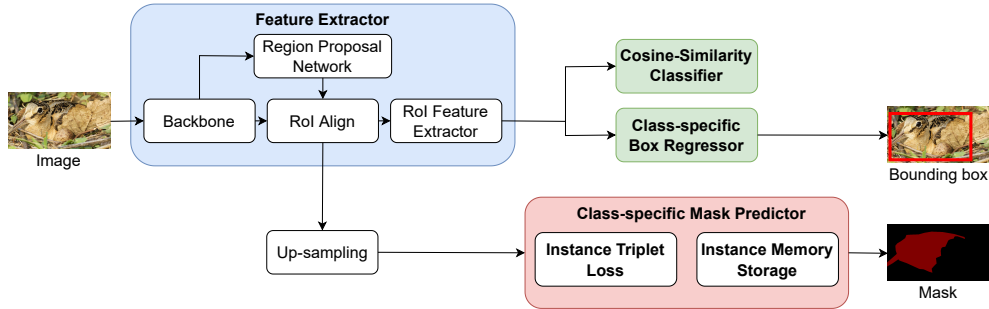
camouflaged animals tend to be located in the center of images. Indeed, to capture images of camouflaged animals in the wild, photographers need to carefully focus on the animals, which leads to the central layout of collected images. Also in [Figure 5](#), we illustrate the center bias of camouflaged images in other CAMO [\[2\]](#) and COD [\[16\]](#) datasets for better visual comparison. In [Figure 4](#), we present the image resolution among camouflage datasets. As we only consider camouflaged images of CAMO++ [\[32\]](#) and COD [\[16\]](#), the density of our CAMO-FS is slightly higher than CAMO++ as a result of our extra collection of images presented in [Table 2](#). In comparison with the previous COD [\[16\]](#) and CAMO [\[2\]](#), our CAMO-FS image resolution distribution is more satisfying in diversity.

To effectively create the data for the few-shot problem, we get  $M$  instances from the CAMO-FS dataset to create training sets (in our setup,  $M = 5$ ) and use the remaining instances for testing. We only remove some objects of the higher-level training set if it exists to create the other few-shot settings. For example, we get all elements to generate 5-shot training data and discard 2 in 5 objects to make a 3-shot one. In this way, the 5-shot benchmark contains objects of the 3-shot dataset and the 3-shot setting contains the objects of the 2-shot one.

To the best of our knowledge, this is among the first works to address few-shot camouflaged instance segmentation and detection. Given the lack of a large-scale dataset for training and testing purposes on camouflaged animal issues, we build a benchmark for the task of few-shot camouflaged instance segmentation and detection.

### 3.2 General Framework

**Few-shot instance segmentation formulation.** In few-shot learning, we have one set of base classes denoted  $C_{base}$  with a large amount of available training data, and one disjoint set of novel classes denoted  $C_{novel}$  containing a small amount of training data. This amount is small to a few samples. The ultimate goal is to train a model to predict well on the novel classes  $C_{test} = C_{novel}$  [\[65, 66\]](#) or on both base and novel data  $C_{test} = C_{base} \cup C_{novel}$  [\[67\]](#). In few-shot classification, this work [\[66\]](#) introduces the method of episodic training. The method sets up a series of episodes  $E_i = (I_q, S_i)$



**Fig. 6:** Our general FS-CDIS framework for Few-Shot Camouflaged Detection and Instance Segmentation.

where  $S_i$  is a support set that contains  $N$  classes from  $C_{train} = C_{novel} \cup C_{base}$  along with  $K$  examples per class (so-called  $N$ -way  $K$ -shot). A network is then trained to classify an input image  $I_q$ , termed query image, out of the classes in  $S_i$ . The key idea is that solving a different classification task for each episode leads to better generalization and results on  $C_{novel}$ . The extended versions of this method are FSOD [36] and FSIS [38, 68]. Those proposals consider all objects in an image as queries and they have a single support set per image instead of per query. However, there exist challenges in FSIS which are not only classification tasks on the query objects but also how to determine their localization and segmentation. Use an image  $I_q$  to query, FSIS returns labels  $y_i$ , bounding boxes  $b_i$ , and segmentation masks  $M_i$  for all objects in  $I_q$  that belong to the set of  $C_{test}$ .

**General framework.** Originated from TFA [37] which uses Faster R-CNN [69], MTFA [70] employs a mask prediction branch to return the pixel-wise mask for the segmentation task. In this work, we leverage the architecture of MTFA model [70] based on Mask R-CNN [40] which is a two-stage training and fine-tuning mechanism. We train the first stage of the framework on 80 classes from the COCO dataset. This stage results in the base model weights for the second stage of novel fine-tuning. In the fine-tuning stage, we apply the few-shot technique to learn the novel concepts of camouflaged instances in our proposed CAMO-FS dataset.

Similar to Mask R-CNN, the input images are fed into a feature extractor  $F$  consisting of backbone  $B$ , RoI Align, RoI feature extractor modules, and a region proposal network. There are three heads specifying three tasks that this scheme supports: a classification head  $C$ , a box regression head  $R$ , and a new attached mask prediction head  $M$ . In the first stage, the network is trained on the base classes  $C_{base}$ . Then in the second stage, we froze the backbone network  $B$  of the feature extractor  $F$  and only perform training on the prediction heads. Thus, only RoI classifier  $C$ , box regressor  $R$ , and mask predictor  $M$  are fine-tuned in the second stage. In Figure 6, there exists a branch called mask predictor  $M$ . We apply similarly to Ganea *et al.* [70] by using this two-stage fine-tuning approach. Firstly, the network is trained on base classes with lots of abundant data and then fine-tuning all predictor head  $C$ ,  $R$ , and  $M$  on novel data of  $K$  shots for each class.

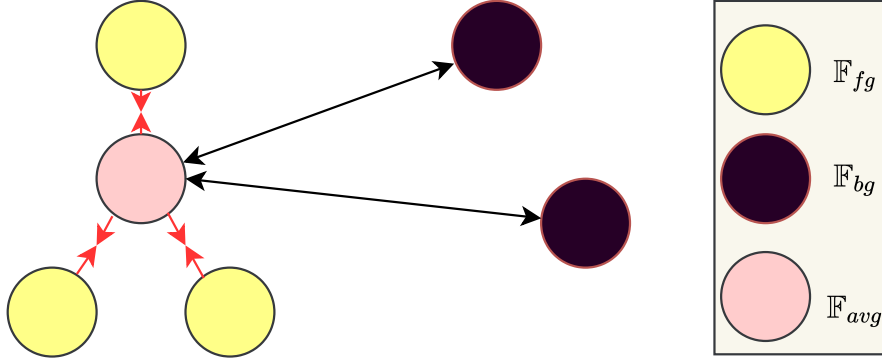


Fig. 7: Visualization of triplet loss for region proposal.

