

UC Berkeley

UC Berkeley Electronic Theses and Dissertations

Title

Representation Learning for Efficient Localized Image Retrieval

Permalink

<https://escholarship.org/uc/item/8mm0j761>

Author

Jaffe, Lucas

Publication Date

2024

Peer reviewed|Thesis/dissertation

Representation Learning for Efficient Localized Image Retrieval

by

Lucas Jaffe

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Doctor of Philosophy

in

Engineering — Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Avidoh Zakhori, Chair
Professor Anant Sahai
Professor Trevor Darrell
Dr. Wesam Sakla

Spring 2024

Representation Learning for Efficient Localized Image Retrieval

Copyright 2024
by
Lucas Jaffe

Abstract

Representation Learning for Efficient Localized Image Retrieval

by

Lucas Jaffe

Doctor of Philosophy in Engineering — Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Avideh Zakhor, Chair

Locating and labeling objects in images remains a central problem in computer vision, and the challenge of doing so efficiently has taken on increasing importance with the deployment of autonomous vehicles, edge recognition applications, and ever-growing foundation models. We consider a variation of object localization called localized image retrieval, which combines the problems of image retrieval and query localization, by ranking the precise sub-regions in a set of scenes which match the user query. In this work, we will examine the topic of representation learning for localized image retrieval, specifically through the lens of the person search problem. We will show novel work in categorizing and improving efficiency of person search models during training and inference by introducing new modular approaches, comparing query-centric and object-centric methods, and developing methods for weakly-supervised and self-supervised pre-training. Finally, we will discuss how identity search, prompt localization, object detection, and tracking can be unified under the single problem of localized image retrieval, and consider connections to generalist models.

Dedication: To Megan and my family

Contents

Contents	ii
List of Figures	iii
List of Tables	iv
1 Introduction	1
1.1 Problem Formulation	2
1.2 Problem Types	4
1.3 Technical Motivation	8
1.4 Dissertation Organization	9
2 Background	10
2.1 Object Detection	10
2.2 Image Retrieval	16
2.3 Person Search	19
2.4 Unsupervised Learning	29
2.5 Model Fine-Tuning	30
2.6 Foreword on Efficiency	31
3 Modular Efficiency	32
3.1 Modularizing Detection and Re-identification	33
3.2 Modularizing Scene and Instance Retrieval	37
3.3 Methods	39
3.4 Experiments and Analysis	47
3.5 Discussion	60
4 Algorithmic Efficiency	62
4.1 Comparing Detector Efficiency	62
4.2 Query-Centric and Object-Centric Models	64
4.3 Methods	65
4.4 Experiments and Analysis	71
4.5 Discussion	75

5	Data Efficiency	76
5.1	Pre-training for Person Search	76
5.2	Methods	78
5.3	Experiments and Analysis	82
5.4	Discussion	94
6	Conclusions and Future Work	97
6.1	A Unified Framework of LIR	97
6.2	Generalist Models	101
6.3	Summary of Main Findings	103
6.4	Societal Impact	104
A	Supplementary Material: Modular Efficiency	105
A.1	Data Processing and Evaluation	105

List of Figures

1.1	Person Search Example (1)	2
1.2	Person Search Example (2)	3
1.3	Person Search Example (3)	3
1.4	Person Search Example (4)	4
1.5	LIR Problem Types	5
2.1	Anchor Diagram from Faster R-CNN	11
2.2	DETR	12
2.3	DETR	14
2.4	Detector Family Comparison	15
2.5	Pix2Seq	15
2.6	Metric Learning	17
2.7	One-Step vs. Two-Step Person Search	18
2.8	Robust Partial Matching	22
2.9	Robust Partial Matching	23
2.10	Context Graph for Person Search	25
2.11	Diverse Knowledge Distillation	26
2.12	Mask-guided Two-stream CNN Model	27
2.13	Norm-Aware Embeddings	27
3.1	One-Step vs. Two-Step Person Search	34
3.2	Ground Truth vs. Detected Box Mismatch	34
3.3	GFN Filtering Process	38
3.4	SeqNet Model Architecture	40
3.5	SeqNeXt Model Architecture (full)	40
3.6	GFN Optimization Issue Diagram	43
3.7	mAP vs. Gallery Size for Person Search Models	51
3.8	GFN Score Histograms for CUHK-SYSU and PRW	57
3.9	CUHK-SYSU and PRW Retrieval Example Images	59
3.10	Pooled Scene Features for GFN	60
3.11	GFN Modularity	61
4.1	SPNet Model Diagram	67

4.2	QC and OC Pathways of the SP Block	69
5.1	Person Search Pre-training Approach Comparison	77
5.2	QC vs. OC Detection Comparison	79
5.3	Labels for Different Supervision Levels	80
5.4	SPNet Pre-train Optimization Hyperparameter Comparison	90
5.5	SPNet Architecture Hyperparameter Comparison	92
5.6	SPNet Anchor Probability Visualization	93
5.7	SPNet Pre-training Limited Sample Comparison	94
5.8	SSL Training Loss Plots	95
6.1	Unified Problem Diagram	98
6.2	Unified Model Diagram	99
6.3	UNINEXT Model Diagram	100

List of Tables

3.1	SOTA Person Search Benchmark	50
3.2	Occluded and Low-Resolution Performance of Person Search Models	51
3.3	Same and Cross Camera Person Search Performance	52
3.4	GFN Objective Comparison	53
3.5	SeqNeXt Model Backbone Comparison	54
3.6	SeqNeXt Image Augmentation and Crop Size Comparison	55
3.7	SeqNeXt and GFN Pooling Size and Embedding Dimension Comparison	56
3.8	GFN Sampling Option Comparison	56
3.9	Compute Saved by GFN	58
3.10	GFN Time Saved	58
4.1	Detector Comparison	64
4.2	QC vs. OC Time Complexity	65
4.3	SPNet Architecture and Optimization Configurations	72
4.4	QC vs. OC Training Comparison	73
4.5	SOTA Person Search Benchmark	74
4.6	BatchNorm vs. FixedNorm Logits Comparison	75
5.1	SPNet Architecture and Optimization Configurations	85
5.2	Person Search Pre-training Time Comparison	85
5.3	SOTA Person Search Model Comparison	87
5.4	SOTA Person Search Benchmark with Pre-training	88
5.5	QC vs. OC Pre-training Comparison of Metrics	88
5.7	SPNet Architecture and Optimization Configurations	89
5.8	QC vs. OC Fine-tuning Comparison	91
5.9	SSL Random Init. Fine-tuning	95
5.10	SSL IN1k Classifier Fine-tuning	96
A.1	CUHK-SYSU and PRW Dataset Metadata	106

Acknowledgments

I want to thank a long list of people I have been lucky to know and work with over many years. To constrain the scope of these acknowledgements under the size of the dissertation itself, I will focus on people from during my PhD, but there are so many others that got me to where I am in the first place. Without naming those many names, I will thank them first.

Next I want to thank my committee members. First, my advisor Avidah Zakhori, who has advised and accommodated me over these last five years during my less than typical quest to earn a PhD while working at LLNL. I want to thank her for her feedback on papers and presentations, which always cut through to key underlying concepts when I was mired in details. And, for her guidance on how to move forward when I was stuck. I want to thank Anant Sahai for bringing me on to GSI for EECS182, where I learned so much more about deep learning, even at the end of my PhD. I want to thank Trevor Darrell, for making the time to be a member of my committee, and for organizing many BAIR workshops which helped me understand the state of the field. And I want to thank Sam Sakla, who taught me everything I knew about deep learning before my PhD, and a good bit after I started it too. Sam was always open to helping me practice talks, give feedback on papers, and to develop ideas at Livermore and Berkeley. Thank you Sam, for your mentorship over many years.

I want to thank many others at LLNL, who have been my coworkers and friends now for a long time. To Jeff Mast, thank you for teaching me software engineering - someday I will have to start using it. To Jim Morton, Paul Rawson, David Conner, and Nate Bahr, thank you for all your support when you had so many other things to do. To Omar DeGuchy, thank you for good conversations on deep learning and board games. To Mike Zelinski, thank you for many technical discussions that shaped the course of my research as I went into my PhD. To Carol Musto and the LLNL Employee Tuition Assistance Program, thank you for your support and guidance in navigating the process of funding my education. To Cindy Gonzales, thank you for being a good coworker, boss, and friend. And to Mike Goldman, who has supported me over many years, thank you for everything.

Thank you to my lab members in the VIP Lab at Berkeley, especially Jerome Quenum and Dennis Choi, who started the program with me in 2019. Thank you to Shirley Salanio, for your invaluable administrative support over my whole time at Berkeley, and Jean Nguyen for seamlessly stepping in to support me at the end. Thank you to everyone I taught with as a GSI in Math54 and EECS182. Thank you to BAIR and all its members and staff for your support and the great events. To many others at Berkeley, professors, students, and staff, thank you for everything you taught me. I have never learned so much in such a short time as I did here.

To my friends, who I have not gotten to see nearly as much as I would like over these last five years: Thank you to Matt Worthington, for many good times climbing and skiing. Thank you to Cameron Cornell, for always being willing to play the most complex board games. Thank you to Alyssa Troksa and Bruce Yang, for many good adventures, and many more to come. Thank you to Bob and Jeannine Freyman for welcoming me into your home, and many cookies sent in the mail. Thank you to many others: Maira Ceron, Eder Valenzeula,

Hanna Anderson, Auston Clemens, Buddhinie Jayathilake, Shamal Perera, Marissa Wood, Anne Cardenas. And to anyone I may have missed. I hope to see all of you more soon.

To my family, I have been lucky to have you all so nearby, and I am grateful for that. To my parents, Emily and David, thank you for your unfailing support throughout my life. To my brother Paul and (soon-to-be) sister-in-law Susanne, thank you for many fun gatherings, and for your recently-lived advice on getting a PhD too.

Most of all, I would like to thank Megan, who has been here for me through everything. For her humor, energy, and presence of mind. For so many tasty meals that she shared with me. And, for encouraging me, always, on the many occasions when I doubted myself. Now that I'm finally done, I will get to cook for you more too.

I will close with some wisdom from Martin Wainwright, who I would also like to thank for making optimization algorithms just a little bit easier to understand. Prof. Wainwright came into class one Spring morning early in 2020, shortly before lectures were made remote due to COVID, and stared pensively at the floor before saying, *“Life... is not an optimization problem. Hm... except when you’re in here.”* And with that, we can start discussing the optimization problem.

Chapter 1

Introduction

We begin by reviewing the definitions of two visual recognition problems: image retrieval and localization. In image retrieval, the goal is to retrieve relevant images to a query from a database, and in localization, the goal is to determine the precise location of some object or query in an image. Localized image retrieval (LIR) is the visual recognition problem which combines image retrieval and localization, by retrieving the sub-regions in a set of images which match a query. The query can be any arbitrary representation, but is commonly derived from either imagery or text. For example, we could form a query using the phrase, “A tall man with dark hair wearing a suit.”, or we could form a query with an image of the specific man we are trying to find. In either case, we would expect to return a list of images showing the most similar person in the scene database to our query, if there are any good candidates.

To set the stage, we walk through an example of LIR for the *person search* problem, depicted in Fig. 1.1. In person search, the goal is to localize and rank matches to a *query* person image in a set of *gallery* scene images. In the example shown in Fig. 1.1, the aim is localize Person A in Scenes 1 and 2. This example demonstrates the *two-step* approach, which will be described in more detail, and compared with a *one-step* approach, in Chapters 2 and 3.

In two-step person search, the first step is to detect all persons in the gallery scenes, shown in Fig. 1.2. Once all persons have been detected, the second step is to input all detected person image crops through a separate neural network to produce embeddings $x \in \mathbb{R}^d$, shown in Fig. 1.3a. Then, we compare query and gallery embeddings, typically using cosine similarity, shown in Fig. 1.3b. We note that any similarity measure or distance metric can be optimized for and used, but cosine similarity is a standard choice. In this case, the goal is for the cosine similarity between positive matches to be greater than that for negative matches.

If this relationship holds in every case, then we will achieve the correct ranking, an example of which is shown in Fig. 1.4. Ideally, the ranking will reflect visual similarity of each detected person with the query, indicating that we have learned a meaningful embedding space. We may also desire the property that embeddings of same person/identity form separable clusters in the embedding space. When embedding cluster sizes and inter-cluster



Figure 1.1: In person search, the goal is to localize and rank matches to the query person outlined in blue (Person A), in the gallery scenes (Scene 1,2). Matches to the query person in the gallery scenes are denoted by green boxes. Image example is from the CUHK-SYSU dataset [1], originally from the movie *The Imitation Game*.

distances are normalized, we can use a fixed similarity/distance threshold to retrieve all matches for a given embedding, without false positives.

In the remainder of this chapter, we will discuss a formal problem formulation of LIR in Sec. 1.1, the common visual recognition problem types which can be framed through LIR in Sec. 1.2, technical motivation for exploring the topic of LIR in Sec. 1.3, and organization of this dissertation in Sec. 1.4.

1.1 Problem Formulation

Having explained the definition of LIR at a high level, we will now propose a formalization that is sufficiently broad to capture multiple problem types, but still opinionated enough to be useful: given a query q with embedding $x_q \in \mathbb{R}^d$ and a database of images $\mathcal{D} = \{I_1, \dots, I_n\}$, where image I has height h , width w , and c channels, *i.e.* $I \in \mathbb{R}^{h \times w \times c}$, the LIR function f_{LIR} maps $x_q, \mathcal{D}$ to localizations l_c and embeddings x_c . Localizations l_c denote the position and extent of candidate matches c to query q in images in $\mathcal{D}$, and embeddings x_c encode similarity of c to q . A localization could be a point or arbitrary polygon, but is typically a rectangular bounding box b_c . Similarity of q and c is typically encoded as cosine similarity between x_q and x_c , but can be encoded via a distance metric as well. Outputs of LIR may be used for retrieval, by ranking with similarity scores, or filtered altogether via a score threshold.

In addition, we provide the following clarifications:

- The query q can be any piece of information we want to localize in the images, including

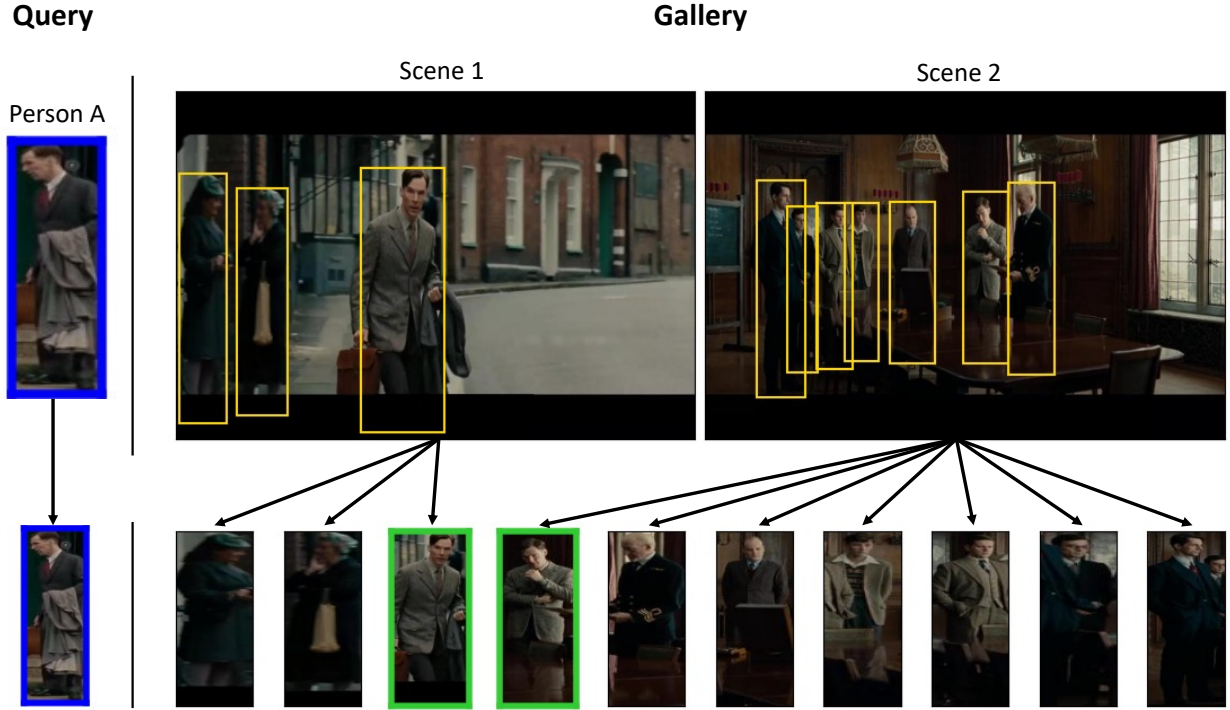
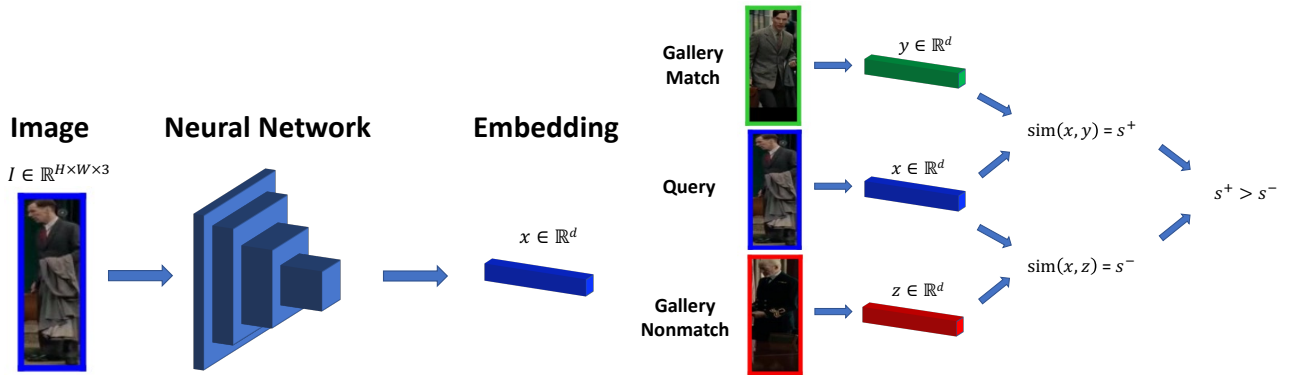


Figure 1.2: The first step in (two-step) person search is to detect all people in the gallery scenes (outlined in yellow), then crop the images corresponding to each detected person. Image crops matching the query person are outlined in green.



(a) Cropped persons are fed through a re-id person crop embedding network to produce person embeddings $x \in \mathbb{R}^d$.

(b) Person embeddings can then be compared using cosine similarity: the cosine similarity of matching pairs should be greater than the cosine similarity of non-matching pairs.

Figure 1.3: Person embedding extraction (left) and comparison (right) processes.



Figure 1.4: All detected person embeddings are compared to the query, which is outlined in blue, and ranked by cosine similarity, left to right in the graphic. Matches to the query are outlined in green.

an image, text, audio, video, or a point / line / squiggle in the image itself. We use the query by embedding it in $\mathbb{R}^d$ with an encoder specific to the corresponding data modality.

- Detected localizations may also have corresponding “objectness” scores or per-class object scores s_c .
- Detected localizations may have corresponding object masks m_c , which give finer-grained information about object extent.
- The database of images $\mathcal{D}$ could consist of only one image. Since there are multiple candidates within each image by design, the concept of retrieval still applies in this context.