Not a simple mask predictor  $M$  that we use, we enhance the performance of the instance segmentation task by employing the two concepts of instance triplet loss and instance memory storage which are clearly described in the next section. The two improvements are inspired not only by the instance segmentation task in general but also by the camouflaged instance segmentation specifications.

### 3.3 Framework Improvement

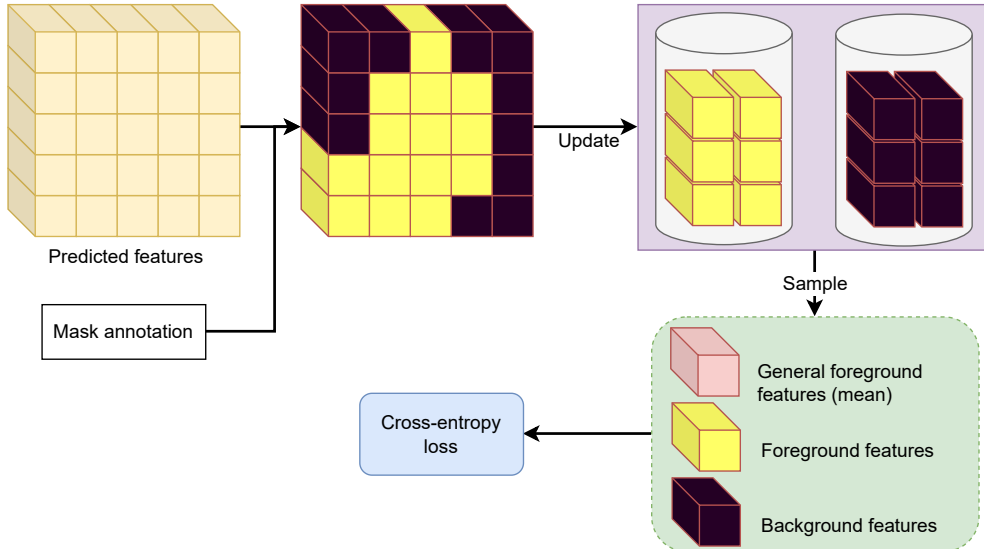
One of the characteristics of camouflage instances is the camouflage texture similar to the background. This makes the precise identification of the boundary areas difficult. It is more critical in the context of few-shot learning where the concepts of a class are represented by only a few samples.

In this work, we thus propose improvements to enhance distinguishable features between background and foreground areas. In particular, we explore two approaches that focus on loss functions. The first one is the triplet loss function which was known as a strong metric to support the network in creating discrimination features between anchor and negative. The second approach is the idea of memory bank, which is used to enhance the distance between foreground and background not only for individual instances but also for each novel class. To this end, our framework is named after FS-CDIS.

To calculate the loss function, we employ the mask annotation for RoI features to collect the  $\mathbb{F}_{bg}$  background and  $\mathbb{F}_{fg}$  foreground features by location on each RoI. Both  $\mathbb{F}_{bg}$  and  $\mathbb{F}_{fg}$  for each proposal are used to calculate the respective loss functions which are presented in the following sections.

#### 3.3.1 Instance triplet loss

With the idea of enhancing the discrimination between camouflaged instances and their backgrounds, we leverage the power of the triplet loss function [71]. Specifically, we treat the pixels of an object as positive points and the background as negative ones. Accordingly, we force the model to learn the distinguished features among the foreground and background representatives. The more distinguished among features,



**Fig. 8:** Visualization of memory loss for region proposal.

the better a model can do to detect or segment camouflaged instances. In this way, we highlight the camouflaged instances so that the model is able to recognize them.

For each RoI, we consider the average foreground features  $\mathbb{F}_{avg} = \frac{1}{|\mathbb{F}_{fg}|} \sum \mathbb{F}_{fg}$  as anchors with the foreground feature  $\mathbb{F}_{fg}$  as positive and the background feature  $\mathbb{F}_{bg}$  as negative to apply the triplet loss function [71]. In this way, the model tries to learn to minimize the distance between foreground representatives and maximize the distance between background representatives as shown in Figure 7. We use cosine similarity to calculate the distance instead of Euclidean distance. The loss function is defined as:

$$\begin{aligned} \mathcal{L}_{triplet} &= \max\{d(\mathbb{F}_{avg}, \mathbb{F}_{fg}) - d(\mathbb{F}_{avg}, \mathbb{F}_{bg}) + margin, 0\} \\ d(x, y) &= 1 - \frac{x \cdot y}{\|x\| \cdot \|y\|} \end{aligned} \quad (1)$$

, where *margin* controls the discrimination between foreground and background features. In our experiments, we set *margin* of 0.5.

### 3.3.2 Instance memory storage

The memory bank is designed to store information within a class and the class information is updated during the training. Still, the model can learn information at a global level and has high consistency for each class. On the other hand, storing and updating the features in the memory bank for each iteration during training also create more variants. By leveraging these advantages, we propose the memory bank for few-shot camouflage instance segmentation. To be specific, we use the memory bank



to contain the background and foreground features per each class and make use of features to calculate the discrimination between areas of object and no object in region proposals (shown in [Figure 8](#)).

**Storing and updating:** The memory bank for each class contains  $2N$  features including  $N$  of foreground features and  $N$  of background features. While the memory bank receives new features, the module concatenates them with existing old features. In case the number of features is greater than the given  $N$  features, the memory bank releases the oldest features to maintain the number of features to  $N$ . This process updates the features in the memory bank and keeps the quantity of the stored features appropriate to the memory size (also known as the memory capacity).

**Sampling:** To calculate the loss value, the memory bank has to provide three elements  $\mathbb{F}_{fg}$ ,  $\mathbb{F}_{bg}$ , and  $\mathbb{F}_{general}$ .  $\mathbb{F}_{fg}$  and  $\mathbb{F}_{bg}$  are all foreground and background features that module storing. The  $\mathbb{F}_{general}$  is the general foreground feature, and it is created for each class by averaging the  $\mathbb{F}_{fg}$ .