1.2 Problem Types

Many existing models for various LIR problems have similar structure, but the concepts are often developed independently. Therefore a core motivation of this work is to understand links

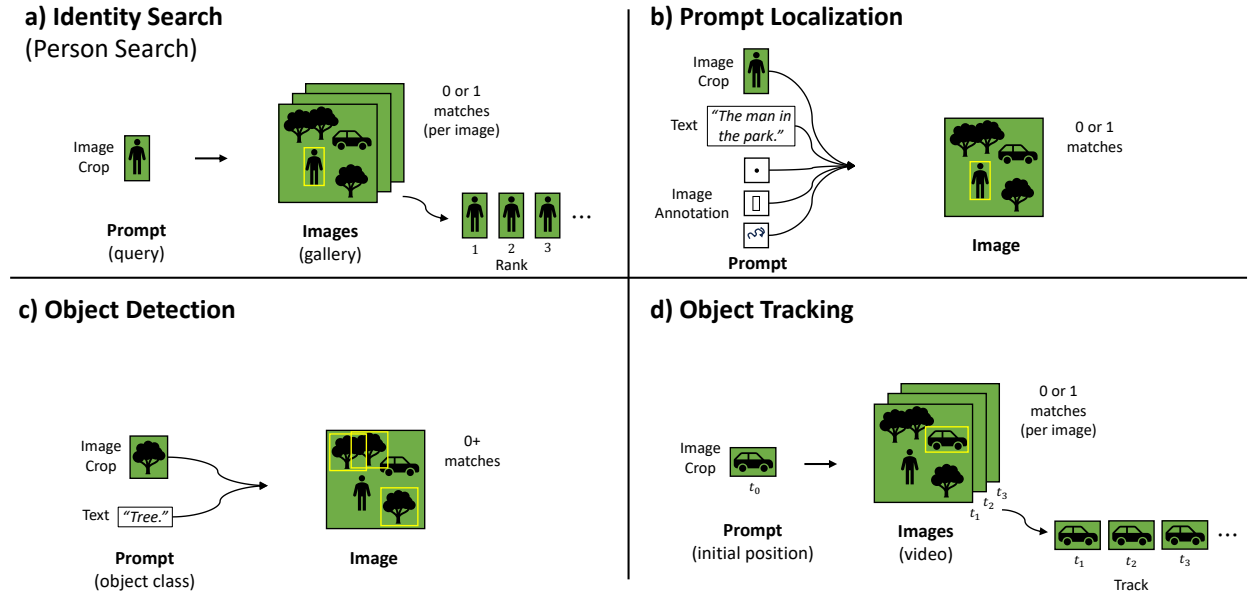


Figure 1.5: Depiction of the four basic problem categories in LIR: identity search, prompt localization, object detection, and object tracking. All problem types localize a prompt in one or more images, but the type of prompt, number of matches per image, and post-processing may differ between the problem types.

between these different instantiations of the same fundamental problem. Below, we give a brief overview of four major problem types which can be framed through LIR. These problem types are *identity search* in Sec. 1.2.1, *prompt localization* in Sec. 1.2.2, *object detection* in Sec. 1.2.3, and *object tracking* in Sec. 1.2.4. We visualize these problem types in Fig. 1.5.

The goal of unifying the problem types is to enable:

- Shared model architecture and inductive biases.
- Unified pre-training, fine-tuning, and inference.
- Cross-task transfer, improving performance for all tasks.

1.2.1 Identity Search

The main subject of our work in LIR is the problem of identity search, in which we aim to localize a query identity image in a set of gallery scenes, and rank results by similarity to the query, shown in Fig. 1.5a. An identity in this case typically refers to some fixed object in the physical world, typically a person or vehicle, but can refer to any object which can be defined as a fixed instance in some context. Identity search is the generalization of person search [2]–[4] to any object type, with the constraint being that we are still performing retrieval at

the instance level. We consider identity search the core problem of LIR, because all other problems can be defined in terms of it. However, we will show that many of the distinctions between LIR problem types are essentially arbitrary.

1.2.2 Prompt Localization

Prompt localization is the task of localizing a “prompt” in a single image, shown in Fig. 1.5b. It is similar to identity search, but we do not consider ranking results in separate gallery scenes with respect to similarity to the prompt. A prompt is synonymous with a query in the way we have defined it, but may refer to an image, a text sequence, or any other data which we can encode as an embedding. Prompt localization models such as SAM [5], SEEM [6], SegGPT [7], and UNINEXT [8] encode quantities such as images, text, and image sub-region markings (points, boxes, scribbles), and localize the given prompt within the target image. In fact, modern models of prompt localization cover most of our requirements for a unified model of LIR, except that existing works have not considered how to incorporate identity-level learning, or considered the retrieval aspect with multiple images, as in identity search.

Other tasks in prompt localization include Referring Expression Comprehension (REC) [9]–[11], Referring Expression Segmentation (RES) [10], [12], [13], and Referring Video Object Segmentation (R-VOS) [14]–[16]. In REC, the goal is to localize a natural language description in an image using a bounding box. RES is the extension of REC to include instance segmentation masks. R-VOS is the extension of RES to video, without an explicit goal of instance tracking.

1.2.3 Object Detection

In object detection [17]–[23], the goal is to localize all instances of objects belonging to a set of classes, shown in Fig. 1.5c. While prompt localization is defined at the instance level (locating a single object matching the prompt), object detection is defined at the class level, meaning that there may be more than one match for a given object class in a given image. Instance segmentation [24]–[27] is the extension of object detection to include instance segmentation masks. Video Object Segmentation (VOS) [28]–[30] is the extension of instance segmentation to video, without an explicit goal of instance tracking.

There are also variations of object detection that depend on how many training examples images we have of each class. In one-shot [31] and few-shot [32] detection, we have only one or a few examples respectively from each class we want to detect. These can both be framed as a variant of prompt localization by treating the training examples as the prompts, and allowing multiple matches per image.

Finally, zero-shot detection is the problem of detecting object categories that we have no image examples of. This problem can be framed as prompt localization, through the use of a text or joint image-text embedding space, in which the category name is treated as text and looked up from the embedding space, then used to localize matches. In OWL-ViT [22], CLIP [33] is used for the image-text embedding space.

1.2.4 Object Tracking

Object tracking, shown in Fig. 1.5d, can be framed as a variant of identity search, in which gallery scenes compose video frame sequences. In the naïve approach, tracking can be achieved by treating each frame as a separate gallery image and using an identity search model. Methods such as [8] show that this naïve approach is sufficient to achieve SOTA tracking performance on common benchmarks, though it is possible to add additional modules which learn explicit temporal features [34], with LIR modules functioning as a “tracking backbone”.

Variants of tracking include Single Object Tracking (SOT) [35]–[37], Multiple Object Tracking (MOT) [38]–[41], Multiple Object Tracking and Segmentation (MOTS) [42]–[44], and Video Instance Segmentation (VIS) [34], [45]–[47].

In SOT, we consider only a single object, and are given the position of the object in the first frame, and must track it through all video frames. In MOT, we usually have to first detect all objects (belonging to certain classes), and then track them each independently through all video frames. MOTS is the extension of MOT to include instance segmentation masks, and essentially equivalent to VIS, but more focused on the simultaneous tracking and segmentation of multiple objects.

1.2.5 Summary

Having delineated each problem type, it should be clear that the distinctions are somewhat arbitrary. Some questions that might occur are: How exactly is a prompt different from an object class? Are VOS and VIS really separate problems? Or, what is the difference between identity search and tracking, if tracking can be solved with identity search? The fact is that these are all essentially the same problem, and can be solved with a single model based on identity search and prompt localization, if the prompts are carefully designed. We will show one approach to doing so in Chapter 6.

1.3 Technical Motivation

Before approaching the main technical content in detail, we aim to clearly motivate the technical purpose of this work. In particular, we describe below the motivation for each part of the subject, *Representation Learning for Efficient Localized Image Retrieval*, and the rationale of our focus on *person search* in particular. In Sec. 1.3.1, we discuss why the joint problem LIR is worth studying. In Sec. 1.3.2, we discuss the motivation for framing our research through the lens of computational efficiency. In Sec. 1.3.3, we explain our focus on representation learning. Lastly, in Sec. 1.3.4, we explain why it is logical to study LIR through the problem of person search.

1.3.1 Localized Image Retrieval

One obvious question arises when we consider the joint problem of localized image retrieval: why not just consider detection and retrieval separately? For person search in particular, there exist both two-step models, which consider the two problems separately, and one-step or end-to-end models, which consider the problems jointly. However, we will show that even for two-step models, it is important to consider the process of detection during retrieval training. We briefly discuss the key tradeoffs between the two model types below, emphasizing the importance of LIR as a distinct topic of study. A detailed treatment is given in Chapter 3.

- One-step models are generally more efficient at inference time, because they share features between detection and retrieval, *i.e.* we don't need to recompute certain features for each task.
- The distribution of detected bounding boxes will not exactly match perfect ground truth bounding boxes, meaning that it must be considered for retrieval model training.
- Some tasks, including variants of identity search and prompt localization that depend on very specific instance prompts, can only be performed as LIR, because there exists no concept of an object class.
- Relatedly, considering the tasks jointly allows for a unified model solving multiple problem types in LIR, discussed further in Chapter 6. Approaches such as [8] show that a unified generalist approach can achieve better performance than specialist models developed for each task.

1.3.2 Efficiency Framing

The primary focus of this work is to analyze localized image retrieval through the framing of computational efficiency. Having established that the problem of LIR is itself a useful delineation, it is clear that having efficient training and inference are beneficial in their own right. But beyond that, efficient models enable deployment in time sensitive and edge applications like automated vehicles and remote sensing. In addition, the trend of large models in language and vision has effectively priced out small and medium-sized research institutions in developing relevant models to an extent. Therefore any mechanisms which improve training efficiency are desirable in that they can allow model training or fine-tuning with fewer compute resources.

1.3.3 Representation Learning

The mechanisms of improving LIR efficiency that we discuss are specifically focused on representation learning. In Chapter 3, we focus on learning person and scene representations that allows us to efficiently filter scenes per person by comparing their representations.

In Chapters 4 and 5, we develop query and anchor representations that have a specific relationship, allowing both query-centric and object-centric detection.

1.3.4 Focus on Person Search

While we have mentioned many problem types that fall under the category of LIR, we focus specifically on person search because it is the most well-studied example of identity search as we have defined it. As discussed, identity search is a core problem of LIR, and it can be used to solve all the other problem types, so it is logical to base model development around it. Therefore, while it would be beneficial to conduct experiments on the other problem types with a variety of datasets, person search serves as a reasonable starting point for model research and development on the topic of unified LIR.

1.4 Dissertation Organization

Chapter 1 describes and motivates the purpose of this work and its organizational structure, and Chapter 2 covers related work which is pre-requisite to understanding the main research contribution. In Chapters 3 to 5, we analyze various framings of LIR efficiency, and introduce new modeling mechanisms for person search. In Chapter 3, we describe two methods of modularizing LIR, including one-step vs. two-step models, and our new model, the Gallery Filter Network [3]. In Chapter 4, we analyze time complexity and computational efficiency of different detectors, and of query-centric vs. object-centric LIR models. We introduce a new model for person search (SPNet) which can operate in both query-centric and object-centric modes with the same weights [4]. In Chapter 5, we analyze pre-training and fine-tuning performance of person search models, and develop a new pre-training mechanism for person search using SPNet, which can use weakly-labeled or unlabeled data [4]. In Chapter 6, we discuss future work on the topic of unifying LIR, and how to improve existing models of unified LIR. In addition, we summarize our main results and discuss societal impacts of the topic area.

Chapter 2

Background

The intent of this Background chapter is to cover pre-requisite material that is required for understanding of our contributions, in a way that is compact and separate from our contribution description. We focus especially on variants of object detection and person search, since these are the most important and relevant topics. We also cover image retrieval, unsupervised learning, and model fine-tuning. Aside from the focus on person re-identification (re-id), we refer to the related area of vehicle re-id, which is the other major object category in instance recognition research.

The outline of this chapter is as follows: we discuss object detection in Sec. 2.1, image retrieval in Sec. 2.2, person search in Sec. 2.3, unsupervised learning in Sec. 2.4, model fine-tuning in Sec. 2.5, and provide a foreword to the efficiency chapters in Sec. 2.6.

2.1 Object Detection

In this section, we focus on the Object Detection problem, covering the major categories of models, and highlighting methods which are relevant to our research approach. In Chapter 4, we argue that anchor-based models are still the best choice as the core of our modeling approach, despite there being many other options. Material in this section will examine the inductive biases and practical structure of each model type which make this choice clear.

We discuss spatially-referenced object detection models in Sec. 2.1.1, set prediction object detection models in Sec. 2.1.2, sequence prediction object detection models in Sec. 2.1.3, and details of bounding box regression for object detection in Sec. 2.1.4.

2.1.1 Spatially-Referenced Models

We define spatially-referenced models of object detection as those which implement an explicit correspondence between spatial locations in an image and features computed for that image. Prior to deep learning, all object detection models were spatially-referenced, and typically computed handcrafted features for each image location, and predicted bounding boxes either

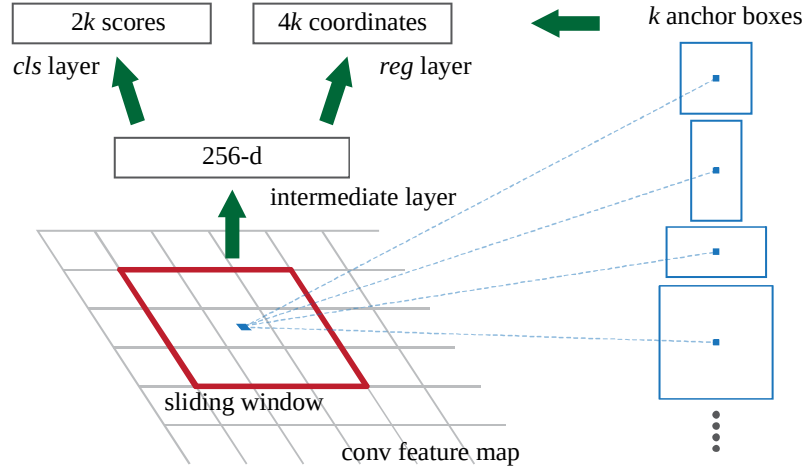


Figure 2.1: Figure from [50] which shows the correspondence between a spatial feature map and its anchors. Each feature map position and its anchors corresponds to the matching spatial position in the original image.

based on a fixed template and scale [48], or by regressing box predictions with respect to deformable parts [49].

After the release of AlexNet [50] in 2012, deep learning models for object detection surpassed handcrafted feature methods starting with R-CNN [50] in 2014. The key innovation was the use of convolutional neural networks to learn hierarchies of feature templates. In the remainder of this subsection, we discuss the two main types of spatially-referenced models: anchor-based models in Sec. 2.1.1.1, and anchor-free models in Sec. 2.1.1.2.

2.1.1.1 Anchor-Based Models

While the original R-CNN and Fast R-CNN [51] used the selective search algorithm [52] to compute object proposals, Faster R-CNN [50] removed the need for this step with the introduction of the Region-Proposal Network (RPN). The RPN was the first anchor-based model of object detection, which utilize the concept of explicitly-defined bounding box anchors corresponding to positions in the feature map, visualized in Fig. 2.1. These anchors can have varying scales and aspect ratios, which are hyperparameters of the model. The Mask R-CNN model [53] extends Faster R-CNN by introducing the RoI Align operation, and implementing prediction of instance segmentation masks for each predicted bounding box. RoI Align allows for precise extraction of features corresponding to sub-feature spatial locations using bilinear interpolation. In the context of *person search*, many models make use of RoI Align for extracting features for object detection and re-id embeddings.

Other anchor-based models include SSD [54], Cascade R-CNN [19], and RetinaNet [55]. The Single Shot MultiBox Detector (SSD) was one of the first single-stage object detectors to achieve competitive performance, and introduced the concept of learning boxes with

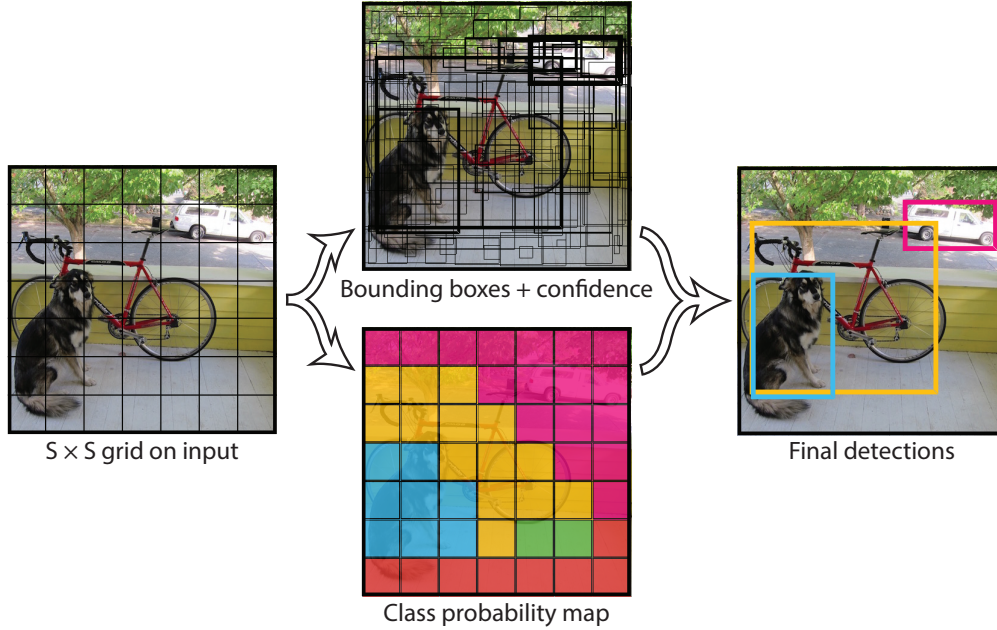


Figure 2.2: Figure from [56] which shows grid cells for the original YOLO model, and corresponding box predictions.

multiple feature map scales. Cascade R-CNN extended the original Faster R-CNN by adding additional box refinement stages with RoI pooling of features from the previous stage. The Feature Pyramid Network (FPN) introduced in [17], provides cross-scale information exchange between feature maps, significantly improving detector performance, but notably introducing other issues for person search, discussed later. RetinaNet uses the FPN, combined with the novel Focal Loss, to handle easy negatives better in a single-stage model, where negatives far outnumber positives.

2.1.1.2 Anchor-Free Models

Anchor-free models still have spatially-referenced features, but do not utilize anchors. The term usually refers to models which make predictions directly corresponding to spatial feature maps (a rectangular grid of features/predictions), unlike for example the original R-CNN, which generates proposals from anywhere in the image using Selective Search. In the YOLO model [56] (shown Fig. 2.2), detected box centers are predicted with respect to the top left corner of the matching grid cell, with object width and height predicted with respect to the bounds of the full image.

In FCOS [18], each feature position in the output feature map(s) predicts whether it is contained within any bounding box. For bounding box regression, a feature matches to the smallest bounding box it is contained within, and predicts the distance to each of the four sides of that box from the feature position. A centerness score is regressed for each position

to help deduplicate boxes by downweighting scores corresponding to boxes which are less “central”.

In CornerNet [57], objects are detected as a pair of keypoints representing the top-left and bottom-right box corners. Separate heatmaps predict top-left and bottom-right corners, and learned embeddings corresponding to the corner predictions are pushed close together if corners correspond to the same object, which allows predicting which corners pair together to form objects. This is based on the idea of associative embeddings from [58]. In CenterNet [59], CornerNet is extended by additionally predicting object centroids, which improves both precision and recall.

In “Bridging the Gap Between Anchor-based and Anchor-free Detection...” [60], the authors observe that the main difference between anchor-based and anchor-free detectors is actually in how positive and negative training samples are defined. When the same definition is used for both model types, they observe that there is no significant difference in performance, which they tested by comparing RetinaNet and FCOS.

2.1.2 Set Prediction Models

Set prediction object detection architectures predict a set of objects which is decoupled from specific positions in the backbone feature map(s). They typically use a set matching loss to match object candidates against ground truth candidates, which solves a LSAP to determine an optimal matching, based on which proposals already overlap with boxes, and how confident these proposals are. This set matching is required in absence of a direct correspondence between a feature map and object positions, which normally facilitates the matching process. Set prediction models also typically utilize learnable queries or proposals, which can exchange information, reducing proposal overlap / redundancy and eliminating the need for NMS. While other detector models, such as the original R-CNN, may decouple proposals from backbone feature positions by getting a set of object proposals from another source (Selective Search for R-CNN), they do not meet the other criteria and so are not typically called set prediction models. In the remainder of this subsection, we discuss two prominent set prediction models: DETR in Sec. 2.1.2.1, and Sparse R-CNN in Sec. 2.1.2.2.

2.1.2.1 DETR

The Detection Transformer model (DETR) [21], shown Fig. 2.3), was the first successful set prediction detector which was truly “end-to-end”. This means that the detector does not use a stop gradient operation like Faster R-CNN, but also that it does not use handcrafted mechanisms like NMS to deduplicate proposals. DETR decouples the image feature map from the bounding box prediction by instead predicting boxes with learnable query vectors, which only receive input from image features via cross attention with a transformer decoder [61]. Query vectors are matched to ground truth boxes via direct set prediction based on an optimal set matching which minimizes a combined classification and box overlap loss, using the Kuhn-Munkres algorithm [62], [63], also called Hungarian matching.

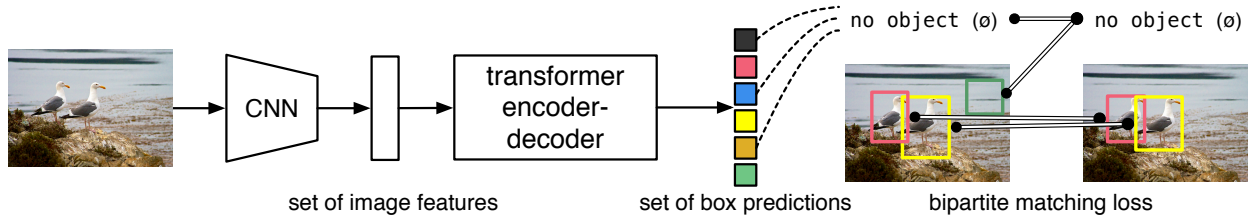


Figure 2.3: Figure from [21] which shows the DETR object detection model. DETR performs a one-to-one matching between object queries and ground truth objects using set matching, instead of associating objects with spatial feature map positions.

While the original DETR was able to achieve comparable performance to other leading detectors when it came out, it took many more training epochs to reach that level of performance. While Faster R-CNN and RetinaNet could fully train on MS COCO 2017 [64] in 36 epochs, DETR requires 500 epochs to fully train. Therefore, it is clear that one or more of the model inductive biases of DETR reduces its optimization speed. The single most important factor appears to be the concept of spatially-referenced features, which later variants of DETR added back to the model in different ways.

Deformable DETR [65] uses a deformable attention mechanism which attends to a small set of key sampling points around a dynamically learned reference point, instead of the entire image feature map. Efficient DETR

All of these methods can fully train on COCO 2017 in $10\times$ fewer epochs than the original DETR.

2.1.2.2 Sparse R-CNN

Sparse R-CNN [66], shown in Fig. 2.4, is similar to DETR in the sense that it is a set prediction model which uses queries, but Sparse R-CNN eliminates the need for the expensive cross-attention between image features and object queries by using RoI Align. Sparse R-CNN uses learnable bounding boxes in place of the learnable query embeddings used by DETR. Notably, Sparse R-CNN optimizes much faster than DETR, and is more similar in character to faster R-CNN, in that it uses spatially-referenced features extracted using RoI Align. The difference is that Sparse R-CNN has the potential to be faster by avoiding per-feature box proposals, and that it does not require NMS since the entire prediction pipeline is sparse.

2.1.3 Sequence Prediction Models

Sequence prediction models of detection, first proposed in Pix2Seq [23], shown in Fig. 2.5, output boxes as a sequential prediction problem, using an autoregressive transformer decoder. Pixel coordinates correspond to tokens which are predicted, 4 coordinate tokens per box in addition to a class token. Pix2Seq has a fundamentally high inference cost because of this

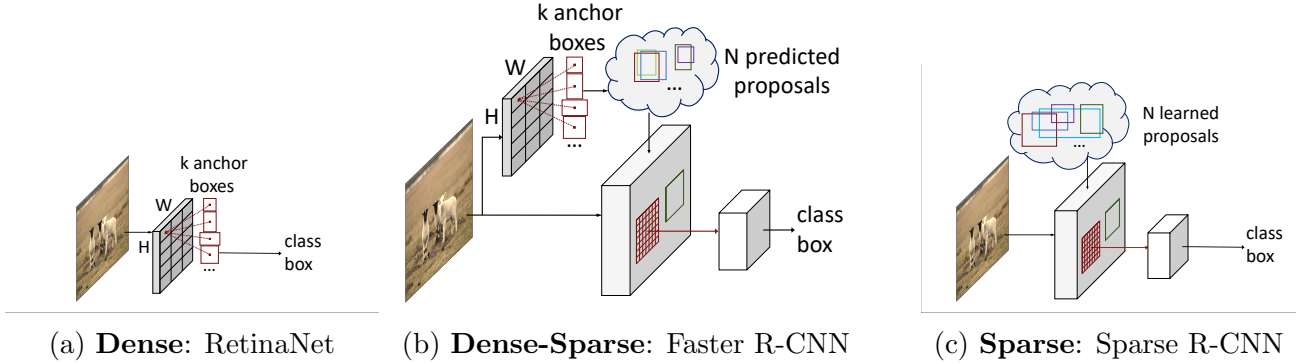


Figure 2.4: Figure from [66] comparing dense prediction as in RetinaNet, to sparse prediction as in Sparse R-CNN, with Faster R-CNN in the middle which uses a dense RPN prediction followed by a sparse refined box prediction.

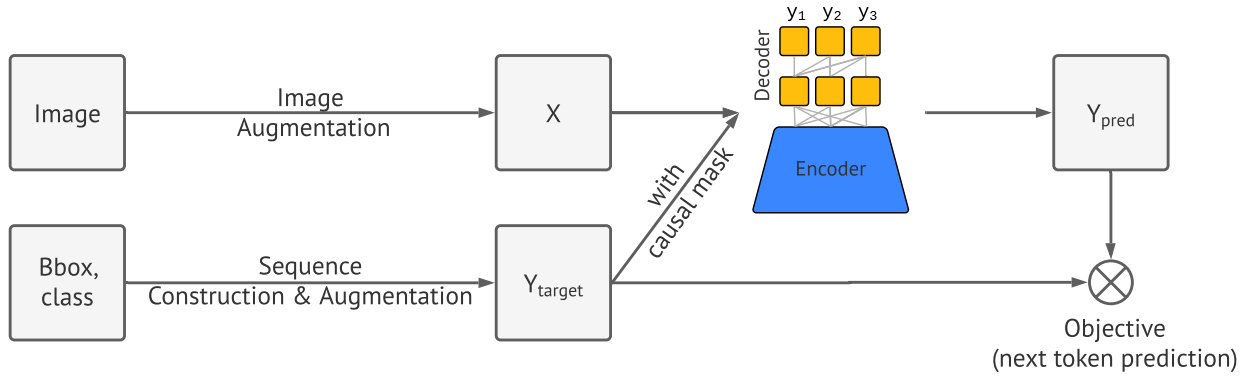


Figure 2.5: Figure from [23] shows Pix2Seq architecture, which predicts boxes and classes as a sequence of tokens. Architecture is nearly-equivalent to DETR, except that objects are predicted using autoregressive decoding, instead of using a fixed set of object queries.

sequential decoding, but has perhaps the greatest potential as a generalist approach to object detection. This is because the token vocabulary of Pix2Seq can be easily integrated into models which can solve many other problems, [67]–[70].

2.1.4 Bounding Box Regression

There are two standard methods of bounding box regression for modern object detection models: 1) regressing bounding box offsets with respect to a spatial reference, and 2) direct regression of bounding box coordinates.

For anchor-based and anchor-free models, bounding box reference coordinates are determined by the position of anchors or feature centers. It is typical to convert ground truth

boxes and predicted boxes to the format (x_c, y_c, w, h) , where x_c, y_c denote the box center coordinates, and w, h are the box width and height. The actual model output targets / regression targets are a function of the target box and reference box parameters. It is typical to use $t_x = \frac{\hat{x} - x}{w}$, $t_y = \frac{\hat{y} - y}{h}$, $t_w = \log(\frac{\hat{w}}{w})$, $t_h = \log(\frac{\hat{h}}{h})$, where parameters with hats represent the ground truth box or predicted box, and parameters without hats represent the reference. Then, the prediction can be decoded at inference time using the model output combined with the reference box parameters.

For bounding box offset regression, it is typical to use mean-squared error loss, ℓ_1 loss, or most-commonly smooth ℓ_1 loss.

For direct regression of bounding box coordinates, the Generalized Intersection over Union (GIoU) loss [71] is a popular choice of objective. GIoU extends the well-known Intersection over Union (IoU) measure to cases where boxes do not overlap; in short, the GIoU is higher the closer two boxes are together, whether or not they overlap.

We present the definitions of IoU and GIoU given in [71]: Given two arbitrary convex shapes $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$. For A and B , define the smallest enclosing convex object as $C \subseteq \mathbb{S} \in \mathbb{R}^n$.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (2.1)$$

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \quad (2.2)$$

The GIoU loss then is simply defined as $\mathcal{L}_{GIoU} = 1 - GIoU$. It is a reasonable choice of loss for direct regression because it directly encodes the quantity we want to minimize, and is continuous and differentiable when boxes don't overlap, unlike standard IoU.

2.2 Image Retrieval

Image retrieval is the other core problem of localized image retrieval, where the goal is to return a list of images ranked by similarity to a query. It is typically solved using metric learning, discussed in Sec. 2.2.1. When image retrieval is done at the identity level, for example by retrieving different images corresponding to the same person, it is called instance-level recognition or identity-level recognition, which is discussed in Sec. 2.2.2.

2.2.1 Metric Learning

The goal of metric learning is to learn embeddings of data points $x \in \mathbb{R}^d$ with some geometric relationship that corresponds to known relationships between the data. We show this for a simple example of $x \in \mathbb{R}^2$ in Fig. 2.6.

By minimizing some objective function, we push points with the same label closer together, and points with different labels further apart, ideally forming separable clusters. One of the first metric learning losses employed for representation learning was the contrastive loss [72],

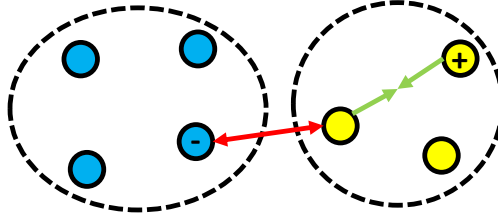


Figure 2.6: Data embeddings from two classes, shown by blue and yellow labels. In metric learning with a contrastive objective, the goal is to push each point towards points with the same label (green arrow), and away from points with a different label (red arrow).

shown in Eq. (2.3). For a pair of points (x_1, x_2) , we define $D = \|x_1 - x_2\|_2$, and $y = 1$ if x_1, x_2 share a label, else $y = 0$. The margin parameter m determines how far apart clusters will be.

$$\ell = \frac{1}{2}y \cdot D^2 + \frac{1}{2}(1 - y) \cdot \max(0, m - D)^2 \quad (2.3)$$

However, the most common loss currently used for metric learning is the normalized temperature-scaled cross-entropy (NTX) loss [73], also called the InfoNCE loss [74], which is a generalization of the multi-class N-pair loss [75] to include a temperature parameter. This loss is commonly thought of as the default “contrastive loss”, despite the earlier contrastive loss from [72]. We show a version in Eq. (2.4), for positive pair (x, x^+) , and set of negative pairs $(x, x^-) \in X^-$. The temperature τ determines how diffuse clusters are (higher temperature, more diffuse).

$$\ell = -\log \frac{\exp(\text{sim}(x, x^+)/\tau)}{\exp(\text{sim}(x, x^+)/\tau) + \sum_{x^- \in X^-} \exp(\text{sim}(x, x^-)/\tau)} \quad (2.4)$$

Other common metric learning objectives include variants of the NTX loss SphereFace [76], CosFace [77], and ArcFace [78], and the triplet margin loss [79], all developed for the face re-id problem.

2.2.2 Instance-Level Recognition

Instance-level or identity-level recognition is the problem of recognizing a different instance of a known object at the identity level. This is separate from class-level recognition, because there can be many instances of the same class. In addition, instance-level recognition is usually approached as an image retrieval problem, in which we aim to return the top matching results ranked in order of similarity to a query image. This is also known as the re-identification or re-id problem. The objects with the greatest practical interest are people and vehicles, and many datasets have been created which annotate images of people [80]–[82] or vehicles [83]–[85] at the identity level.

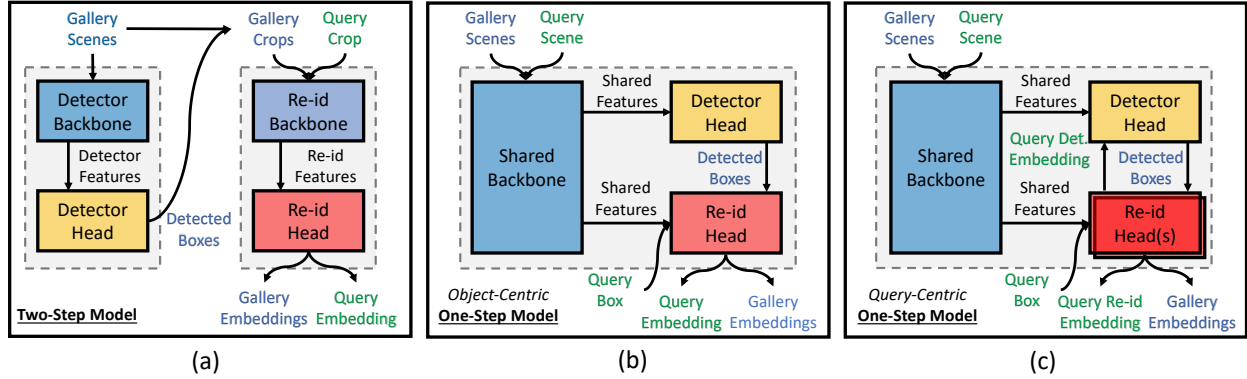


Figure 2.7: Diagrams comparing the two-step person search model (a) with object-centric (b) and query-centric (c) one-step counterparts. Input/output labels are colored green for query elements and blue for gallery elements. All models share in common presence of backbones, detector heads, and re-id heads, arranged in different configurations.

It is typical to approach the re-id problem using only rectangular image crops of the objects of interest, which is what distinguishes this problem from object search / identity search. Numerous methods have been developed for person re-id [86]–[99], which focus on handling low resolution images [92], crowded scenes causing person overlap and occlusion [91], [98], multiple image modalities [93], and domain shift [90], [94], [95], [97].

2.3 Person Search

In the *person search* problem, a *query* person image crop is used to localize co-occurrences in a set of scene images, known as a *gallery*. Beginning with the release of two benchmark person search datasets, PRW [100] and CUHK-SYSU [1], there has been continual development of new deep learning models for person search. Most methods utilize the Online Instance Matching (OIM) Loss from [1] for the re-id feature learning objective. The breadth and depth of research on the topic of person search reveal a network of interlinked challenges which are broadly applicable to the problem of localized image retrieval. In the following paragraphs, we will detail some models and methods of person search, before moving on to a discussion of challenges in the person search problem.

One-Step vs. Two-Step Models. *Person search* is typically divided into two subtasks: *detection* (*localization+classification*) of all person bounding boxes in a set of gallery scenes, and *re-identification* (*re-id*) of all detected boxes with respect to a query person box. This can be accomplished with a separate model for each subtask, referred to as *two-step* [100]–[103], or using a single model, referred to as *one-step* or *end-to-end* [1], [3], [104]–[116]. A figure distinguishing one-step and two-step models is shown in Fig. 2.7. In the two-step model, shown in Fig. 2.7a, boxes are detected with one model, and query/gallery image crops are fed

to a separate model to extract re-id embeddings. In the object-centric one-step model, shown in Fig. 2.7b, the detector and re-id heads share a feature backbone. In the query-centric one-step model, shown in Fig. 2.7c, detection is conditioned on the query embedding, meaning detections for a given scene are computed separately for each query. Query re-id and detector embeddings may be computed using the same or separate heads, in which case the embedding dimension can differ between the two

Two-step models offer modularity, while end-to-end models offer efficiency in the form of feature re-use, simplified training, and in many cases superior generalization. The problem of Person Search, originally identified in [2], gained popularity with the release of two datasets: *CUHK-SYSU* [1] and *Person Re-identification in the Wild* (PRW) [100], both presented at CVPR 2017.

Query-Centric vs. Object-Centric Models. Most person search models implement an *object-centric* (OC) approach [110]–[115], in which we detect persons independent of any query, then afterwards compare queries to detected persons. By contrast, *query-centric* (QC) person search models [4], [104], [117]–[121] use query person images during the detection process, producing proposals tailored to each query, but significantly increasing time complexity of inference computation. See Fig. 2.7b,c for a comparison.

The query-centric approach to person search is less common due to its greater inference time complexity, but multiple successful approaches have emerged. In [118], [119], QEEPS and QGN use query information both during detection and computation of re-id embeddings by combining query features with backbone features using squeeze-and-excitation layers [122]. IGPV [120] and TCTS [121] are two-step models which extract person embeddings from a separate network, and combine these embeddings with features from the detector network.

Weakly-Supervised Learning. In the context of person search, weak supervision refers to training on person bounding boxes without identity-level labels. Several methods [123]–[128] have emerged in recent years to tackle this problem by using weakly-supervised objectives directly on the target dataset. These methods focus on clustering visual features and applying contextual information to determine pseudo-labels used for a contrastive re-id loss. We note that it is important to explore alternatives to the contextual and clustering methods typically used to form pseudo-labels, as these can incur harmful biases, especially when applied to new pre-training datasets with unknown person distribution properties.