Let  $\mathbb{F}_{fg}^i$  be the  $i$ -th foreground feature and  $\tau$  be a temperature hyper-parameter in [\[72\]](#). In our experiments, we set  $\tau$  as 1. The memory loss function for camouflaged instances is introduced as follows:

$$\mathcal{L}_{memory} = -\log \frac{\exp(\mathbb{F}_{general} \cdot \mathbb{F}_{fg}^i / \tau)}{\sum_{j=0}^{|\mathbb{F}_{bg}|} \exp(\mathbb{F}_{general} \cdot \mathbb{F}_{bg}^j / \tau) + \exp(\mathbb{F}_{general} \cdot \mathbb{F}_{fg}^i / \tau)} \quad (2)$$

To this end, the final loss of our training process, which contains an instance triplet loss and memory storage is defined as follows:

$$\mathcal{L}_{final} = \mathcal{L}_{mrcnn} + \alpha \mathcal{L}_{triplet} + \beta \mathcal{L}_{memory}. \quad (3)$$

Here, the parameter  $\alpha$  of  $\mathcal{L}_{triplet}$  and  $\beta$  of  $\mathcal{L}_{memory}$  are used during the training process to keep the balance between the two loss functions. Details of these functions are mentioned in the following section.

## 4 Experiments

We first overview the metrics and the experiment settings and the implementation details in [Section 4.1](#) and then we evaluate and discuss our improvement on the general framework, as well as ablation study for our core proposed methods in [Section 4.2](#).

### 4.1 Overview

As specified in this work, we utilize the proposed CAMO-FS dataset containing images of camouflaged animals in the wild to establish the evaluation of our baseline and proposed improvement. We follow the concept procedure published in FSOD [\[36–38\]](#). In the first stage of the base phase, we train our model with abundant data from 80 classes of the COCO dataset as proposed in [\[36\]](#). In the second stage of the novel phase, we evaluate the performance of having  $K = 1, 2, 3, 5$  shots per each novel class.

To report our results on detection and instance segmentation tasks, we use average precision (AP) and average recall (AR). To be detailed, we report AP@50 and AP@75, along with AR@10. Besides, we also report AP and AR at small, medium, and large scales of the instances to further understand the model performance. For more details,

**Table 4:** State-of-the-art comparison on CAMO-FS dataset among the baseline model of MTFA [70], Mask RCNN<sup>†</sup> [40], iFS-RCNN [61], and our proposed methods FS-CDIS with instance triplet loss (-ITL) and instance memory storage (-IMS). Our performance improves over the utilized baselines.

Model		Novel AP									
Method	Backbone/ Baseline	Instance Segmentation					Object Detection				
		1	2	3	5	Avg.	1	2	3	5	Avg.
MTFA [70]	COCO-80 ResNet-50	2.48	6.67	5.81	6.40	5.34	1.98	6.47	5.82	6.17	5.11
M-RCNN <sup>†</sup> [40]		4.08	6.79	6.90	8.29	6.52	2.82	5.09	5.46	6.18	4.89
iFS-RCNN [61]		4.17	6.26	5.73	6.38	5.64	3.92	6.06	5.47	6.60	5.51
MTFA [70]	COCO-80 ResNet-101	3.66	6.21	6.16	5.95	5.50	2.93	5.90	5.84	5.84	5.13
M-RCNN <sup>†</sup> [40]		4.39	7.69	7.94	10.09	7.53	3.03	5.80	6.20	7.79	5.71
iFS-RCNN [61]		4.27	6.55	6.07	7.80	6.17	3.79	6.28	6.01	8.08	6.04
<b>Our performance (↑)</b>											
FS-CDIS-ITL	ResNet-101	4.46	5.57	6.41	8.48	<b>6.23</b>	4.04	7.28	7.49	9.76	<b>7.14</b>
FS-CDIS-IMS	MTFA	5.46	6.95	7.36	9.61	<b>7.35</b>	4.50	6.95	7.55	10.36	<b>7.34</b>
FS-CDIS-ITL	ResNet-101	5.73	7.97	8.52	9.92	<b>8.04</b>	5.08	7.56	7.85	9.67	<b>7.34</b>
FS-CDIS-IMS	M-RCNN	5.52	7.84	8.65	9.82	<b>7.96</b>	4.92	7.39	7.96	9.52	<b>7.45</b>
FS-CDIS-ITL	ResNet-101	5.35	6.01	7.80	6.23	<b>6.35</b>	4.71	5.66	7.10	6.06	<b>5.88</b>
FS-CDIS-IMS	iFS-RCNN	2.99	6.83	6.14	9.03	<b>6.25</b>	2.74	6.39	5.94	8.44	<b>5.88</b>

*M-RCNN<sup>†</sup> is Mask R-CNN [40] with sigmoid classifier.*

readers can visit the homepage of the COCO dataset for detection and segmentation evaluation metrics <sup>1</sup>.

Our MTFA [70] baseline is implemented using Detectron2 framework [73]. Our backbone is ResNet-101 [74] with Feature Pyramid Network [75]. Each experiment is set up with a single GPU GeForce RTX 2080Ti with a batch size of 2 images. The novel phase has a learning rate of 0.00125 inferred from the MTFA configuration. We set the balance parameters  $\alpha = 1e^{-1}$  and  $\beta = 1e^{-2}$  when we train the model with instance triplet and instance memory loss function, respectively. Please visit the publication [37] or [73] for more details on other parameters.

## 4.2 Results and Discussion

**State-of-the-art comparison.** To prove the effectiveness of our proposed methods, we conducted experiments on our proposed CAMO-FS dataset. We tested with  $K = \{1, 2, 3, 5\}$  shots, respectively. Since several recent work have not published their source code [63, 64], we adopted the typical models addressing both detection and instance segmentation tasks to compare with our proposed methods. Table 4 presents the evaluation of the performance of our methods of instance triplet loss and memory storage over our baseline MTFA [70], the model of Mask R-CNN [40] with sigmoid classifier, and the state-of-the-art method iFS-RCNN [61] in the approach of few-shot instance segmentation. We reported experiments on those models and chose the common COCO-80 ResNet-101 as their base model to apply our proposed methods. The details of this decision are declared in the ablation section. In terms of instance segmentation, we improved over MTFA [70], Mask RCNN<sup>†</sup> [40], and iFS-RCNN [61] by getting average AP values of 6.23%, 8.04%, 6.35%, respectively thanks to instance triplet loss,

<sup>1</sup><https://cocodataset.org/#detection-eval>

**Table 5:** Our improvement of instance triplet loss and instance memory storage on MTFA [37]. The best performance is marked in **boldface**. # denotes the Number of shots, “Memory” is Instance Memory Storage and “Triplet” is Instance Triplet Loss.