Pre-training. Person search models typically use backbone weights initialized from standard ImageNet-1k classifier training, while other model weights are initialized randomly. To our knowledge, the only work prior to ours to explore another backbone initialization strategy is SOLIDER [129]. The SOLIDER framework trains on unlabeled person crops from the LUPerson dataset [130] by optimizing backbone features using cluster pseudo-labels based on tokens learned with DINO [131]. We note that SOLIDER initializes only the person search model backbone, not the rest of the model, and trains only on person crops, not full scene images.

Use of Additional Data. While usage of additional data beyond the target fine-tuning dataset is standard in other domains of visual recognition, it has only recently been done for person search. In [123], person image crops from the Market dataset [80] are injected into

the contrastive loss with crops from the target dataset, significantly improving performance. SOLIDER uses additional person data from the LUPerson dataset [130] for pre-training. We note also that some models have indirectly used additional data via knowledge distillation from a trained re-id person crop model, such as [109].

Section Overview. In the subsections that follow, we broadly divide person search methods by which problems they address. These can be roughly-split into four categories, though we discuss overlap between them: 1) detector challenges in Sec. 2.3.1, 2) re-id challenges in Sec. 2.3.2, 3) the detection re-id conflict in Sec. 2.3.3, and 4) runtime efficiency in Sec. 2.3.4.

2.3.1 Detector Challenges

The first step for most person search methods is detection of persons in a given scene. The key challenges of detection are how to reduce redundant or erroneous proposals, and how to learn accurate boxes which have high overlap with the ground truth. A related issue is how to address misaligned boxes in the ground truth data itself. We discuss the issues of redundant proposals in Sec. 2.3.1.1, detected box misalignment in Sec. 2.3.1.2, and ground truth box misalignment in Sec. 2.3.1.3.

2.3.1.1 Redundant Proposals

Methods including PSTR [113], LEAPS [114], and PSDiff [132] seek to reduce the number of object proposals via a sparse proposal architectural constraint. The PSTR model is based on DETR [21], LEAPS is based on Sparse R-CNN [66], and PSDiff is based on DiffusionDet [133]. These architectures eliminate the need for non-maximum suppression (NMS) [134] by incorporating selection of objects directly into the learning process. Information exchange between proposal features allows these models to learn proposal deconfliction, which is not possible in dense detectors, where proposals do not have knowledge of each other.

Query-centric approaches to person search [4], [104], [117]–[121] eliminate the need for NMS by linking the query to the detection process. While this ensures that there is at most one match per scene to a given query, it significantly increases the cost of detection, by requiring a separate detection computation for each query-gallery pair. This cost is discussed in Chapter 4.

2.3.1.2 Detected Box Misalignment

The detected box misalignment challenge is the challenge of how well detected boxes overlap with the ground truth, and handling resulting implications for re-id. Person search methods typically address this either by learning more accurate boxes, learning boxes which are tailored for re-id, or making re-id learning itself resilient to misaligned boxes.

In RDLR [103], proposals are refined under the supervision of the re-id loss using a differentiable RoI transform layer which allows gradients to be backpropagated from the re-id loss to bounding box regression. In NAE [108], a mask is learned to match only the portion

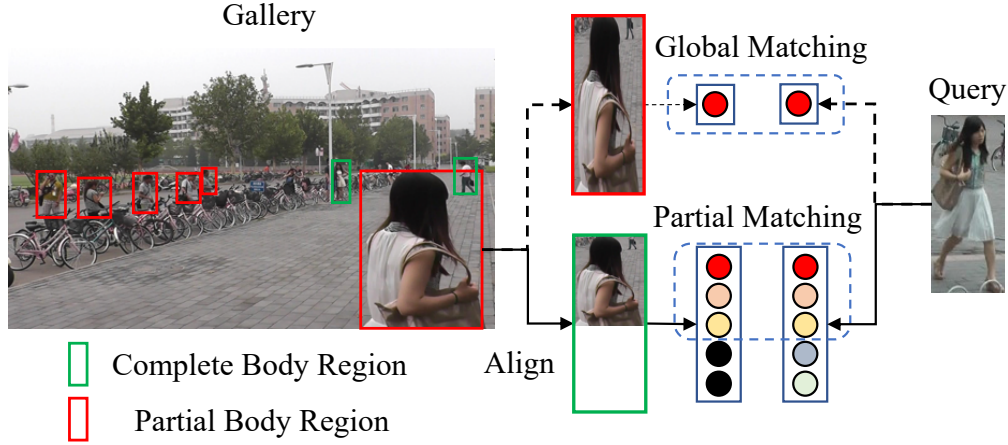


Figure 2.8: Figure from [105] which demonstrates partial matching problem and feature alignment solution. By not forcing correspondence between misaligned embeddings, the partial matching method improves re-id results.

of the predicted box which overlaps the corresponding ground truth. This mask is applied to re-id features so that only features from the matching person are used to compute the re-id embedding. In PGSFL [135], a prototype-guided attention module is used to learn which parts of the proposal image crop are relevant for re-id, to indirectly address misalignment. For instance, it learns to attend to consistent regions across images from the same identity *i.e.*, face, upper body, and lower body. In DKD [109], intentionally-misaligned crops are used by the re-id module to learn robustness to misaligned crops from the detector.

2.3.1.3 Ground Truth Box Misalignment

Methods such as [105], shown in Fig. 2.8, can handle detector misalignment, but also the case when the ground truth boxes themselves are misaligned or cropped by the scene extent. This model aligns box features by first performing human keypoint recognition, then dividing persons into horizontal part features, then matching only those part features which have keypoint correspondence between two crops by depressing the response of features in noisy or invalid regions. While use of horizontal part features as in [105], [136] assumes upright person posture, this assumption often holds in typical person image datasets. However, it is worth noting that this will not apply to persons in different positions, *e.g.* laying down, and could present an attack vector for adversarial methods.

2.3.2 Re-id Challenges

After detecting bounding boxes, we perform re-id by extracting embeddings corresponding to detected boxes. Embeddings can be extracted either by putting detected person image crops through a separate network, or by pooling backbone features shared with the detector.

Challenges of person re-id include issues with the data itself such as pose/viewpoint variation, scale variation, occlusions, lighting conditions, and changes in appearance such as clothing, with examples shown in Fig. 2.9. We discuss these data issues in Sec. 2.3.2.1. There are also architectural challenges more relevant to person search specifically, such as how to handle multiscale features, and optimization challenges such as how to handle small batch size. We discuss these optimization challenges in Sec. 2.3.2.2. Finally, there are issues with how to handle context, including both local and global context of each person in the scene, which we discuss in Sec. 2.3.2.3.

2.3.2.1 Data Challenges

Challenges with variation in appearance due to pose/viewpoint variation, lighting conditions, and other appearance changes such as clothing are typically handled by focusing on the representation learning objective. In the case of person search, most models use the original Online Instance Matching (OIM) loss from [1]. Several methods enhance this objective using variations of a triplet loss [79], including the Triplet-aided OIM Loss from [110], OIM Triphard Loss from [115], and triplet-wise knowledge distillation from [109]. Other variations include the Symmetric OIM (SOIM) loss for handling hard positives, and the Hierarchical OIM (HOIM) loss for integrating recognition of background samples into the re-id objective. Other re-id objectives include the Center loss from [137], which improves intra-class compactness, and the Circle loss from [138], which is constructed for the case of multiple positive pairs for a given identity.

Occlusion is explicitly addressed in some methods, including [105], [112], [139]. In [105], as previously mentioned, partial matching of corresponding horizontal person features is used to address occlusions. In [139], spatial attention to re-id features addresses occlusions by focusing on regions beneficial to re-id. In [112], the Cascade Occluded Attention Transformer (COAT) is designed to specifically handle occlusions using an occluded attention mechanism which simulates realistic occlusions within a set of visual tokens.

Handling of multiscale features is tightly-coupled with detector performance, and is a particular challenge for re-id in person search. The multiscale feature issue is that features coming from different levels in a feature pyramid, typically different scale layers in a CNN or Feature Pyramid Network (FPN), are not perfectly aligned. Therefore, when person boxes corresponding to the same identity are assigned to different levels in the feature pyramid, their features are not aligned. While using multiscale features in this way for object detection with a Feature Pyramid Network tends to benefit performance, the same is not the case for person re-id. One strategy to handle this is to simply extract all detector features from only one layer in the FPN, *e.g.* P_3 as in [110]. This was found to harm detector performance, but overall person search performance was significantly improved. Another strategy is to introduce more connections between layers in the FPN, as in [115], which adds a bottom-up FPN connection, in addition to standard top-down, and an attention module which captures long-distance correlation between feature maps.

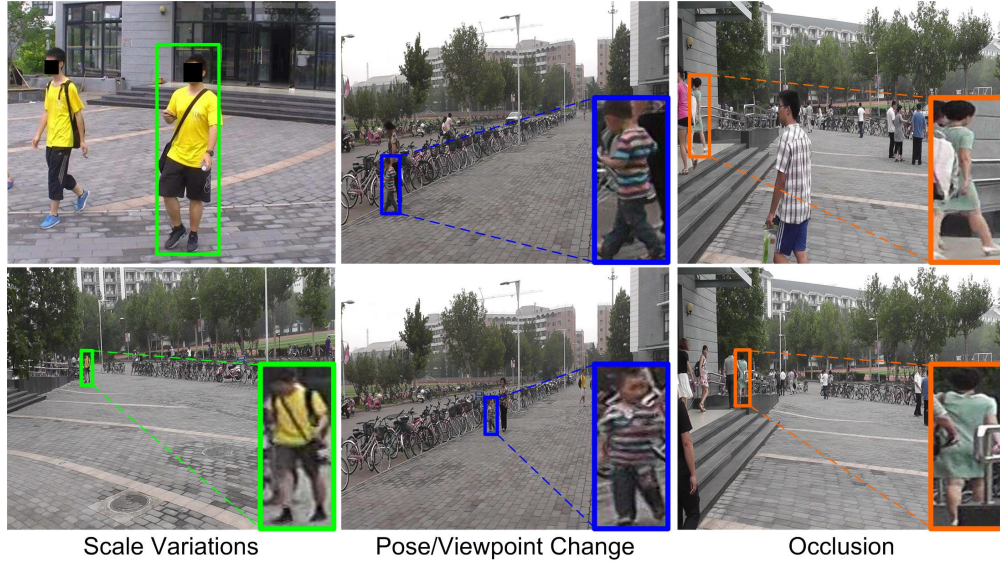


Figure 2.9: Figure from [112] which shows typical examples of scale variation, pose/viewpoint change, and occlusions.

2.3.2.2 Optimization Challenges

Another challenge for re-id in person search is that image samples are typically much larger, and computation per sample is greater. This is because we utilize full scene images instead of just person crops. The result is that we must either reduce batch size, or reduce image size, to lower memory usage for a given model. If we reduce batch size, sample diversity per batch is also reduced, and the quality of information in the gradient update is lower. The OIM loss addresses this challenge by storing a lookup table of exponential moving average prototype embeddings corresponding to each known person identity, in addition to a queue of recent embeddings for unknown identities. However, the embeddings in the lookup table and queue can quickly become out of sync with the current network state if the gradient updates are large. Therefore, it is logical to use a MOCO-style approach, as in [138]. This ensures that embedding comparisons are consistent and up to date by using an exponential moving average network to generate queue embeddings for comparison against embeddings from the main network on the current batch.

Another strategy is to reduce image size, which allows one to increase batch size for a given memory budget. The typical image augmentation strategy for person search methods is to use a “window resize” image resizing strategy, which resizes images to fit in a horizontal or vertical window of min side length 900 and max side length 1500 pixels during both training and testing.

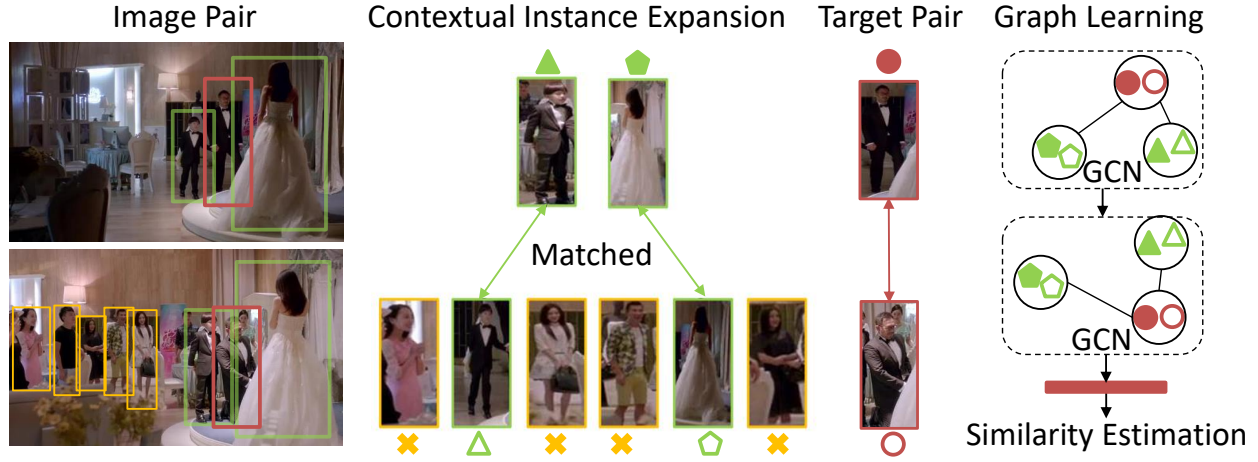


Figure 2.10: Figure from [136] which shows the process of learning a context graph for person search. Each node comprises a match between a query and gallery instance of a given person identity, and persons without a match on both images are not part of the graph.

2.3.2.3 Scene Context

The effect of local and global scene context on re-id bounding boxes is another key factor distinguishing person search re-id with basic image crop re-id. Exactly how to use context is a debated topic, because learning of certain contextual clues, *e.g.* groups of people traveling together, do not generalize well to all settings. For example, if we learn to score matching pairs of people higher in the query and gallery scene, the model could be less effective in cases when the people are alone, if we are not careful.

For global scene context in particular, methods which utilize information about groups of people include [136] and [111]. In [136], a Graph Convolutional Network (GFN) [140] is used to model similarity of probe-gallery identity pairs, which are treated as graph nodes (shown in Fig. 2.10). Similarly in [111], Context Bipartite Graph Matching (CBGM) is used during model inference by using the Kuhn-Munkres algorithm [62], [63] for bipartite matching between all person boxes by cosine similarity of embeddings in the query and gallery scenes, and keeping only the top matching box for the query in the gallery, if there is one.

Some methods also consider context around re-id boxes as a source of potentially unwanted features. Even when features from re-id boxes are extracted using RoI Align, the large receptive field of these features can capture information that is irrelevant or harmful to generalization. Some methods make use of knowledge distillation / student-teacher networks [141], [142] to distill knowledge from a separate re-id network to the main network. In [109], the re-id network is trained only on person crops, and re-id embeddings from the person search network are pushed towards embeddings from the re-id network by minimizing KL Divergence between them, shown in Fig. 2.11. A similar mechanism is used in [143] at multiple points in the model to enforce feature correspondence with a crop-only teacher network.

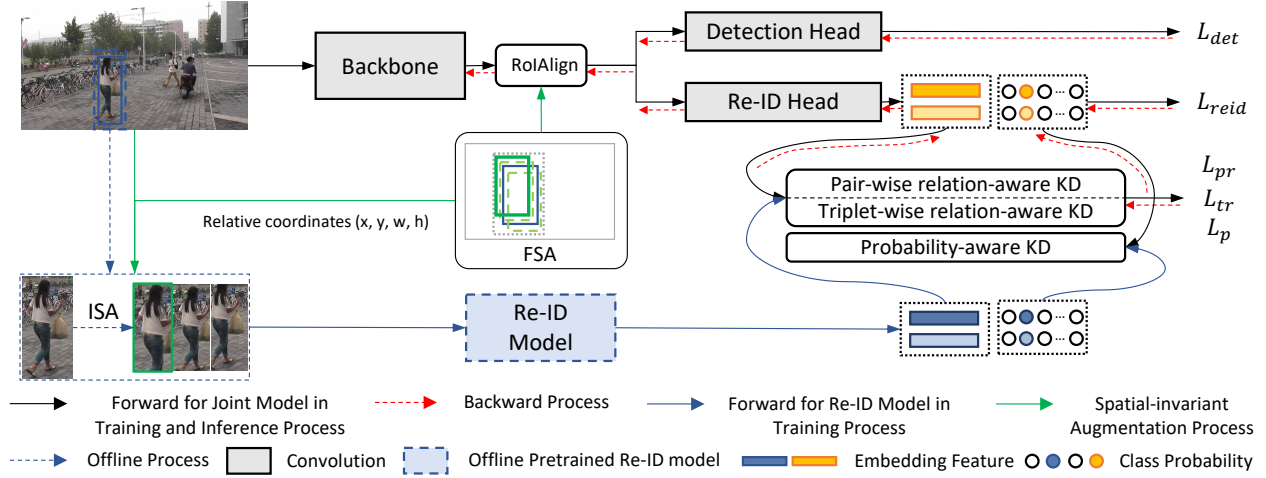


Figure 2.11: Figure from [109] which shows knowledge distillation of the main person search network re-id embeddings using a separate crop-only re-id embedding network.

In [107], the person search network backbone itself is used to extract person crop features, by feeding the resized image crops directly into the backbone. In this case, when RoI Align embeddings and person crop network embeddings are pushed together with KL Divergence, gradients flow through the shared backbone, unlike in [109]. This has the effect of removing context in re-id features, without requiring knowledge distillation using a separate re-id network.

In [101], the authors go further still by masking out pixels within image crops themselves that do not correspond to the target person. They feed both masked and unmasked person crops to the separate re-id network, allowing it to use information from both where beneficial, shown in Fig. 2.12. This type of strong inductive bias can make optimization faster, as it can be difficult for the model to suppress irrelevant pixels within image crops on its own, given the large receptive field size of re-id features.

2.3.3 Detection Re-id Conflict

A more subtle challenge of person search, and localized image retrieval more broadly, is how to handle what is often described as an inherent conflict between the tasks of person detection and re-id: person detection is about learning commonness for different instances of the person category, and re-id is about learning uniqueness between person identities. More generally, this is an example of a problem in representation learning about how to handle hierarchical object categories, and there are several methods designed to address it including box embeddings [144], [145], and Poincaré embeddings [146]. However, in the context of person search, methods are focused on the specific conflict between person detection and person re-id, and therefore employ more targeted approaches that do not attempt to solve

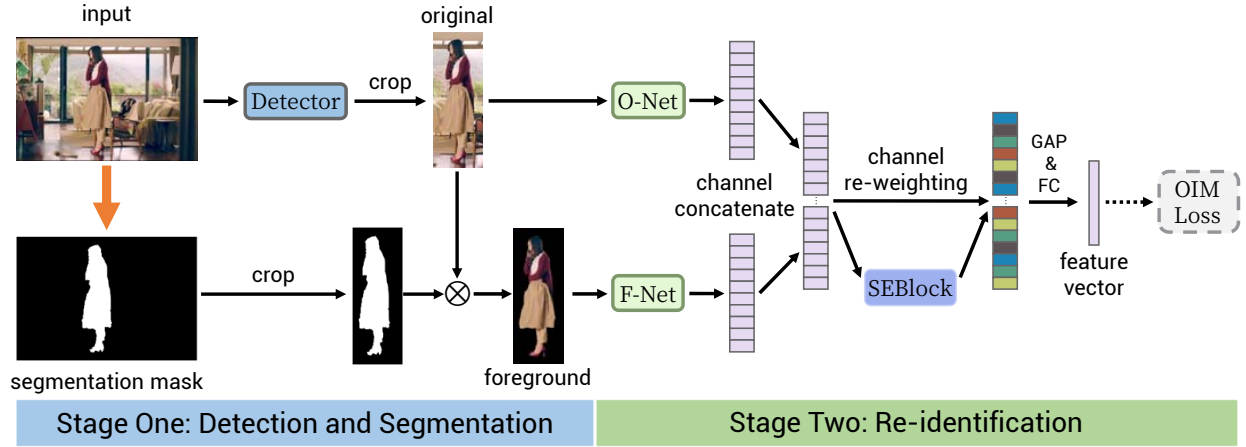


Figure 2.12: Figure from [101] which shows masking of person crops, with masked and unmasked person crop features fed to a separate re-id network.

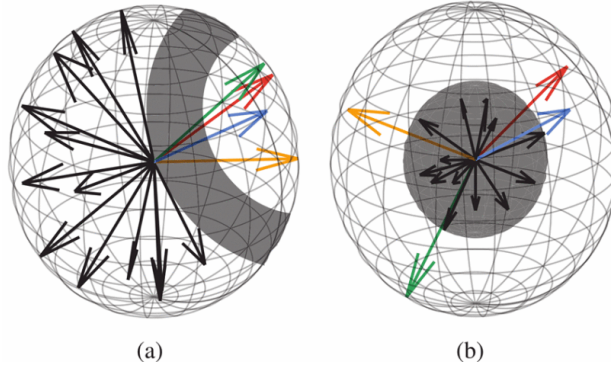


Figure 2.13: Figure from [108] shows the unconstrained embedding space (a), where person embeddings are confined to a smaller region by background embeddings, and the norm-aware embedding space (b), which uses polar decomposition of embeddings into classification and re-id to reduce this issue.

the general hierarchical representation problem.

In [108], the Norm-Aware Embedding (NAE) method attempts to explicitly decouple detection and re-id by using the polar decomposition of a person embedding. The norm of the embedding is treated as the classification score (person vs. background), and the angle of the embedding is used for re-id via cosine similarity in the OIM loss. By decoupling background from persons in the embedding space, NAE prevents the angles of person embeddings from being harmfully-confined to a smaller region of the embedding space, visualized in Fig. 2.13.

Other methods attempt to integrate detection and re-id by directly using background embeddings as part of the re-id loss. In [106], the Hierarchical OIM (HOIM) loss adds an

extra embedding queue of only background embeddings to the standard OIM loss, which they claim makes the model more robust to false alarms and misalignment, while increasing feature diversity.

In SeqNet [111], the detection re-id conflict is addressed by creating more capacity and separation in the model between the two tasks. Specifically, the model is a Cascade R-CNN variant [19] which performs detection in multiple stages, and extracts re-id embeddings from a separate head using only boxes output by the last stage. The extra capacity in this model for detection and re-id heads, in addition to the separation between the two tasks, result in strong performance, and this model design is utilized in several later models [3], [147], [148]. In [147], CNN modules in SeqNet are replaced with transformers, and in [148], fused CNN and ViT modules are used, in combination with NAE from [108].

While two-step models explicitly avoid the detection re-id conflict by training separate models for each task, Task Incremental Learning (TIL) [149] approaches use a one-step model, but train the two tasks sequentially. In [150], the detector module is trained first, then its weights are frozen before training the rest of the model for the re-id task. This has the benefit of preserving detector performance, but the potential to reduce re-id performance by preventing gradient updates to non-detector weights.

Another approach is to influence gradient updates to re-id based on detector results. In [151], adaptive gradient weighting channels gradients through re-id and part classification networks, weighting them based on the quality of the detection.

2.3.4 Runtime Efficiency

Finally, there are several methods which endeavor to reduce training and/or inference time as a goal in itself, while preserving person search re-id metrics. This is cited as one reason for using one-step rather than two-step person search models. Two-step models may take longer to train, as they require training of each step independently, and they may be less efficient during inference, since they do not share features, increasing total computation per scene. We further consider specifically how one-step vs. two-step models affect model efficiency in Chapter 3. Likewise, object-centric models are more efficient than query-centric models during inference, since they only need to compute detector proposals once per scene, rather than once per query per scene.

Some models make architecture choices which are centered around reducing the cost of object detection. In AlignPS [110], the choice of an anchor-free model is primarily to improve efficiency over two-stage or multi-stage anchor-based detectors. Other methods make use of the sparse detection paradigm [114], [132] to reduce detector computation. In our work [3], the Gallery Filter Network, discussed in Chapter 3, removes the need for detection in some scenes altogether by filtering scenes via comparison of query and scene embeddings.

2.4 Unsupervised Learning

Unsupervised learning methods are a popular choice for model pre-training, because they allow us to exploit large quantities of unlabeled data, vs. less abundant labeled data for the target task. We introduce self-supervised learning methods in Sec. 2.4.1, discuss unsupervised person re-identification in Sec. 2.4.2, and cover self-supervised learning specifically for localization in Sec. 2.4.3.

2.4.1 Self-Supervised Learning

Self-supervised learning is a variant of unsupervised learning where we solve a pretext task on unlabeled data. It is common to use the NTX loss or InfoNCE loss as we previously defined it, combined with methods including SimCLR [73], [152], CPC [74], [153], or MoCo [154], [155]. In SimCLR, we augment the same image twice, using transforms like cropping, flipping, and rotation, then teach the model that augmentations of the same image are more similar to each other than to other images. Other approaches involve predicting cluster assignments [156], inpainting masked image regions [157], and predicting the “next” or adjacent feature of an image given the current and previous features [158], in addition to many others [131], [159]–[161].

2.4.2 Unsupervised Person Re-identification

For unsupervised person re-id specifically, most approaches involve assigning pseudo-labels to image crops [162]–[173] or crop part features [129]. The goal is to either get pseudo-labels as close to real identity labels as possible, or at least to group similar instances in such a way that they could correspond to different images of the same identity.

2.4.3 Self-Supervised Learning for Localization

Self-supervised learning for object localization [174]–[177] is a less explored area, since it is challenging to formulate a pretext task for localization which leads to downstream generalization. In UP-DETR [178], the original DETR is pre-trained using the *random query patch detection* task, which involves combining random patch features in a given image with the object queries, then regression queries to localize those patches. This was found to significantly speed up DETR fine-tuning for supervised object detection. We can think of this process as query-centric detection pre-training, followed by object-centric fine-tuning without layer modifications.

In DETReg [179], DETR is pre-trained using pseudo-object proposals from the selective search algorithm. This initializes the model using a reasonable distribution of object-like boxes, and also speeds up downstream optimization. While this approach has the potential to transfer well if the selective search object distribution resembles that of the downstream task,

that may be unlikely in practice, and the model will have to “unlearn” bad pseudo-boxes during fine-tuning.

Other approaches such as ReSim [180] aim to improve feature consistency between corresponding image sub-regions of two augmentations of the same image.

2.5 Model Fine-Tuning

Fine-tuning models which have already been trained for other tasks is a critical topic in modern visual recognition, as foundation models and other large vision models provide valuable weight initializations for many downstream recognition tasks. We discuss linear probing in Sec. 2.5.1, Learning without Forgetting in Sec. 2.5.2, and LoRA in Sec. 2.5.3.

2.5.1 Linear Probing

While linear probing [181] was initially used to analyze features at different layers in a model for how well they perform at discriminative tasks such as image classification, it has been used more recently as a step in the model fine-tuning process. In [182], the authors suggest that first linear probing, then full fine-tuning can improve fine-tuning results by preventing distortion of pre-trained features.

2.5.2 Learning without Forgetting

Learning without forgetting [183] is a mechanism for utilizing pre-trained model outputs to retain high quality features from pre-trained models, while updating weights for a new task. The loss function is equivalent to the Knowledge Distillation loss [141], and minimizes the difference between old model outputs on new task data, with new model outputs on new task data. It is shown below in Eq. (2.5):

$$\mathcal{L}_{\text{LWF}}(y, \hat{y}) = - \sum_{i=1}^l y'_i \log \hat{y}'_i, \quad y'_i = \frac{y_i^{1/T}}{\sum_j y_j^{1/T}} \quad (2.5)$$

with l unique labels, and class logits $y, \hat{y}$. y corresponds to outputs of the original classifier model, and $\hat{y}$ to outputs of the model currently being optimized. The temperature value is set to $T = 2$ by default.

The idea is that by keeping the models output on the old task similar to what it was before fine-tuning, we can retain most of the useful features needed to solve that task, while updating weights for the new task. More recently, learning without forgetting has been used for Task Incremental Learning, to train for new tasks while retaining old task performance [184].

2.5.3 LoRA

Low-Rank Adaptation of Large Language Models (LoRA) [185] is an efficient mechanism of updating large model weights, without needing to pass gradients through all model weights. The idea is that we learn a low-rank matrix which is added to existing model weight matrices at different layers, then simply combine the matrix with old model weights to use it at inference time. The low-rank matrix can be efficiently learned using a low-rank decomposition, which has many fewer weights than the original model weight matrices. For pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, we can represent the weight update as $W_0 + \Delta W = W_0 + BA$, with $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $\text{rank } r \ll \min(d, k)$.

2.6 Foreword on Efficiency

In the deep learning literature, recent empirical studies have demonstrated performance scaling with data volume and model size in both NLP [186], [187] and vision [188], [189]. However, the quantity of available data, and the model size which could benefit from it vastly exceed available compute and budgets in many situations. Therefore, a key research question in the current moment is not how to use the largest model and dataset possible, but how instead how to use a fixed compute budget or data curation budget as efficiently as possible.

In the pursuit of efficiency when training large models, it is also critical to consider how to best use existing pre-trained model weights. In most environments, it is not possible to replicate training of large models like GPT-4 [190], DALL-E-3 [191], or even CLIP [33], yet it is possible to utilize components of these models to extract high-quality features, without needing to update the model weights.

Therefore in the context of efficient LIR models, we must consider how to most effectively use feature extractors and other pre-trained weights when developing new methods. For example, it may be interesting to consider pre-training of image backbones using self-supervised methods. But in practice, we must consider existing image backbone weights, and whether we should be training new backbones from scratch, or utilizing existing backbone weights for feature extraction, fine-tuning, Learning without Forgetting [183], or LoRA [185].

Chapter 3

Modular Efficiency

In the context of this work, modular efficiency refers to the concept of subdividing LIR into two or more tasks with separate models, and training and/or performing inference with these models separately to improve efficiency. In this chapter, we will discuss two forms of LIR modularization: 1) detection vs. re-identification modularization, and 2) scene retrieval vs. instance retrieval modularization. We will begin by discussing the motivation for modularization in general. Then, we will discuss detection vs. re-identification modularization and the corresponding model types in Sec. 3.1; this section is mainly a review and synthesis of key issues on the topic. Finally, the majority of the chapter will be devoted to discussion of scene retrieval vs. instance retrieval, and the corresponding Gallery Filter Network (GFN) model¹ in Secs. 3.2 to 3.4. These sections constitute our novel contribution, and we present detailed explanations, experiments, and analysis of our method.

In general, modularization of composite tasks can result in efficiency improvements in multiple ways. First, individual submodels may be trained and/or evaluated separately. Second, it is easier to utilize existing pre-trained models. Third, we can use task specific data that may not apply for the joint task.

However, there can be downsides as well: for sequential tasks, distribution of training inputs to later steps may not match distribution coming out of earlier steps, resulting in poor generalization. In addition, data pre-processing may result in information loss for modular models, *e.g.*, resizing of image chips for the re-identification task.

We will show how these pros and cons result in tradeoffs for modularizing detection and re-identification, and that the negative aspects do not apply to our method of modularizing scene and instance retrieval.

The outline of this chapter is as follows: we cover modularization of detection and re-identification in Sec. 3.1, modularization of scene and instance retrieval in Sec. 3.2, methods for our proposed SeqNeXt person search model and the Gallery Filter Network in Sec. 3.3, experiments and analysis in Sec. 3.4, and a discussion in Sec. 3.5.

¹Our work on this topic was initially published in [192]

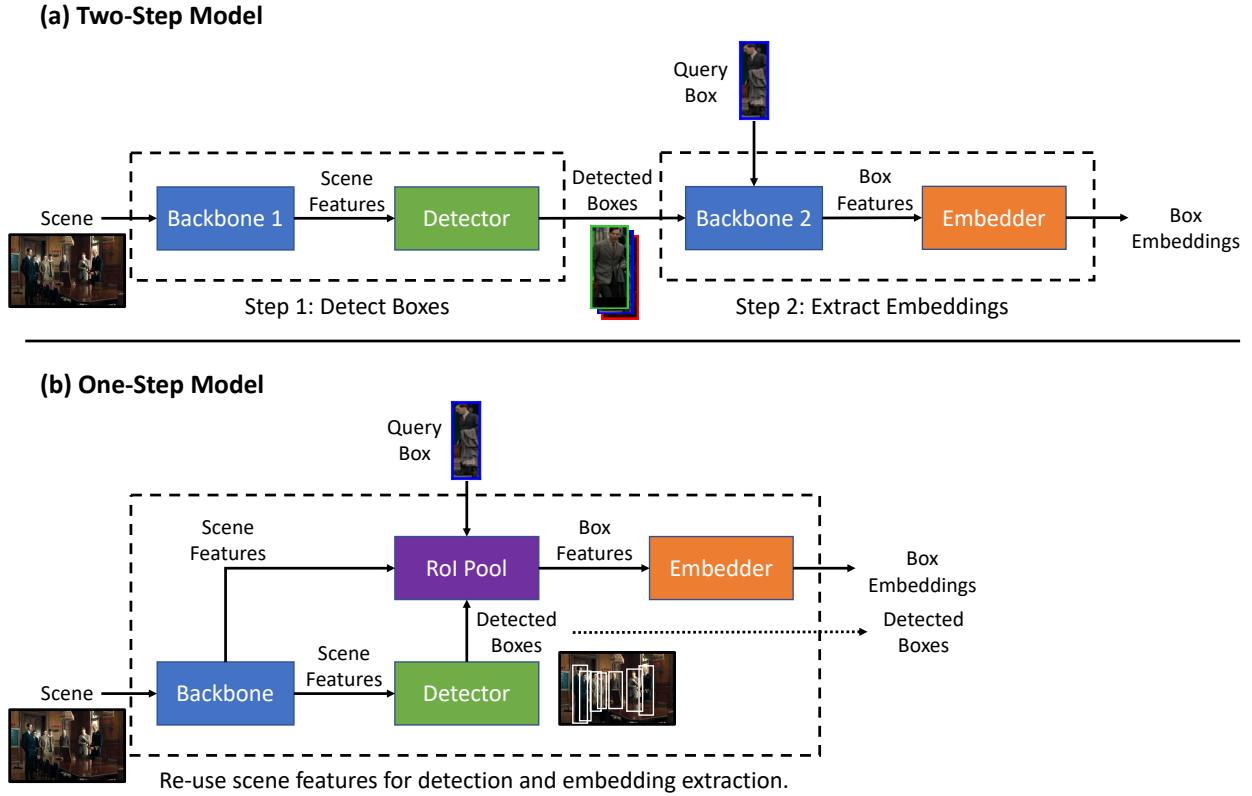


Figure 3.1: One-step (b) vs. two-step (a) person search models. One-step models use a shared set of backbone features for both detection and re-id, while two-step models recompute detector features for each task separately.

3.1 Modularizing Detection and Re-identification

As discussed in Chapter 1, the problem of localized image retrieval (LIR) may be split into two parts: 1) *object detection*, in which all object bounding boxes are localized within each gallery scene and 2) *instance re-identification* (re-id), in which detected object crops are compared against a query object crop. In person search, two-step methods [100]–[103], [120], [121] tackle each of these parts explicitly with separate models, shown in Fig. 3.1a. In contrast, one-step or end-to-end methods [1], [104]–[113], [115], [117], [118], [135]–[138], [151] use a single model, typically sharing backbone features for detection and re-identification, shown in Fig. 3.1b.

Two-step models offer modularity, while end-to-end models offer efficiency in the form of feature re-use, simplified training, and in many cases superior generalization. In the following subsections, we build upon the material in Chapter 2 to explain specific issues with one-step vs. two-step LIR. While these issues have been pointed out individually in other works, this



Figure 3.2: Demonstration of mismatch between ground truth and detected person bounding boxes. Figure from [121].

is the first work to our knowledge to discuss them in a unified setting and draw connections between them.

We discuss the impact of bounding box distribution in Sec. 3.1.1, the influence of scene context in Sec. 3.1.2, efficiency vs. modularity in Sec. 3.1.3, pre-training methodologies in Sec. 3.1.4, the issue of competing detection and re-id losses in Sec. 3.1.5, multi-scale re-id in Sec. 3.1.6, and query-centric models in Sec. 3.1.7.

3.1.1 Bounding Box Distribution

The first and most apparent issue when considering the joint problem of localized image retrieval is the distribution of ground truth boxes used to train the re-id module. As demonstrated in Fig. 3.1a and Fig. 3.1b, it is typical to detect persons in images via rectangular bounding boxes, then use the image features corresponding to this bounding box region for re-id. However, as shown in Fig. 3.2, there is a mismatch in the shape and size of bounding boxes output by the detector, and the ideal ground truth counterparts.

The degree of this mismatch is a function of annotation quality, annotation ambiguity, and detector performance, but a significant misalignment is typical, meaning that a re-id module which is capable of handling this mismatch will perform much better than one which can only function with perfectly aligned ground truth bounding boxes.

This mismatch is noted in the seminal person search paper [2], and was later studied in detail in [121]. In summary, if detection and re-id are considered separately, the re-id module will presumably be less effective in handling the true distribution of boxes output from the detector.

3.1.2 Influence of Scene Context

Another critical factor when considering detection and retrieval as separate problems is the impact on features used for re-id. Typically in a two-step model, image crops are extracted from the original scenes using the bounding box predictions from the detector. These image crops are then fed to a separate model, shown in Fig. 3.1a. While local features within the

bounding box crop will be seen by the re-id network, any scene context outside of the crop is discarded.

In some cases, this will be a positive aspect, as the model will not be biased by extraneous context information that could be irrelevant to retrieval in a separate scene. In [107], an end-to-end model is co-trained with a separate crop-only re-id network in order to explicitly remove scene context from crop embeddings extract from a scene feature map via RoI-align.

In other cases, even information inside the crop is considered, as in [101], where the model learns representations with and without local background information, and concatenates features from the two.

In still other cases, scene context is explicitly desired, as in methods of weakly-supervised person pseudo-labeling [123]–[128], where local context can be critical in assigning identity pseudo-labels when they are not explicitly known. For example, a person in a video frame is likely to be in a similar position in the next video frame, and people traveling in groups may remain in those groups for an extended period.

The framing of LIR allows us to study the impact of scene context for both detection and retrieval, and to understand cases in which it is useful or harmful.

3.1.3 Efficiency vs. Modularity

The clearest benefits of two-step methods come from their inherent modularity. Having separate modules for detection and re-id allows for separate training and inference processes. For example, we can simply train one model for person detection, and one model for person re-id, and then stitch them together. Since research is more active for these two problem categories separately than the combined person search problem, we are able to take advantage of more innovations in the research community in this way. However, this modularity comes at the cost of efficiency in that the modules have more total parameters, and total runtime cost is greater. This is mainly due to the need to recompute backbone features in a two-step model, which is typically one of the most expensive steps in detection.