#	Method	AP	AP50	AP75	APs	APm	API	AR1	AR10	ARs	ARm	ARI
<b>Instance Segmentation</b>												
1	Baseline MTFA	3.66	5.37	4.09	22.42	4.35	2.01	11.30	13.58	25.97	<b>12.96</b>	12.53
	MTFA + Triplet	4.46	8.21	4.60	21.33	4.13	<b>4.01</b>	12.36	15.04	23.17	9.49	16.67
	MTFA + Memory	<b>5.46</b>	<b>9.20</b>	<b>6.17</b>	<b>27.79</b>	<b>6.20</b>	<b>4.01</b>	<b>17.08</b>	<b>19.99</b>	<b>29.41</b>	11.45	<b>20.89</b>
2	Baseline MTFA	6.21	8.92	7.28	32.64	<b>7.75</b>	3.50	18.88	21.12	<b>35.82</b>	<b>15.49</b>	20.14
	MTFA + Triplet	5.57	9.45	6.04	25.83	3.01	5.37	15.67	17.33	26.13	7.37	17.50
	MTFA + Memory	<b>6.95</b>	<b>10.72</b>	<b>7.60</b>	<b>33.62</b>	5.73	<b>6.44</b>	<b>20.00</b>	<b>22.15</b>	34.25	13.86	<b>20.92</b>
3	Baseline MTFA	6.16	8.95	6.68	33.74	6.19	5.08	20.25	22.95	36.83	16.31	21.63
	MTFA + Triplet	6.41	10.67	6.72	30.39	5.17	5.30	20.69	22.98	31.90	15.69	22.53
	MTFA + Memory	<b>7.36</b>	<b>11.23</b>	<b>8.49</b>	<b>37.03</b>	<b>6.24</b>	<b>5.64</b>	<b>24.40</b>	<b>27.69</b>	<b>38.44</b>	<b>17.02</b>	<b>26.71</b>
5	Baseline MTFA	5.95	8.67	6.94	34.71	<b>6.25</b>	4.85	21.29	24.42	36.86	<b>14.51</b>	24.83
	MTFA + Triplet	8.48	13.43	9.80	36.66	5.75	8.04	23.83	26.66	37.03	11.62	25.91
	MTFA + Memory	<b>9.61</b>	<b>14.61</b>	<b>11.73</b>	<b>38.60</b>	5.79	<b>10.40</b>	<b>26.65</b>	<b>30.37</b>	<b>39.21</b>	12.26	<b>30.02</b>
<b>Object Detection</b>												
1	Baseline MTFA	2.93	5.86	2.20	20.95	4.18	2.03	9.25	10.84	21.74	<b>11.49</b>	8.77
	MTFA + Triplet	4.04	8.65	2.98	20.50	4.90	<b>4.22</b>	12.89	<b>15.53</b>	20.73	11.45	17.46
	MTFA + Memory	<b>4.50</b>	<b>9.14</b>	<b>3.45</b>	<b>22.88</b>	<b>5.61</b>	3.54	<b>13.14</b>	15.22	<b>23.14</b>	8.78	<b>16.33</b>
2	Baseline MTFA	5.90	8.87	6.83	33.04	9.74	3.10	17.26	19.25	34.04	<b>15.74</b>	19.61
	MTFA + Triplet	<b>7.28</b>	<b>11.22</b>	<b>8.25</b>	32.31	<b>10.72</b>	<b>6.83</b>	<b>20.52</b>	<b>22.69</b>	32.34	14.88	<b>23.52</b>
	MTFA + Memory	6.95	10.88	7.75	<b>33.93</b>	7.49	6.81	19.84	22.01	<b>34.10</b>	15.04	21.47
3	Baseline MTFA	5.84	8.98	6.29	34.56	7.78	4.31	19.13	21.83	35.80	15.93	21.09
	MTFA + Triplet	7.49	<b>11.51</b>	8.23	<b>38.45</b>	8.61	<b>6.38</b>	<b>24.88</b>	<b>27.52</b>	<b>38.55</b>	17.66	27.44
	MTFA + Memory	<b>7.55</b>	11.45	<b>8.50</b>	38.07	<b>9.21</b>	5.70	24.20	27.29	38.50	<b>18.10</b>	<b>27.56</b>
5	Baseline MTFA	5.84	9.13	6.04	35.44	8.17	4.22	19.67	22.96	35.94	<b>14.16</b>	22.58
	MTFA + Triplet	9.76	14.37	11.12	<b>40.05</b>	<b>8.82</b>	9.89	25.93	29.28	<b>40.05</b>	12.53	30.32
	MTFA + Memory	<b>10.36</b>	<b>16.27</b>	<b>11.79</b>	39.32	8.08	<b>11.36</b>	<b>26.34</b>	<b>30.30</b>	39.35	12.37	<b>30.91</b>

and 7.35%, 7.96%, 6.25%, respectively thanks to instance memory storage. Regarding object detection, our FS-CDIS got average amounts of 7.14%, 7.34%, 5.88%, respectively with instance triplet loss and 7.34%, 7.45%, 5.88%, respectively with instance memory storage. The detailed performance of our methods is in Table 4. Despite the limited results, we defeated the very early models on detection and instance segmentation tasks on camouflaged images.

**Proposed modules evaluation.** In Table 5, we also present the results of the baseline MTFA [70] with its original default configuration along with our proposed improvements. On top of the baseline MTFA [70], we establish fine-tuning configuration on this model by training all heads of classification, box regression, and mask prediction on few-shot novel data. The reported results prove the performance of the proposed instance triplet loss and instance memory storage.

In general, our approaches achieve outstanding results in comparison with the baseline. Our improvements surpass MTFA by a remarkable margin. Regarding the instance segmentation, our method improves 1.9%, 3.5%, and 2.3% in terms of AP, AP@50, and AP@75, respectively. These results manifest the efficiency of our methods in the context of few-shot camouflaged instances. Both loss functions enhance the discrimination between foreground and background features which strongly supports the model to segment pixels that belong to the camouflaged animals. Regarding the memory bank and the triplet loss function, the results of the memory loss function are higher than those of the triplet loss function by about 1%. We realize that storing representatives for each class is a crucial element in few-shot learning. This technique not

**Table 6:** Ablation study on the base model with 1-shot results. The best, and second best performances are marked in **red**, and *blue*, respectively. “Memory” is Instance Memory Storage and “Triplet” is Instance Triplet Loss.

Method	Base Model	Segmentation			Detection		
		AP	AP50	AP75	AP	AP50	AP75
Triplet	COCO-80 R-101	<b>4.46</b>	<b>8.21</b>	<b>4.60</b>	<b>4.04</b>	<b>8.65</b>	<b>2.98</b>
	COCO-80 R-50	3.68	<i>6.79</i>	3.81	2.85	<i>6.67</i>	1.65
	COCO-60 R-101	<i>3.87</i>	6.26	<i>3.90</i>	<i>3.37</i>	6.51	<i>2.69</i>
	COCO-60 R-50	2.56	4.25	2.79	2.28	4.13	2.26
Memory	COCO-80 R-101	<b>5.46</b>	<b>9.20</b>	<b>6.17</b>	<b>4.50</b>	<b>9.14</b>	<b>3.45</b>
	COCO-80 R-50	<i>3.87</i>	<i>6.81</i>	<i>3.91</i>	<i>3.40</i>	<i>6.94</i>	2.76
	COCO-60 R-101	2.89	4.50	3.26	2.76	4.66	<i>2.81</i>
	COCO-60 R-50	2.63	4.50	3.02	2.25	4.50	1.65

also expands the variants during training but also increases the consistency per class, thereby model can segment difficult objects better. In these ways, we also improve the corresponding results in camouflage object detection.

In [Table 5](#), our improvements help the model segment animals in various sizes. Specifically, all three metrics including APs, APm, and APi improve in comparison with the baseline model, which demonstrates that our model well segments small, medium, and large animals. This situation also happens in the detection problem. When data is very scarce as in a 1-shot or 2-shot setting, the triplet loss function has comparative results with the memory loss function. However, in the context of 3-shot or 5-shot settings, the memory loss function demonstrates outstanding efficiency thanking to storing and updating the memory via iterations to create discriminative features on a global level. [Figure 9](#) illustrates the qualitative comparison among the results of 5-shot settings of the baseline MTFA [70] and our proposed methods of Instance Triplet Loss and Instance Memory Storage. We chose to visualize the images with the confidence threshold of 0.5, which released a huge number of predictions with low confidence from the models. The two final rows are exemplary cases that either triplet loss or memory storage cannot well handle these camouflaged instances.

**Base model ablation study.** We also conduct ablation experiments on different backbone base models of the COCO settings including general and few-shot concepts. To be detailed, we report the performance of our proposed method of instance triplet loss and instance memory storage over four different backbones. The considered backbones are ResNet-50 and ResNet-101 [74]. The two base datasets are MS-COCO with 80 classes and 60 classes, respectively. Thus, it led to the combination of four different base models (i.e. COCO-80 R-101, COCO-80 R-50, COCO-60 R-101, and COCO-60 R-50). As can be seen from [Table 6](#), the performance of applying COCO-80 R-101 base weight yields better results among others evaluated on AP, AP@50, and AP@75 in both segmentation and detection tasks. In both cases of our two proposed improvements, the ablation results demonstrate our selection of COCO-80 R-101 is the best among the tested backbones of the base phase. For the segmentation task, we achieve 4.46 and 5.46 of AP reported for triplet loss and memory storage, respectively. For the detection task, we reach 4.04 and 4.50 also of AP for the two proposals, respectively.

**Table 7:** Ablation study on the margin and the  $\alpha$  ratio of the instance triplet loss in 1-shot settings. The best, and second best performances are marked in **red**, and *blue*, respectively.

AP		Segmentation					Detection				
$\alpha$	Margin	0	0.25	0.50	0.75	1.00	0	0.25	0.50	0.75	1.00
	1		3.89	4.50	3.92	<b>5.16</b>	4.43	3.34	3.65	3.22	<b>4.22</b>
$1e^{-1}$		<b>4.82</b>	<i>4.74</i>	4.46	<i>4.58</i>	<b>4.57</b>	<b>4.36</b>	<b>4.27</b>	<i>4.04</i>	<i>4.16</i>	<b>3.79</b>
$1e^{-2}$		<i>4.29</i>	<b>4.74</b>	<b>4.69</b>	4.46	<i>4.39</i>	<i>4.02</i>	<i>3.97</i>	<b>4.24</b>	4.06	3.71

**Table 8:** Ablation study on the capacity of the instance memory storage in 1-shot settings. The best, and second best performances are marked in **red**, and *blue*, respectively.

Capacity	Segmentation			Detection		
	AP	AP50	AP75	AP	AP50	AP75
32	4.56	7.30	5.02	3.85	7.72	2.91
64	4.51	<i>7.67</i>	4.49	3.94	<i>8.97</i>	2.84
128	4.53	7.55	4.87	4.13	7.98	3.62
256	4.56	7.50	5.02	4.01	8.22	3.39
512	<b>4.76</b>	7.57	<b>5.37</b>	<b>4.48</b>	8.25	<b>4.44</b>
1024	<i>4.72</i>	<b>8.06</b>	<i>5.20</i>	<i>4.14</i>	<b>8.45</b>	<i>3.79</i>

In summary, the chosen backbone of the base weight presents a higher performance of around 1% to 2% of evaluated on common metrics as in the table. To be explained, the base from COCO-80 contains more semantic concepts in comparison with the COCO-60 base, which leads to the higher performance reported. Note that all the results in this ablation section are reported for the 1-shot setting.

**Ablation on instance triplet loss component.** In terms of the instance triplet loss described in Eq. 1, we establish ablation experiments to evaluate the performance of the model with different configurations of margin and  $\alpha$  value. To this end, we set up the margin varying from 0 to 1, with a step of 0.25. For the  $\alpha$  ratio of the loss function (Eq. 3), we check out  $\alpha = \{1, 1e^{-1}, 1e^{-2}\}$ . To be enhanced, the margin value indicates how distinguished foreground and background features are. Meanwhile, the  $\alpha$  controls the effect of the instance triplet loss on the total loss of the framework. Table 7 presents the evaluation of both detection and segmentation issues in 1-shot manner. As can be inferred from the table, the effect of  $\alpha$  decides which margin should be selected for the triplet loss. With  $\alpha = 1$  meaning we keep the original ratio of the loss, the segmentation result in 1-shot setting yields the highest performance of 5.16% mAP with a 0.75 margin value. Meanwhile, the detection result gets the highest performance of 4.36% with  $\alpha = 1e^{-1}$  and zero margin. This table offers a better understanding of the impact of  $\alpha$  and the margin over the total performance.