We note that some benefits of modularity in the one-step vs. two-step case can be captured via approaches such as knowledge distillation, which allows us to use outputs from a pre-trained model as targets for a one-step network. This is used in approaches such as [109], where a separate re-id network is trained only on person crops, and then person embeddings from that network are used to supervise embeddings output by the one-step model.

3.1.4 Pre-training Methodologies

The benefits of modularity extend to the concept of pre-training as well. While pre-training for detection and re-id are both well-established, it is more difficult to define pre-training for end-to-end LIR, and there are seemingly fewer datasets which can exploit this framing. One of our key contributions is to introduce a mechanism of pre-training LIR models [4] in Chapter 4, which can utilize image data with or without labels.

Building on our previous example, there are many datasets containing unlabeled person identities, which can take advantage of unsupervised re-id pre-training mechanisms [162]–[167]. In addition, there are numerous datasets for generic object detection [64], [193], [194], and person detection specifically, [195]–[197], that do not contain identity labels, but are useful for pre-training object detectors alone.

3.1.5 Competing Losses

Another factor in the co-training of detection and retrieval is that the two problems require competing objective functions. In the case of person search, detection requires learning features which are common to all people, while retrieval requires learning features which uniquely distinguish each person identity from others. This inherent conflict is the subject of many person search papers [101], [108], [149], and is a point in favor of the two-step approach, which is deconflicted by design.

3.1.6 Multi-scale Re-id

Another issue that arises when treating re-id as a separate task is how to handle multi-scale views of the same identity. It is common in person re-id to simply resize person image crops to a fixed square size, causing a tradeoff between efficiency and information utilization. When the fixed size is larger than person crop native resolution, there is a loss of efficiency, and when the fixed size is smaller, there is a loss of information. In addition, person scale and aspect ratio distributions are lost in the process of fixed resizing.

These problems can be better handled in the end-to-end framework, where it is typical to extract person features from the backbone feature map using RoI Align [53]. While RoI pooling include RoI align still produces a fixed-size feature map, the neural network backbone can operate on crops in their native resolution and aspect ratio. This framing allows all information in the native pixel space to be used, though when considering purely from the re-id perspective it does result in significant loss of efficiency in processing background pixels, which typically comprise most of a given image.

While the end-to-end framing can benefit multi-scale re-id, the question of how to implement multi-scale features is an open topic in person search, with many works addressing it [110], [113]. The common challenge is what layer or combination of layers in the backbone to use for feature extraction for a given scale of person crop. There are competing pressures with how gradients from object detection and person re-id affect different layers in multi-scale feature models, due to influence of context from the receptive field, and commonness vs. uniqueness challenge previously discussed. When using a Feature Pyramid Network (FPN), it is typical to extract features from just one layer instead of multiple, [3], [4], [110], which compromises detector performance but significantly boosts re-id.

3.1.7 Query-Centric Models

Query-centric models of localized image retrieval condition detection on the query, thereby generating object candidates specific to a given query. In the context of person search, the query embedding is typically generated by the re-id module. This is the case in both one-step [118], [119] and two-step models [120], [121]. This again creates a linkage between re-id and detection which should be studied in a holistic context.

The query-centric framing also enables a novel pre-training mechanism discussed in Chapter 5, and has connections to prompt-based foundation models, which we discuss in Chapter 6.

3.2 Modularizing Scene and Instance Retrieval

Next, we consider how to modularize LIR into scene retrieval and instance retrieval. For both one-step and two-step LIR models, the same steps are needed: 1) computation of detector backbone features, 2) detection of instance bounding boxes, and 3) computation of feature embeddings for each bounding box, to be used for retrieval. Improvement of identity search model efficiency is typically focused on reducing the cost of one or more of these steps. We propose the second and third steps can be avoided altogether for some subset of gallery scenes by splitting the retrieval process into two phases: scene retrieval, followed by typical instance retrieval. We call the module implementing scene retrieval the Gallery Filter Network (GFN), since its function is to filter scenes from the gallery.

The two-phase process is visualized in Fig. 3.3. In the first phase, shown as 1a, 1b, and 1c in Fig. 3.3, the Gallery Filter Network discards scenes unlikely to contain the query person. To do so, we extract embeddings correspond to both persons and scenes (1a), compute cosine similarity between person and scene embeddings (1b), then discard scenes falling below some similarity threshold with the query (1c). The second phase, shown as 2a, 2b, 2c, and 2d in Fig. 3.3, is the standard person retrieval process, in which persons are detected in high scoring scenes (2a), corresponding embeddings extracted from detected boxes (2b), and these embeddings are compared to the query (2c) to produce a ranking (2d).

By performing the cheaper query-scene comparison before detection is needed, the GFN allows for a modular computational pipeline for practical systems, in which one process can determine which scenes are of interest, and another can detect and extract person embeddings only for interesting scenes. This could serve as an efficient filter for video frames in a high frame rate context, or to efficiently reduce the search space when querying large image databases.

The GFN also provides a mechanism to incorporate global context into the gallery ranking process. Rather than combining global context features with intermediate model features as in [115], [120], we explicitly compare global scene embeddings to query embeddings. The resulting score can be used not only to filter out gallery scenes using a hard threshold, but also to weight predicted box scores for remaining scenes.

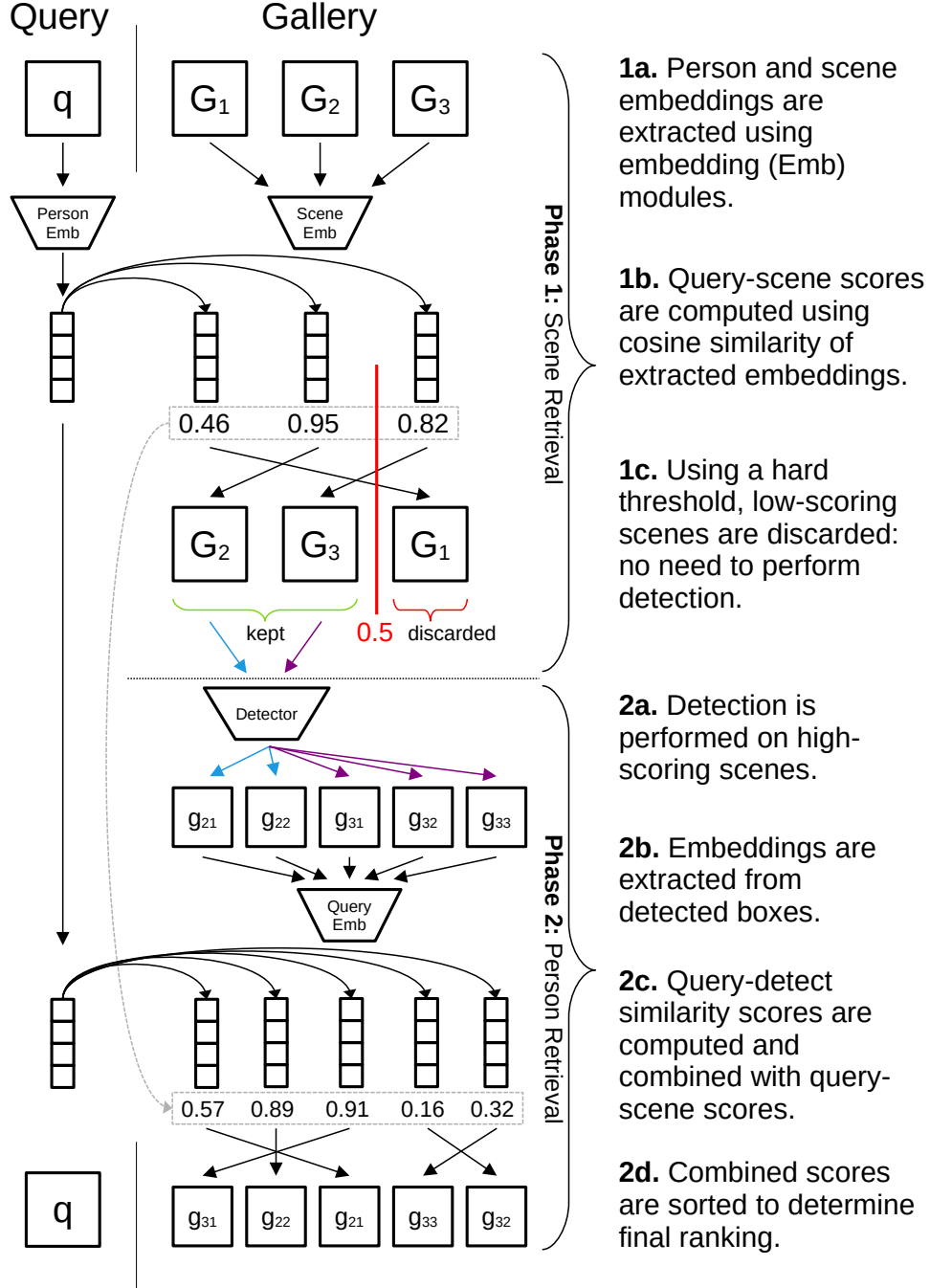


Figure 3.3: An illustration of our proposed two-phase retrieval inference pipeline.

We show in Sec. 3.4 that both the hard-thresholding and score-weighting mechanisms are effective for the benchmark PRW and CUHK-SYSU datasets, resulting in state-of-the-art retrieval performance: +2.7% top-1 accuracy on the PRW dataset over the previous best model. In addition, we demonstrate improved efficiency, with over 50% per-query cost savings on the CUHK-SYSU dataset vs. same the model without the GFN. We also make contributions to the data processing and evaluation frameworks that are used by most person search methods with publicly available code. That work is described in Supplementary Material Appendix A.1.

Our novel contributions in this chapter are 1) the Gallery Filter Network: A novel module for learning query-scene similarity scores which efficiently reduces retrieval gallery size via hard-thresholding, while improving detected embedding ranking with global scene information via score-weighting, 2) performance improvements and removal of unneeded elements in the SeqNet person search model [111], dubbed SeqNeXt, and 3) standardized tooling for the data pipeline and evaluation frameworks typically used for the PRW and CUHK-SYSU datasets, which is extensible to new datasets. All of our code and model configurations are made publicly available².

3.3 Methods

In this section, we describe models, optimization objectives, and implementation details for our SeqNeXt person search model in Sec. 3.3.1 and the Gallery Filter Network in Sec. 3.3.2.

3.3.1 SeqNeXt Person Search Model

We introduce a new person search model called SeqNeXt, which is based on the previous SeqNet person search model [111], shown in Fig. 3.4. Both models are one-step person search models, which take queries and gallery scenes as input, and output ranked matches to the query detected in gallery scenes. In SeqNet, a single-scale two-stage Faster R-CNN [20] model is used, which first computes refined boxes with Faster R-CNN, then extracts features from the refined boxes using RoI-Align to use for re-id features and further box refinement. By using this additional refinement step before re-id feature extraction, SeqNet produces more accurate features for re-id, leading to improved metrics.

Our goal is to improve upon SeqNet in both efficiency and performance: we make modifications to the model backbone, simplify the two-stage detection pipeline, and improve the training recipe. Since our proposed model inherits heavily from SeqNet, and uses a ConvNeXt base, we refer to it simply as SeqNeXt to distinguish it from the original model. In the rest of this subsection, we detail the SeqNeXt forward pass, describe data augmentation, and explain training objective functions.

Overview of SeqNeXt Forward Pass. A detailed model diagram of SeqNeXt, containing a breakdown of which layers are used, and how losses are computed, is given in Fig. 3.5. The

²Project repository: <https://github.com/LukeJaffe/GFN>

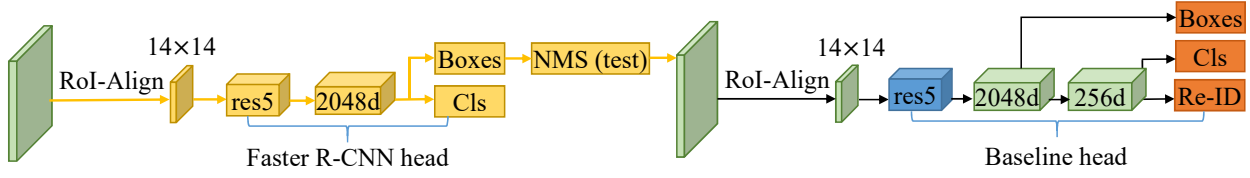


Figure 3.4: SeqNet model architecture, which sequentially performs detection and re-id embedding extraction. Figure from [111].

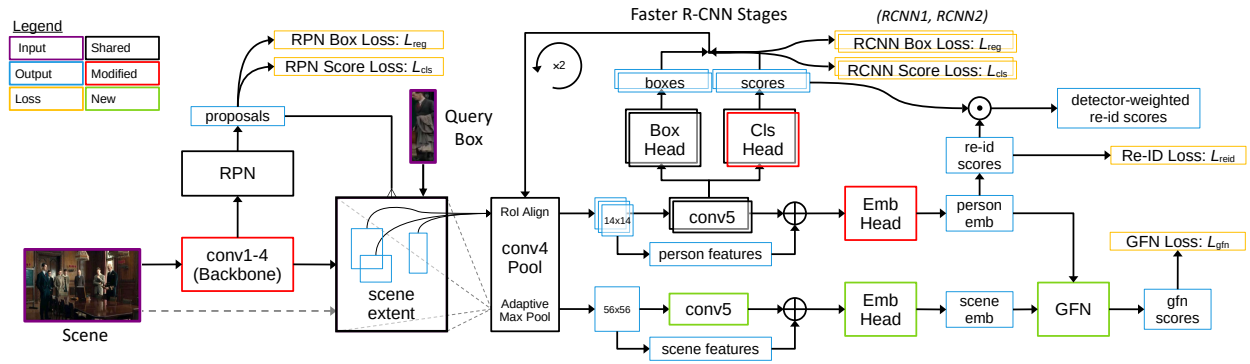


Figure 3.5: Full architecture diagram of the SeqNeXt person search model augmented with the GFN. Modules modified from SeqNet, shown in Fig. 3.4, are colored red, and new modules, related to the GFN, are colored green.

model follows the standard Faster R-CNN paradigm, with images passing through backbone layers `conv1-conv4`, and features from `conv4` being used to generate proposals via the RPN. Like Cascade R-CNN [19], there are two steps of box refinement with two corresponding Faster R-CNN stages, denoted in Fig. 3.5 by the $\times 2$ cycle arrow and overlapping boxes for the black `conv5` modules, black Box Head modules, black and red Cls Head modules, blue “boxes”, and blue “scores”.

The `conv4` features are pooled for RPN proposals and passed through the `conv5` head, shown in black in Fig. 3.5, to generate refined proposals labeled as “boxes”, shown in blue in Fig. 3.5, via the Box Head, shown in black in Fig. 3.5. This process is repeated with the boxes from the first Faster R-CNN stage, using a separate box head, shown in black in Fig. 3.5, to generate the final boxes, labeled as “boxes” and shown in blue in Fig. 3.5. `conv4` features are also used to generate query and gallery person embeddings in the same way: the person box passes through the `conv4` Pool block and then a duplicated `conv5` head, shown in black, and `conv4`, `conv5` features are concatenated and passed through an embedding (Emb) head, shown in red. The same process is used to generate scene embeddings, by passing scene features through the `conv4` Pool block and then a duplicated `conv5` head, shown in

green, and `conv4`, `conv5` features are concatenated and passed through an embedding (Emb) head, shown in green. In the `conv4` Pool block, RoI Align[53] is used for person and proposal features, while adaptive max pooling is used for scene features. GFN scores are generated using person and scene embeddings from the same or different scenes. Person re-id scores are combined with the score output of the second R-CNN stage to produce detector-weighted scores.

Backbone Features. Following SeqNet’s usage of the first four CNN blocks (`conv1-4`) from ResNet50 for backbone features, we use the analogous layers in terms of downsampling from ConvNeXt, also referred to as `conv1-4` for convenience. While the original ResNet50 [198] backbone used in SeqNet and most other person search models has been effective to date, many newer architectures have since been introduced. With the recent advent of vision transformers (ViT) [199] and a cascade of improvements including the Swin Transformer [200] and the Pyramid Vision Transformer (v2) [201], used by the PSTR person search model [113], transformer-based feature extraction has increased in popularity. However, there is still an efficiency gap with CNN models, and newer CNNs including ConvNeXt [202] have closed the performance gap with ViT-based models, while retaining the inherent efficiency of convolutional layers. For this reason, we explore ConvNeXt for our model backbone as an improvement to ResNet50, which is more efficient than ViT alternatives.

Multi-Stage Refinement and Inference. We simplify the detection pipeline of SeqNet by duplicating the Faster R-CNN head [20] in place of the Norm-Aware Embedding (NAE) head from [108]. We still weight re-id scores using the scores output from the detector, but use the second-stage class score instead of the first-stage as in SeqNet. This is depicted in Fig. 3.5 as “detector-weighted re-id scores”. We note that although SeqNet technically trains the NAE, it does not use the NAE score at inference time, so it is consistent to remove it altogether. Additionally during inference, we do not use the Context Bipartite Graph Matching (CBGM) algorithm from SeqNet.

Augmentation. Following resizing images to 900×1500 at training time, which we call Window Resize, we employ one of two random cropping methods with equal probability: 1) Random Focused Crop (RFC): randomly take a 512×512 crop in the original image resolution which contains at least one known person, 2) Random Safe Crop (RSC): randomly crop the image such that all persons are contained, then resize to 512×512 . This cropping strategy allowed us to train with larger batch sizes, while benefiting performance with improved regularization. At inference time, we resize to 900×1500 , as in other models. We also consider a variant of Random Focused Crop (RFC2), which resizes images so the “focused” person box is not clipped.

Objective Diagram Setup. To understand all objective functions for re-id and the GFN, we introduce Fig. 3.6. The idea is that we can show a simple example containing three different scenes with two different person identities to demonstrate problems with naïve loss functions, and their solutions. In the “System Setup” of Fig. 3.6 (top left), we show how the three scenes (1, 2, 3) and two person identities (A, B) map to a corresponding simple diagram: the circles represent scene images which contain one or more different person identities. This diagram format is used in Fig. 3.6a, 3.6b, 3.6c, and 3.6e. For all diagrams in Fig. 3.6, we

borrow from the spring analogy for metric learning used in *DrLIM* [72], with the concept of *attractions* and *repulsions*. Attraction is represented by green arrows, and repulsion by red arrows. The details of each part of this figure will be explained in the following sections.

Objective Functions. For the re-id loss of SeqNeXt, we employ the standard Online Instance Matching (OIM) Loss [1], represented as $\mathcal{L}_{\text{reid}}$. This is visualized in Fig. 3.6a. The Gallery Filter Network losses, $\mathcal{L}_{\text{gfn}}$, are explained in Sec. 3.3.2.

The detector loss is the sum of classification and box regression losses from the RPN, and the two Faster R-CNN stages comprising conv5, Box Head, and Cls Head layers, expressed as:

$$\mathcal{L}_{\text{det}} = \sum_{m \in M} \mathcal{L}_{\text{cls}}^m + \mathcal{L}_{\text{reg}}^m, \quad M = \{\text{RPN}, \text{RCNN1}, \text{RCNN2}\} \quad (3.1)$$

The full loss is the sum of the detector, re-id, and GFN losses:

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{reid}} + \mathcal{L}_{\text{gfn}} \quad (3.2)$$

3.3.2 Gallery Filter Network

In building the Gallery Filter Network, our goal is to design a module which removes low-scoring scenes, and reweights boxes from higher-scoring scenes. Let s_{reid} be the cosine similarity of a predicted gallery box embedding with the query embedding, s_{det} be the detector box score, s_{gfn} be the cosine similarity for the corresponding gallery scene from the GFN, $\sigma(x) = \frac{e^{-x}}{1+e^{-x}}$, α be a temperature constant, and λ_{gfn} be the GFN score threshold. At inference time, scenes scoring below λ_{gfn} are removed, and detection is performed for remaining scenes, with the final score for detected boxes given by

$$s_{\text{final}} = s_{\text{reid}} \cdot s_{\text{det}} \cdot \sigma(s_{\text{gfn}}/\alpha).$$

The module should discriminate as many scenes below λ_{gfn} as possible, while positively impacting the scores of boxes from any remaining scenes. To this end, we consider three variations of the standard contrastive objective [73], [74] in Secs. 3.3.2.1 to 3.3.2.3, in addition to a number of architectural and optimization considerations in Sec. 3.3.2.4. In Sec. 3.3.2.1, we introduce a baseline objective which directly computes similarity between persons and scenes. In Sec. 3.3.2.2, we introduce an objective which improves upon the baseline by combining query and scene embeddings and consequently disentangling conflicting optimization interactions. We will show in Sec. 3.4 that this objective achieves superior results. In Sec. 3.3.2.3, we consider a control objective which simply compares scene embeddings: this objective is intended to show that our improved objective is not merely learning some simpler function, and is benefitting from both query and scene information.

3.3.2.1 Baseline Objective

The goal of the baseline GFN optimization is to push person embeddings toward scene embeddings when a person is contained within a scene, and to pull them apart when the

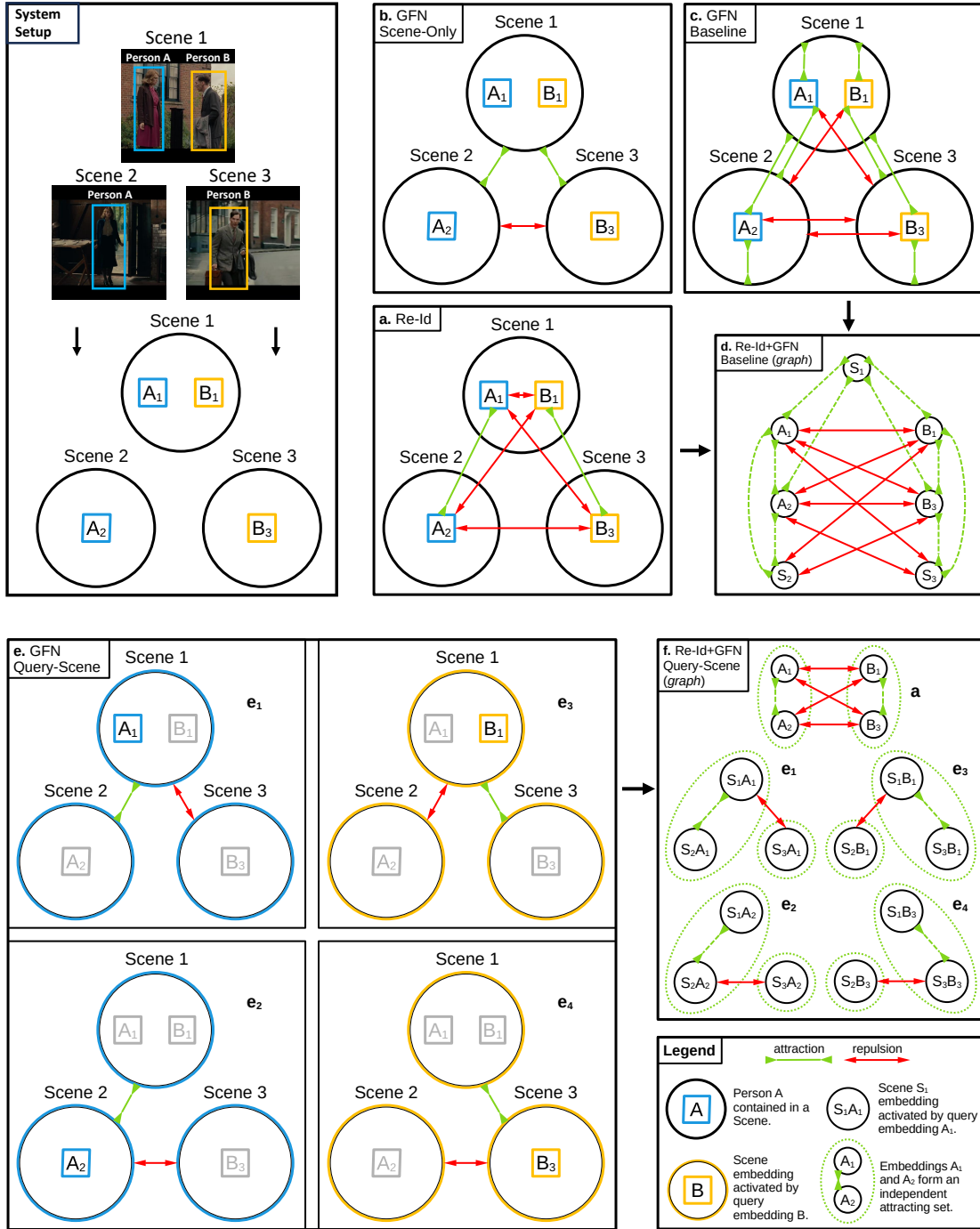


Figure 3.6: Visual representation of the re-id and GFN optimization objectives. We show a system of three scenes with two unique person identities, represented in the “System Setup”. Green connectors represent attraction, meaning two embeddings are pushed together by an objective, and red connectors represent repulsion, meaning two embeddings are pulled apart by an objective. The goal of this figure is to visualize the problem with our initial GFN baseline objective shown in (c), and show how we arrived at the improved GFN query-scene objective shown in (e) and (f).

person is not in the scene, shown in Fig. 3.6c.

Let $x_i \in \mathbb{R}^d$ denote the embedding extracted from person q_i located in some scene s_j . Let $y_j \in \mathbb{R}^d$ denote the embedding extracted from scene s_j . Let X be the set of all person embeddings x_i , and Y the set of all scene embeddings y_j , with $N = |X|$, $M = |Y|$.

We define the query-scene indicator function to denote positive query-scene pairs as

$$\mathcal{I}_{i,j}^Q = \begin{cases} 1 & \text{if } q_i \text{ present in } s_j \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

We then define a set to denote indices for a specific positive pair and all negative pairs: $K_{i,j}^Q = \{k \in 1, \dots, M \mid k = j \text{ or } \mathcal{I}_{i,k}^Q = 0\}$. Define $\text{sim}(u, v) = u^\top v / \|u\| \|v\|$, the cosine similarity between two $u, v \in \mathbb{R}^d$, and τ is a temperature constant. Then the loss for a positive query-scene pair is the cross-entropy loss

$$\ell_{i,j}^Q = -\log \frac{\exp(\text{sim}(x_i, y_j)/\tau)}{\sum_{k \in K_{i,j}^Q} \exp(\text{sim}(x_i, y_k)/\tau)} \quad (3.4)$$

The baseline Gallery Filter Network loss sums positive pair losses over all query-scene pairs:

$$\mathcal{L}_{\text{gfn}}^Q = \sum_{i=1}^N \sum_{j=1}^M \mathcal{I}_{i,j}^Q \ell_{i,j}^Q \quad (3.5)$$

3.3.2.2 Combined Query-Scene Objective

While it is possible to train the GFN directly with person and scene embeddings using the loss in Eq. (3.5), we show that this objective is ill-posed without modification. The problem is that we have constructed a system of opposing attractions and repulsions. We can formalize this concept by interpreting the system as a graph $G(V, E)$, visualized in Fig. 3.6d. Let the vertices V correspond to person, scene, and/or combined person-scene embeddings, where a solid line edge in E , shown as a red arrow in Fig. 3.6d, connecting any two nodes in V represents a negative pair used in the optimization objective. Pairs of nodes connected by green dashed arrows represent positive pairs in the optimization objective. Let any group of nodes connected by green dashed arrows (not edges in G) be an independent set, meaning that no two nodes in the set are adjacent in G . Then, each connected component of G must be multipartite, or the optimization problem will be ill-posed by design, as in the baseline objective.

To learn whether a person is contained within a scene while preventing this conflict of attractions and repulsions, we need to apply some unique transformation to query and scene embeddings before the optimization. One such option is to combine a query person embedding separately with the query scene and gallery scene embeddings to produce fused representations. This allows us to disentangle the web of interactions between query and

scene embeddings, while still learning the desired relationship, visualized in Fig. 3.6e. The person embedding used to fuse with each scene embedding in a pair is left colored either blue or orange, and the corresponding scenes are colored according to that person embedding. Person embeddings present in scenes which are not used are grayed out.

In the graph-based presentation, shown in Fig. 3.6f, this modified scheme using query-scene embeddings will always result in a graph comprising some number of star graph connected components. Since these star graph components are multipartite by design, the issue of conflicting attractions and repulsions is avoided.

To combine a query and scene embedding into a single query-scene embedding, we define a function $f : \mathbb{R}^d, \mathbb{R}^d \rightarrow \mathbb{R}^d$, such that $z_{i,j} = f(x_i, y_j)$ and $w_i = f(x_i, y^{x_i})$, where y^{x_i} is the embedding of the scene that person i is present in. Borrowing from SENet [122] and QEEPS [118], we choose a sigmoid-activated elementwise gating, with $\odot$ used for elementwise product. “BN” is a Batch Normalization layer, to mirror the architecture of the other embedding heads, and β is a temperature constant.

$$f(x, y) = \text{BN}(\sigma(x/\beta) \odot y) \quad (3.6)$$

Other choices are possible for f , but the elementwise-product is critical, because it excites the features most relevant to a given query within a scene, eliciting the relationship shown in Fig. 3.6e. More learnable layers could be added as well, such as a MLP applied to the query embedding similar to SENet, but we did not find benefit from more complex architectures.

The loss for a positive query-scene pair is the cross-entropy loss

$$\ell_{i,j}^C = -\log \frac{\exp(\text{sim}(w_i, z_{i,j})/\tau)}{\sum_{k \in K_{i,j}^Q} \exp(\text{sim}(w_i, z_{i,k})/\tau)} \quad (3.7)$$

The query-scene combined Gallery Filter Network loss sums positive pair losses over all query-scene pairs:

$$\mathcal{L}_{\text{gfn}}^C = \sum_{i=1}^N \sum_{j=1}^M \mathcal{I}_{i,j}^Q \ell_{i,j}^C \quad (3.8)$$

where N denotes the number of person embeddings, and M denotes the number of scene embeddings.

3.3.2.3 Scene-Only Objective

As a control for the query-scene objective, we also define a simpler objective which uses scene embeddings only, depicted in Fig. 3.6b. This objective attempts to learn the less discriminative concept of whether two scenes share any persons in common, and has the same optimization issue of conflicting attractions and repulsions as the baseline objective. At inference time, it is used in the same way as the other GFN methods.

We define the scene-scene indicator function to denote positive scene-scene pairs as

$$\mathcal{I}_{i,j}^S = \begin{cases} 1 & \text{if } s_i \text{ shares any } q \text{ in common with } s_j \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

Similar to Sec. 3.3.2.1, we define an index set: $K_{i,j}^S = \{k \in 1, \dots, M \mid k = j \text{ or } \mathcal{I}_{i,j}^S = 0\}$. Then the loss for a positive scene-scene pair is the cross-entropy loss

$$\ell_{i,j}^S = -\log \frac{\exp(\text{sim}(y_i, y_j)/\tau)}{\sum_{k \in K_{i,j}^S} \exp(\text{sim}(y_i, y_k)/\tau)} \quad (3.10)$$

The scene-only Gallery Filter Network loss sums positive pair losses over all scene-scene pairs:

$$\mathcal{L}_{\text{gfn}}^S = \sum_{i=1}^M \sum_{j=1}^M [i \neq j] \mathcal{I}_{i,j}^S \ell_{i,j}^S \quad (3.11)$$

where $[i \neq j]$ is 1 if $i \neq j$ else 0.

3.3.2.4 Architecture and Optimization

We consider a number of design choices for the architecture and optimization strategy of the GFN to improve its performance.

Architecture. Scene embeddings are extracted in the same way as person embeddings using the `conv4` Pool layer in Fig. 3.5, except that a larger 56×56 pooling size with adaptive max pooling is used vs. the person pooling size of 14×14 with RoI Align. This larger scene pooling size is needed to adequately summarize scene information, since the scene extent is much larger than a typical person bounding box. In addition, the scene `conv5` head and Emb Head, which are green in Fig. 3.5, are duplicated from the corresponding person modules (no weight-sharing), which are black and red in Fig. 3.5 respectively.

Lookup Table. Similar to the methodology used for the OIM objective [1], we use a lookup table (LUT) to store scene and person embeddings from previous batches, refreshing the LUT fully during each training epoch. We compare the person and scene embeddings in each batch, which have gradients, with some subset of the embeddings in the LUT, which do not have gradients. Therefore only comparisons of embeddings within the batch, or between the batch and the LUT, have gradients.

Query Prototype Embeddings. Rather than using person embeddings directly from a given batch, we can use the identity prototype embeddings stored in the OIM LUT for optimizing the GFN, similar to [135]. These moving average identity embeddings stored in the OIM LUT are likely to be closer to the respective identity cluster centroid than any single batch embedding, and will change less between iterations. Therefore, using them could help regularize the GFN optimization.

To optimize using these OIM LUT embeddings, we lookup the corresponding identity for a given batch person identity in the OIM LUT during training, and substitute that into the objective. In doing so, we discard gradients from batch person embeddings, meaning that we only pass gradients through scene embeddings, and therefore only update the scene embedding module. This has the effect of reducing the impact of the GFN optimization on person embeddings, since person embeddings will change only indirectly without direct gradient updates. This choice is examined in an ablation in Sec. 3.4.4.

3.4 Experiments and Analysis

In this section, we describe datasets and evaluation methods used in Sec. 3.4.1, implementation details of SeqNeXt and the GFN in Sec. 3.4.2, a comparison to state-of-the-art person search methods in Sec. 3.4.3, several ablation studies for our models in Sec. 3.4.4, an analysis of filtering using the GFN in Sec. 3.4.5, a qualitative analysis of real person search examples in Sec. 3.4.6, and visualization of features for SeqNeXt and the GFN in Sec. 3.4.7.

3.4.1 Datasets and Evaluation

Datasets. For our experiments, we use the two standard person search datasets, CUHK-SYSU [1], and *Person Re-identification in the Wild* (PRW) [100]. CUHK-SYSU comprises a mixture of imagery from hand-held cameras, and shots from movies and TV shows, resulting in significant visual diversity. It contains 18,184 scene images annotated with 96,143 person bounding boxes from tracked (known) and untracked (unknown) persons, with 8,432 known identities. PRW comprises video frames from six surveillance cameras at Tsinghua University in Hong Kong. It contains 11,816 scene images annotated with 43,110 person bounding boxes from known and unknown persons, with 932 known identities.

The standard test retrieval partition for the CUHK-SYSU dataset has 2,900 query persons, with a gallery size of 100 scenes per query. The standard test retrieval partition for the PRW dataset has 2,057 query persons, and uses all 6,112 test scenes in the gallery, excluding the identity. For a more robust analysis, we additionally divide the given train set into separate train and validation sets, further discussed in Supplementary Material Appendix A.1.

Evaluation Metrics. As in other works, we use the standard re-id metrics of mean average precision (mAP), and top-1 accuracy (top-1). For detection metrics, we use recall and average precision at 0.5 IoU (Recall, AP).

In addition, we show GFN metrics mAP and top-1, which are computed as metrics of scene retrieval using GFN scores. To calculate these values, we compute the GFN score for each scene, and consider a gallery scene a match to the query if the query person is present in it.

3.4.2 Implementation Details

We use SGD optimizer with momentum for ResNet models, with starting learning rate $3e-3$, and Adam [203] for ConvNeXt models, with starting learning rate $1e-4$. We train all models for 30 epochs, reducing the learning rate by a factor of 10 at epochs 15 and 25. Gradients are clipped to norm 10 for all models.

Models are trained on a single Quadro RTX 6000 GPU (24 GB VRAM), and 30 epoch training time using the final model configuration takes 11 hours for the PRW dataset, and 21 hours for the CUHK-SYSU dataset.

Our baseline model used for ablation studies has a ConvNeXt Base backbone, embedding dimension 2,048, scene embedding pool size 56×56 , and is trained with 512×512 image crops using the combined cropping strategy (RSC+RFC). It uses the combined prototype feature version of the GFN objective. The final model configuration, used for comparison to other state-of-the-art models, is trained with 640×640 image crops using the altered combined cropping strategy (RSC+RFC2). It uses the combined batch feature version of the GFN objective.

Model Details. We set the OIM scalar (inverse temperature) parameter to 30.0 as in [111], with an OIM circular queue size of 5,000 for CUHK-SYSU and 500 for PRW. The OIM momentum parameter is also left at 0.5. For the GFN, the training temperature parameter is 0.1, and the GFN gating function temperature parameter is 0.2. During training, we use a batch size of 12 for ResNet50 backbone models, and a batch size of 8 for ConvNeXt backbone models.

For the ResNet50 backbone, we freeze all batch norm layers, and all weights through the `conv1` layer of the model. For ConvNeXt backbones, we freeze only the `conv1` layer of the model. All backbones are initialized using weights from pre-training on ImageNet1k [204].

We use automatic-mixed precision (AMP), which significantly reduces all training and inference times. To avoid `float16` overflow, we refactor all loss functions to divide before summation when computing mean reduction. This increases likelihood of underflow, but results in more stable training overall.

GFN Sampling Strategies. Since we are unable to use the entire GFN LUT to form loss pairs in any given batch due to memory limitations, we have a choice about which LUT embeddings to select for the GFN optimization. By default, for each query person present in the current batch, we sample one matching scene embedding and the person embeddings for all persons in that scene. In addition, we consider sampling a “hard negative” scene, defined as a scene which shares at least one person identity in common with the query scene, but that does not contain the query person identity. An ablation for related choices is considered in Sec. 3.4.4.