**Ablation on instance memory storage component.** As for the instance memory storage, as introduced in Eq. 2, and Eq. 3, there are several parameters that need analyzing, listed as the amount of capacity in the memory storage and the  $\beta$  ratio

**Table 9:** Ablation study on the  $\beta$  ratio of the instance memory loss (Eq. 3) in 1-shot settings. The best, and second best performances are marked in **red**, and *blue*, respectively.

$\beta$	Segmentation			Detection		
	AP	AP50	AP75	AP	AP50	AP75
$1e^{-1}$	3.36	6.58	2.91	3.69	8.02	2.90
$1e^{-2}$	<i>4.57</i>	<i>8.02</i>	<i>4.74</i>	3.73	7.78	2.76
$1e^{-3}$	4.51	7.15	4.67	3.87	7.16	3.51
$1e^{-4}$	<b>5.12</b>	<b>8.71</b>	<b>5.54</b>	<b>4.58</b>	<b>9.23</b>	<b>3.69</b>
$1e^{-5}$	4.44	7.58	3.89	<i>4.06</i>	<i>7.99</i>	<i>3.63</i>

controlling the effect of the memory storage loss in the total loss. Table 8, and Table 9 present the ablation experimental results of those issues, respectively. In terms of the capacity of the memory storage, we establish experiments on a range of memory capacity of  $2^i$  where  $i = \{5, 6, 7, 8, 9, 10\}$ . The reported results figure out that the performance on both segmentation and detection tasks increases with a larger capacity of memory storage. To be detailed, with a capacity of 512, the mAP metric achieves the highest value among configurations, i.e. 4.76% and 4.48% for segmentation and detection, respectively. Empirically, we select 512 to be the suitable capacity of the memory storage, not the largest. To this end, the larger capacity can confuse the model in the process of learning when retrieving information in such a large memory bank. Besides, Table 9 expresses the effectiveness of the memory loss to the total loss function. As can be inferred,  $\beta = 1e^{-4}$  gives the best performance evaluated on mAP, AP50, and AP75 among all configurations.

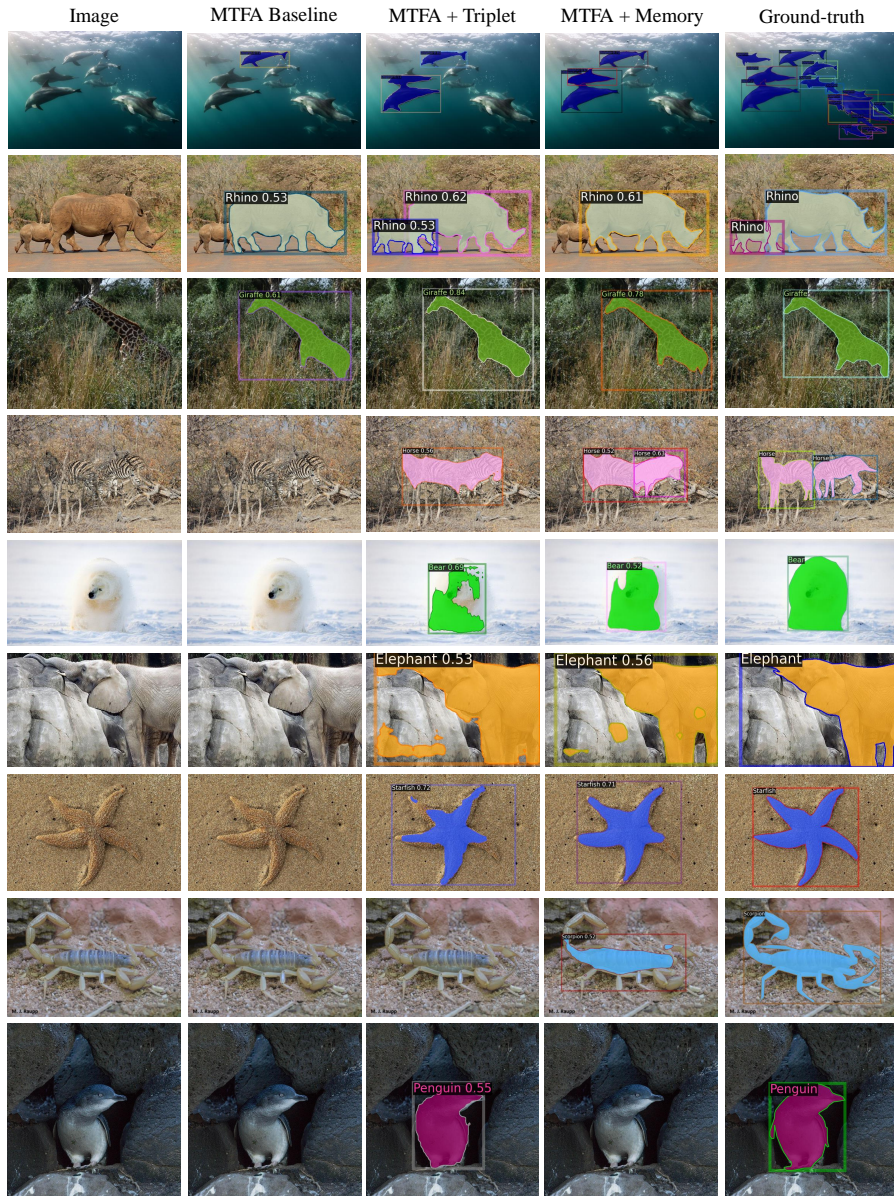
## 5 Conclusion

In this work, we investigated the interesting yet challenging problem of few-shot learning for camouflaged animal detection and segmentation. We first collect a new dataset, CAMO-FS, for benchmarking purposes. We then propose a novel method to efficiently detect and segment the camouflaged animals in the images. In particular, we introduce the instance triplet loss and the instance memory storage. The extensive experiments demonstrated that our proposed method achieves state-of-the-art performance on the newly constructed dataset. We expect our work will encourage more research work in this field. In the future, we would like to extend our work with more shots for new classes. In addition, we aim to improve the computational model by taking the context into consideration.

## Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.





**Fig. 9:** Qualitative comparison among the selected baseline MTFA [70] and our proposed methods. The results are from 5-shot settings. “Memory” denotes Instance Memory Storage and “Triplet” denotes Instance Triplet Loss. Predicted images are visualized with a confidence threshold of 0.5, which released a huge number of predictions with low confidence from the models. The two final rows indicate exemplary cases that either triplet loss or memory storage fails to handle camouflaged instances.



## Authors Contribution Statement

The authors confirm contribution to the paper as follows: study conception and design: Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Tam V. Nguyen; data collection: Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Nhat-Duy Nguyen; analysis and interpretation of results: Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Thanh-Toan Do, Tam V. Nguyen; supervision: Vinh-Tiep Nguyen, Thanh Duc Ngo, Minh-Triet Tran; draft manuscript preparation: Thanh-Danh Nguyen, Anh-Khoa Nguyen Vu, Nhat-Duy Nguyen. All authors reviewed the results and approved the final version of the manuscript.

## Ethical and Informed Consent for Data Used

The data used in this study did not involve ethical and informed consent.

## Data Availability and Access

The data will be made available upon request from the authors.

## Acknowledgments

This research was supported by the VNUHCM-University of Information Technology's Scientific Research Support Fund.

## References

- [1] Singh, S., Dhawale, C., Misra, S.: Survey of object detection methods in camouflaged image. *IERI Procedia* **4**, 351–357 (2013)
- [2] Le, T.-N., Nguyen, T.V., Nie, Z., Tran, M.-T., Sugimoto, A.: Anabranh network for camouflaged object segmentation. *CVIU* **184**, 45–56 (2019)
- [3] Le, T.-N., Nguyen, H.H., Yamagishi, J., Echizen, I.: Openforensics: Large-scale challenging dataset for multi-face forgery detection and segmentation in-the-wild. In: *ICCV* (2021)
- [4] Kervrann, C., Heitz, F.: A markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics. *IEEE TIP* **4**(6), 856–862 (1995)
- [5] Boykov, Y., Funka-Lea, G.: Graph cuts and efficient nd image segmentation. *IJCV* **70**(2), 109–131 (2006)
- [6] Li, X., Sahbi, H.: Superpixel-based object class segmentation using conditional random fields. In: *ICASSP*, pp. 1101–1104 (2011)

- [7] Sulimowicz, L., Ahmad, I., Aved, A.: Superpixel-enhanced pairwise conditional random field for semantic segmentation. In: ICIP, pp. 271–275 (2018)
- [8] Galun, M., Sharon, E., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: ICCV, pp. 716–723 (2003)
- [9] Song, L., Geng, W.: A new camouflage texture evaluation method based on wssim and nature image features. In: International Conference on Multimedia Technology, pp. 1–4 (2010)
- [10] Xue, F., Yong, C., Xu, S., Dong, H., Luo, Y., Jia, W.: Camouflage performance analysis and evaluation framework based on features fusion. *Multimedia Tools and Applications* **75**, 4065–4082 (2016)
- [11] Pan, Y., Chen, Y., Fu, Q., Zhang, P., Xu, X.: Study on the camouflaged target detection method based on 3d convexity. *Modern Applied Science* **5**(4), 152 (2011)
- [12] Liu, Z., Huang, K., Tan, T.: Foreground object detection using top-down information based on em framework. *IEEE TIP* **21**(9), 4204–4217 (2012)
- [13] P. Sengottuvelan, A.W., Shanmugam, A.: Performance of decamouflaging through exploratory image analysis. In: ICETET, pp. 6–10 (2008)
- [14] Yin, J., Han, Y., Hou, W., Li, J.: Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering* **15**, 2201–2205 (2011)
- [15] Gallego, J., Bertolino, P.: Foreground object segmentation for moving camera sequences based on foreground-background probabilistic models and prior probability maps. In: ICIP, pp. 3312–3316 (2014)
- [16] Fan, D.-P., Ji, G.-P., Sun, G., Cheng, M.-M., Shen, J., Shao, L.: Camouflaged object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2777–2787 (2020)
- [17] Song, L., Geng, W.: A new camouflage texture evaluation method based on wssim and nature image features. In: 2010 International Conference on Multimedia Technology, pp. 1–4 (2010). IEEE
- [18] Siricharoen, P., Aramvith, S., Chalidabhongse, T., Siddhichai, S.: Robust outdoor human segmentation based on color-based statistical approach and edge combination. In: The 2010 International Conference on Green Circuits and Systems, pp. 463–468 (2010). IEEE
- [19] Galun, M., Sharon, E., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: ICCV, vol. 3, p. 716 (2003)

- [20] Kavitha, C., Rao, B.P., Govardhan, A.: An efficient content based image retrieval using color and texture of image sub-blocks. *International Journal of Engineering Science and Technology (IJEST)* **3**(2), 1060–1068 (2011)
- [21] Xue, F., Yong, C., Xu, S., Dong, H., Luo, Y., Jia, W.: Camouflage performance analysis and evaluation framework based on features fusion. *Multimedia Tools and Applications* **75**(7), 4065–4082 (2016)
- [22] Hou, J.Y.Y.H.W., Li, J.: Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering* **15**, 2201–2205 (2011)
- [23] Skurowski, P., Abdulameer, H., Baszczyk, J., Depta, T., Kornacki, A., Kozie, P.: Animal camouflage analysis: Chameleon database. Unpublished Manuscript (2018)
- [24] Zhai, Q., Li, X., Yang, F., Chen, C., Cheng, H., Fan, D.-P.: Mutual graph learning for camouflaged object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12997–13007 (2021)
- [25] Li, A., Zhang, J., Lv, Y., Liu, B., Zhang, T., Dai, Y.: Uncertainty-aware joint salient object and camouflaged object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10071–10081 (2021)
- [26] Mei, H., Ji, G.-P., Wei, Z., Yang, X., Wei, X., Fan, D.-P.: Camouflaged object segmentation with distraction mining. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8772–8781 (2021)
- [27] Yan, J., Le, T.-N., Nguyen, K.-D., Tran, M.-T., Do, T.-T., Nguyen, T.V.: Mirror-net: Bio-inspired camouflaged object segmentation. *IEEE Access* **9**, 43290–43300 (2021)
- [28] Zhu, J., Zhang, X., Zhang, S., Liu, J.: Inferring camouflage objects by texture-aware interactive guidance network. In: *AAAI* (2021)
- [29] Lv, Y., Zhang, J., Dai, Y., Li, A., Liu, B., Barnes, N., Fan, D.-P.: Simultaneously localize, segment and rank the camouflaged objects. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11591–11601 (2021)
- [30] Price, N., Green, S., Troschianko, J., Tregenza, T., Stevens, M.: Background matching and disruptive coloration as habitat-specific strategies for camouflage. *Scientific reports* **9**(1), 1–10 (2019)
- [31] Le, T.-N., Nguyen, V., Le, C., Nguyen, T.-C., Tran, M.-T., Nguyen, T.V.: Camouflfinder: Finding camouflaged instances in images. In: *Proceedings of the AAAI*

- Conference on Artificial Intelligence, vol. 35, pp. 16071–16074 (2021)
- [32] Le, T.-N., Cao, Y., Nguyen, T.-C., Le, M.-Q., Nguyen, K.-D., Do, T.-T., Tran, M.-T., Nguyen, T.V.: Camouflaged instance segmentation in-the-wild: Dataset, method, and benchmark suite. *IEEE Transactions on Image Processing* **31**, 287–300 (2022)
  - [33] Pia Bideau, E.L.-M.: It’s moving! a probabilistic model for causal motion segmentation in moving camera videos. In: *ECCV* (2016)
  - [34] Lamdouar, H., Yang, C., Xie, W., Zisserman, A.: Betrayed by motion: Camouflaged object discovery via motion segmentation. In: *ACCV* (2020)
  - [35] Fan, Q., Zhuo, W., Tang, C.-K., Tai, Y.-W.: Few-shot object detection with attention-rpn and multi-relation detector. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
  - [36] Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: *ICCV* (2019)
  - [37] Wang, X., Huang, T.E., Darrell, T., Gonzalez, J.E., Yu, F.: Frustratingly simple few-shot object detection. In: *ICML* (2020)
  - [38] Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., Lin, L.: Meta r-cnn: Towards general solver for instance-level low-shot learning. In: *ICCV* (2019)
  - [39] Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271 (2017)
  - [40] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: *ICCV*, pp. 2980–2988 (2017)
  - [41] Li, B., Yang, B., Liu, C., Liu, F., Ji, R., Ye, Q.: Beyond max-margin: Class margin equilibrium for few-shot object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021)
  - [42] Hu, H., Bai, S., Li, A., Cui, J., Wang, L.: Dense relation distillation with context-aware aggregation for few-shot object detection. In: *Proceedings of (CVPR)* (2021)
  - [43] Xiao, Y., Marlet, R.: Few-shot object detection and viewpoint estimation for objects in the wild. In: *European Conference on Computer Vision*, pp. 192–210 (2020). Springer
  - [44] Han, G., He, Y., Huang, S., Ma, J., Chang, S.-F.: Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3263–3272 (2021)

- [45] Han, G., Huang, S., Ma, J., He, Y., Chang, S.-F.: Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 780–789 (2022)
- [46] Li, A., Li, Z.: Transformation invariant few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
- [47] Wu, J., Liu, S., Huang, D., Wang, Y.: Multi-scale positive sample refinement for few-shot object detection. In: ECCV (2020)
- [48] Zhang, G., Luo, Z., Cui, K., Lu, S.: Meta-detr: Image-level few-shot object detection with inter-class correlation exploitation. arXiv preprint arXiv:2103.11731 (2021)
- [49] Han, G., Ma, J., Huang, S., Chen, L., Chang, S.-F.: Few-shot object detection with fully cross-transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5321–5330 (2022)
- [50] Sun, B., Li, B., Cai, S., Yuan, Y., Zhang, C.: Fsce: Few-shot object detection via contrastive proposal encoding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7352–7362 (2021)
- [51] Zhang, S., Wang, L., Murray, N., Koniusz, P.: Kernelized few-shot object detection with efficient integral aggregation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19207–19216 (2022)
- [52] Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., Zhang, C.: Defrcn: Decoupled faster r-cnn for few-shot object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8681–8690 (2021)
- [53] Zhang, W., Wang, Y.-X.: Hallucination improves few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13008–13017 (2021)
- [54] Zhu, C., Chen, F., Ahmed, U., Shen, Z., Savvides, M.: Semantic relation reasoning for shot-stable few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
- [55] Khandelwal, S., Goyal, R., Sigal, L.: Unit: Unified knowledge transfer for any-shot object detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
- [56] Liu, W., Zhang, C., Lin, G., Liu, F.: Crnet: Cross-reference networks for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

- [57] Dong, N., Xing, E.P.: Few-shot semantic segmentation with prototype learning. In: BMVC, vol. 3 (2018)
- [58] Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: Panet: Few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
- [59] Saha, O., Cheng, Z., Maji, S.: Ganorcon: Are generative models useful for few-shot segmentation? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9991–10000 (2022)
- [60] Tian, Z., Lai, X., Jiang, L., Liu, S., Shu, M., Zhao, H., Jia, J.: Generalized few-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11563–11572 (2022)
- [61] Nguyen, K., Todorovic, S.: ifs-rcnn: An incremental few-shot instance segmenter. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7010–7019 (2022)
- [62] Gao, B.-B., Chen, X., Huang, Z., Nie, C., Liu, J., Lai, J., Jiang, G., Wang, X., Wang, C.: Decoupling classifier for boosting few-shot object detection and instance segmentation. In: NeurIPS 2022 (2022)
- [63] Han, Y., Zhang, J., Xue, Z., Xu, C., Shen, X., Wang, Y., Wang, C., Liu, Y., Li, X.: Reference twice: A simple and unified baseline for few-shot instance segmentation. arXiv preprint arXiv:2301.01156 (2023)
- [64] Wang, H., Liu, J., Liu, Y., Maji, S., Sonke, J.-J., Gavves, E.: Dynamic transformer for few-shot instance segmentation. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 2969–2977 (2022)
- [65] Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. *Advances in neural information processing systems* **30** (2017)
- [66] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. *Advances in neural information processing systems* **29** (2016)
- [67] Gidaris, S., Komodakis, N.: Dynamic few-shot visual learning without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4367–4375 (2018)
- [68] Fan, Z., Yu, J.-G., Liang, Z., Ou, J., Gao, C., Xia, G.-S., Li, Y.: Fgn: Fully guided network for few-shot instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9172–9181 (2020)
- [69] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS, pp. 91–99 (2015)

- [70] Ganea, D.A., Boom, B., Poppe, R.: Incremental few-shot instance segmentation. In: Proceedings of (CVPR), pp. 1185–1194 (2021)
- [71] Balntas, V., Riba, E., Ponsa, D., Mikolajczyk, K.: Learning local feature descriptors with triplets and shallow convolutional neural networks. In: Bmvc, vol. 1, p. 3 (2016)
- [72] Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3733–3742 (2018)
- [73] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R.: Detectron2. <https://github.com/facebookresearch/detectron2> (2019)
- [74] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
- [75] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)