3.4.3 Comparison to State-of-the-art

We show a comparison of state-of-the-art methods on the standard benchmarks in Tab. 3.1. The GFN benefits all metrics, especially top-1 accuracy for the PRW dataset, which improves

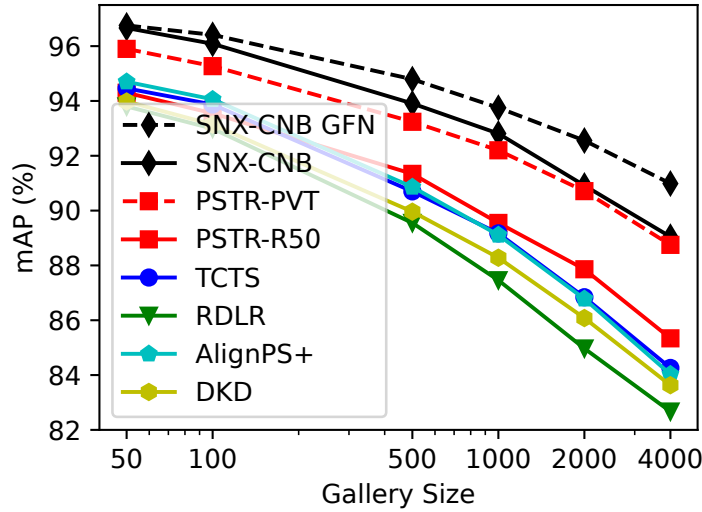


Figure 3.7: Effect of gallery size on mAP for the CUHK-SYSU dataset. SNX-CNB denotes SeqNeXt ConvNeXt Base. GFN helps more as gallery size increases.

by 4.6% for the ResNet50 backbone, and 2.9% for the ConvNeXt Base backbone. The greater improvement on PRW than CUHK-SYSU is likely due to the greater frequency of top-1 matches in the PRW dataset that are similar to the query. Since PRW images all come from video data, there is typically at least one matching frame in the gallery that is very similar to the query frame, and similar scenes result in high GFN score-weighting. This GFN score-weighting can make up for low re-id scores when it is challenging to compare person embeddings alone, *e.g.* when there are very few pixels on the person of interest.

In addition, our best model, SeqNeXt+GFN with ConvNext Base, improves mAP by 1.8% on PRW and 1.2% on CUHK-SYSU over the previous best PSTR model. This benefit extends to larger gallery sizes for CUHK-SYSU, shown in Fig. 3.7. In fact, the GFN score-weighting helps more as gallery size increases. This is expected, since the benefit of down-weighting contextually-unlikely scenes, vs. discriminating between persons within a single scene, has a greater effect when there are more scenes compared against.

The GFN benefits CUHK-SYSU retrieval scenarios with occluded or low-resolution query persons, as shown in Tab. 3.2. This shows that high quality query person views are not essential to the function of the GFN.

The GFN also benefits both cross-camera and same-camera retrieval, as shown in Tab. 3.3. Strong cross-camera performance shows that the GFN can generalize to varying locations, and does not simply choose the scene which is the most visually similar. Strong same-camera performance shows that the GFN is able to use query information, even when all gallery scenes are contextually similar.

To showcase these benefits, we provide qualitative results in Sec. 3.4.6. These examples

Method	Backbone	CUHK-SYSU		PRW	
		mAP	top-1	mAP	top-1
<i>Two-step</i>					
IDE [100]	ResNet50	-	-	20.5	48.3
MGTS [101]	VGG16	83.0	83.7	32.6	72.1
CLSA [102]	ResNet50	87.2	88.5	38.7	65.0
IGPN [120]	ResNet50	90.3	91.4	47.2	87.0
RDLR [103]	ResNet50	93.0	94.2	42.9	70.2
TCTS [121]	ResNet50	93.9	95.1	46.8	87.5
<i>End-to-end</i>					
OIM [1]	ResNet50	75.5	78.7	21.3	49.4
IAN [137]	ResNet50	76.3	80.1	23.0	61.9
NPSM [104]	ResNet50	77.9	81.2	24.2	53.1
RCAA [117]	ResNet50	79.3	81.3	-	-
CTXG [136]	ResNet50	84.1	86.5	33.4	73.6
QEEPS [118]	ResNet50	88.9	89.1	37.1	76.7
APNet [105]	ResNet50	88.9	89.3	41.9	81.4
HOIM [106]	ResNet50	89.7	90.8	39.8	80.4
BINet [107]	ResNet50	90.0	90.7	45.3	81.7
NAE+ [108]	ResNet50	92.1	92.9	44.0	81.1
PGSFL [135]	ResNet50	92.3	94.7	44.2	85.2
DKD [109]	ResNet50	93.1	94.2	50.5	87.1
DMRN [138]	ResNet50	93.2	94.2	46.9	83.3
AGWF [151]	ResNet50	93.3	94.2	53.3	87.7
AlignPS [110]	ResNet50	94.0	94.5	46.1	82.1
SeqNet [111]	ResNet50	93.8	94.6	46.7	83.4
SeqNet+CBGM [111]	ResNet50	94.8	95.7	47.6	87.6
COAT [112]	ResNet50	94.2	94.7	53.3	87.4
COAT+CBGM [112]	ResNet50	94.8	95.2	54.0	89.1
MHGAM [115]	ResNet50	94.9	95.9	47.9	88.0
PSTR [113]	ResNet50	94.2	95.2	50.1	87.9
PSTR [113]	PVTv2-B2	95.2	96.2	56.5	89.7
SeqNeXt (ours)	ResNet50	94.1	94.7	50.8	86.0
SeqNeXt+GFN (ours)	ResNet50	94.7	95.3	51.3	90.6
SeqNeXt (ours)	ConvNeXt	96.1	96.5	57.6	89.5
SeqNeXt+GFN (ours)	ConvNeXt	96.4	97.0	58.3	92.4

Table 3.1: Standard performance metrics mAP and top-1 accuracy on the benchmark CUHK-SYSU and PRW datasets are compared for state-of-the-art *two-step* and *end-to-end* models. ConvNeXt backbone = ConvNeXt Base.

Occluded	mAP	top-1
SeqNeXt	91.1	89.8
SeqNeXt+GFN	92.0	90.9
Low-Resolution	mAP	top-1
SeqNeXt	91.4	92.4
SeqNeXt+GFN	92.0	93.1

Table 3.2: Performance metrics on two CUHK-SYSU retrieval partitions using either Occluded (top) or Low-Resolution (bottom) query persons.

Method	Same Cam ID		Cross Cam ID	
	mAP	top-1	mAP	top-1
HOIM [106]	-	-	36.5	65.0
NAE+ [108]	-	-	40.0	67.5
SeqNet [111]	-	-	43.6	68.5
SeqNet+CBGM [111]	-	-	44.3	70.6
AGWF [151]	-	-	48.0	73.2
COAT [112]	-	-	50.9	75.1
COAT+CBGM [112]	-	-	51.7	76.1
SeqNeXt (ours)	82.9	98.5	55.3	80.5
SeqNeXt+GFN (ours)	85.1	98.6	56.4	82.1

Table 3.3: Performance on the PRW test set for query and gallery scenes from the same camera (Same Cam ID) or different cameras (Cross Cam ID).

show that the GFN uses local person information combined with global context to improve retrieval ranking, even in the presence of difficult confusers.

3.4.4 Ablation Studies

We conduct a series of ablations using the PRW dataset to show how detection, re-id, and GFN performance are each impacted by variations in model architecture, data augmentation, and GFN design choices.

In the corresponding metrics tables, we show re-id results by presenting the GFN-modified scores as mAP and top-1, and the difference between unmodified mAP and top-1 with Δ mAP and Δ top-1. This highlights the change in re-id performance specifically from the GFN score-weighting. To indicate the baseline configuration in a table, we use the † symbol, and the

final model configuration is highlighted in gray. The difference between these configurations is explained in Sec. 3.4.2.

GFN Objective. We analyze the impact of the various GFN objective choices discussed in Sec. 3.3.2. Comparisons are shown in Tab. 3.4: “None” does not use the GFN, Scene-Only uses the objective in Sec. 3.3.2.3, Base uses the baseline objective in Sec. 3.3.2.1, Combined (Comb.) uses the query-scene objective in Sec. 3.3.2.2, Batch indicates that batch query embeddings are used, Proto indicates that prototype query embeddings are used. Baseline model is marked with †, final model is highlighted gray.

Most importantly, the re-id mAP performance without the GFN is relatively high, but the re-id top-1 performance is much lower than the best GFN methods. Conversely, the Scene-Only method achieves competitive re-id top-1 performance, but reduced re-id mAP. This is because the Scene-Only method is effective at retrieving a scene which is contextually-identical to the query, but struggles with more nuanced scenes that benefit from query embeddings as in the combined query-scene method.

The Base methods were found to be significantly worse than all other methods, with GFN score-weighting actually reducing GFN performance. The Combined methods were the most effective, better than the Base and Scene-Only methods for both re-id and GFN-only stats, showcasing the improvements discussed in Sec. 3.3.2.2. In addition, the success of the Combined objective can be explained by two factors: 1) similarity relationship between scene embeddings and 2) query information given by query-scene embeddings. The Scene-Only objective, which uses only similarity between scene embeddings, is functional but not as effective as the Combined objective, which uses both scene similarity and query information. Since the Scene-Only objective incorporates background information, and does not use query information, we reason that the provided additional benefit of the Combined objective comes from the described mechanism of query gating of scene features, and not from *e.g.*, simple matching of the query background with the gallery scene image.

Finally, the Batch and Proto modifiers to the Combined and Base methods were found to be relatively similar in performance. Since the Proto method is simpler and more efficient, we use it for the baseline model configuration.

Model Modifications. We consider how changes to the SeqNet architecture impact performance, including usage of a second Faster R-CNN head instead of the NAE head, denoted by the red Cls Head in Fig. 3.5, and usage of the second detector stage score instead of the first stage score during inference, denoted by the blue score boxes in Fig. 3.5. Results are shown in Tab. 3.5.

Using the ConvNeXt Base backbone instead of ResNet50 does not improve detection performance, but it significantly improves re-id performance, especially mAP, by 7-8%. Using the first stage score significantly helps detection performance, but it reduces re-id performance.

Image Augmentation. Shown in Tab. 3.6, we compare the Window Resize augmentation to the two cropping methods used, and a strategy combining the two. In Tab. 3.6, augmentation methods include WRS (Window Resize to 900×1500), RSC (Random Safe Crop to square crop size), RFC (Random Focused Crop to square crop size), RFC2 (variant of RFC explained

	Detection		Re-id				GFN	
GFN Objective	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
None	96.0	93.6	58.6	88.7	-	-	-	-
Scene-Only	96.0	93.4	56.5	91.9	-0.9	+2.8	16.1	73.3
Base Batch	95.7	93.1	53.9	86.6	-2.6	-2.0	23.8	58.4
Base Proto	96.0	93.6	55.0	86.2	-3.0	-2.7	22.9	57.8
Comb. Batch	96.2	93.6	59.5	92.2	+1.1	+2.9	20.5	78.8
Comb. Proto†	96.0	93.4	58.8	92.3	+1.1	+3.5	20.4	78.5

Table 3.4: Comparison of different options for the GFN optimization objective. “None” does not use the GFN, Scene-Only uses the objective in Sec. 3.3.2.3, Base uses the baseline objective in Sec. 3.3.2.1, Combined (Comb.) uses the query-scene objective in Sec. 3.3.2.2, Batch indicates that batch query embeddings are used, Proto indicates that prototype query embeddings are used. Baseline model is marked with †, final model is highlighted gray.

	Detection		Re-id				GFN	
Model	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
RN50 NAE-FCS	97.6	93.5	50.3	89.4	0.0	+3.5	16.7	78.5
RN50 RCNN-SCS	96.0	93.1	51.1	90.6	+0.1	+3.8	16.3	78.0
CNB NAE-FCS	97.9	94.9	58.7	91.4	+1.3	+3.4	21.0	78.9
CNB RCNN-SCS†	96.0	93.4	58.8	92.3	+1.1	+3.5	20.4	78.5

Table 3.5: Comparison of model backbone (RN50=ResNet50, CNB=ConvNeXt Base), NAE vs. R-CNN head for the second detector stage, and first (stage) classifier score (FCS) or second (stage) classifier score (SCS) used at inference time. Baseline model is marked with †, final model is highlighted gray.

in Sec. 3.3.1), and RSC+RFC(2) which performs either cropping method randomly with equal probability. The baseline model is marked with †, and the final model is highlighted gray.

We find that the Window Resize method achieves comparable re-id performance with other methods, but much lower detection performance. This may be attributed to the regularizing effect of random cropping for detector training.

In addition, we find that Random Safe Cropping alone results in better detection performance than Random Focused Cropping alone, but worse re-id performance. This shows that the regularizing effect of random crops that may be in the wrong scale is more important for detection, and having features in the target scene scale is more important for re-id.

	Detection		Re-id				GFN	
Method	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
WRS	89.3	87.7	57.3	91.1	+0.9	+4.7	19.6	78.3
RSC	95.9	93.1	55.8	91.0	+0.7	+3.7	18.5	77.6
RFC	95.0	92.7	58.4	91.2	+1.4	+3.4	20.8	77.8
RFC2	95.4	93.1	58.2	91.1	+1.4	+3.8	21.1	78.4
RSC+RFC†	96.0	93.4	58.8	92.3	+1.1	+3.5	20.4	78.5
RSC+RFC2	96.1	93.8	58.7	92.3	+1.3	+3.3	20.8	78.9

Crop Size	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
256×256	95.3	91.9	51.4	90.1	0.1	3.3	16.7	78.0
384×384	96.3	93.6	56.7	92.0	0.6	3.0	19.6	79.2
512×512†	96.0	93.4	58.8	92.3	1.1	3.5	20.4	78.5
640×640	95.3	92.9	59.6	92.3	1.4	3.4	21.8	79.6

Table 3.6: Comparison of image augmentation methods (top), and image crop sizes (bottom). Augmentation methods include WRS (Window Resize to 900×1500), RSC (Random Safe Crop to square crop size), RFC (Random Focused Crop to square crop size), RFC2 (variant of RFC explained in Sec. 3.3.1), and RSC+RFC(2) which performs either cropping method randomly with equal probability. Baseline model is marked with †, final model is highlighted gray.

Combining the two results in better performance than either alone for both detection and re-id.

Scene Pooling Size and Embedding Dimension. We analyze choices for the RoI Align pooling size for the scene embedding head, and choices for the embedding dimension for both the query and scene embedding heads. Comparisons are shown in Tab. 3.7.

GFN performance increases nearly-monotonically with scene pooling size, with diminishing returns for GFN score-weighted re-id performance. We also note that larger scene pooling size results in a significant increase in memory consumption, so we use 56×56 by default, which captures most of the performance gain, with some memory savings.

It is clear that the scene pooling size should be larger than the query pooling size to ensure that all person information in a scene is adequately captured. The relationship between person box size distribution vs. scene size, with the ratio of respective pooling sizes could be further investigated.

For the embedding dimension, performance also increases nearly-monotonically with size, for both re-id and the GFN-only stats. Although there are diminishing returns in performance, similar to the scene pooling size, we choose the relatively large value of 2,048 because it results in little additional memory consumption or compute time.

	Detection		Re-id				GFN	
Pool Size	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
14×14	96.1	93.5	58.1	91.6	+0.1	+3.3	18.2	77.9
28×28	95.9	93.4	58.5	92.3	+0.7	+3.6	19.7	79.2
56×56†	96.0	93.4	58.8	92.3	+1.1	+3.5	20.4	78.5
112×112	96.1	93.6	58.8	92.4	+1.2	+3.6	22.1	79.8
Emb Dim	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
128	96.1	93.6	58.0	91.6	0.7	3.8	19.6	77.9
256	95.9	93.4	58.2	92.0	1.0	4.3	20.1	78.3
512	96.1	93.5	58.7	91.8	1.0	4.0	20.0	77.6
1024	96.2	93.6	59.3	92.2	1.1	3.5	20.0	78.0
2048†	96.0	93.4	58.8	92.3	1.1	3.5	20.4	78.5

Table 3.7: Comparison of pooling sizes for the RoI Align block used to compute scene embeddings (top) and comparison of the embedding dimension used for both query and scene embeddings (bottom). Baseline model is marked with †, final model is highlighted gray.

GFN Sampling. We analyze choices for the GFN sampling procedure, with comparisons shown in Tab. 3.8. Critically, we find that all sampling options with the LUT are better than not using the LUT at all, as shown by both the large increase in GFN stats, and the contribution of GFN score-weighting to re-id stats. This is expected but important, because it shows that batch-only query-scene comparisons are insufficient, as they usually just compare a query to the scene it is present in due to sampling constraints. Using LUT comparisons results in better performance despite no gradients flowing through the LUT.

Among sampling mechanisms that use the LUT, results for GFN score-weighted re-id stats were relatively similar, and more trials with more samples per trial are likely needed to distinguish a standout method.

3.4.5 Filtering Analysis

GFN Score Threshold. We consider selection of the GFN score threshold value to use for filtering out gallery scenes during retrieval. In Fig. 3.8, we show histograms of GFN scores for both CUHK-SYSU and PRW, delineated by whether the score corresponds to a matching (Match) or non-matching (Diff) query-gallery pair. We note that the CUHK-SYSU score clusters are significantly more separable than the PRW score clusters, showing that the GFN as a scene-filtering mechanism works better on the CUHK-SYSU dataset. This is because there is much greater visual variability in scenes from CUHK-SYSU than from PRW, making it easier to learn which scenes do or do not contain a given query.

	Detection		Re-id				GFN	
Sampling	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
No LUT	96.2	93.7	57.5	90.8	-0.3	+2.1	13.3	72.8
P1N0	96.1	93.6	59.5	91.9	+1.3	+2.4	21.0	78.7
P1N1†	96.0	93.4	58.8	92.3	+1.1	+3.5	20.4	78.5
P2N0	96.2	93.7	59.1	91.9	+1.2	+3.6	20.9	79.5
P2N1	96.0	93.6	59.1	91.7	+1.2	+3.4	21.1	79.5

Table 3.8: Comparison of different sampling options for optimization of the GFN. $PxNy$ indicates that x positive scenes and y hard negative scenes are sampled for each person in the batch. No LUT means we use only batch query and scene embeddings, and no LUT is used. Baseline model is marked with †, final model is highlighted gray.

We introduce another metric to help analyze computation savings from the filtering operation: the fraction of negative gallery scenes which can be filtered out (negative predictive value) when using a threshold which keeps 99% of positive gallery scenes (recall). For the histograms shown, this value is 91.4% for CUHK-SYSU, and only 11.5% for PRW.

In short, this is because there is greater variation in scene appearance in CUHK-SYSU than PRW. This results in most query-gallery comparisons for CUHK-SYSU evaluation occurring between scenes from clearly different environments, *e.g.*, two different movies. While the GFN score-weighting improves performance for both same-camera and cross-camera retrieval, shown in Tab. 3.3, query-scene scores used for hard thresholding may be less discriminative for nearly-identical scenes as in PRW vs. CUHK-SYSU, shown in Fig. 3.8. Still, the GFN top-1 score for the final PRW model was 78.4%, meaning that 78.4% of queries resulted in the correct gallery scene being ranked first using only the GFN score.

Compute Cost. In Tab. 3.9, we show the breakdown of percent time spent on shared computation, GFN-only computation, and detector-only computation. Since most computation time ($\sim 60\%$) is spent on detection, with only ($\sim 5\%$) of time spent on GFN-related tasks, there is a large cost savings from using the GFN to avoid detection by filtering gallery scenes. Exactly how much time is saved in practice depends on the relative number of queries vs. the gallery size, and how densely populated the gallery scenes are with persons of interest.

To give an understanding of compute savings for a single query, we show some example calculations using the conservative recall requirement of 99%. For CUHK-SYSU, we have 99.9% of gallery scenes negative, 91.4% of negative gallery scenes filtered, and 61.0% of time spent doing detection on gallery scenes, resulting in 55.7% computation saved using the GFN compared to the same model without the GFN. For PRW, the same calculation yields 6.6% computation saved using the GFN.

We demonstrate a further breakdown in Tab. 3.10, where we show how much time is saved (TS) by using the GFN for different levels of recall. We show that for 95% recall on

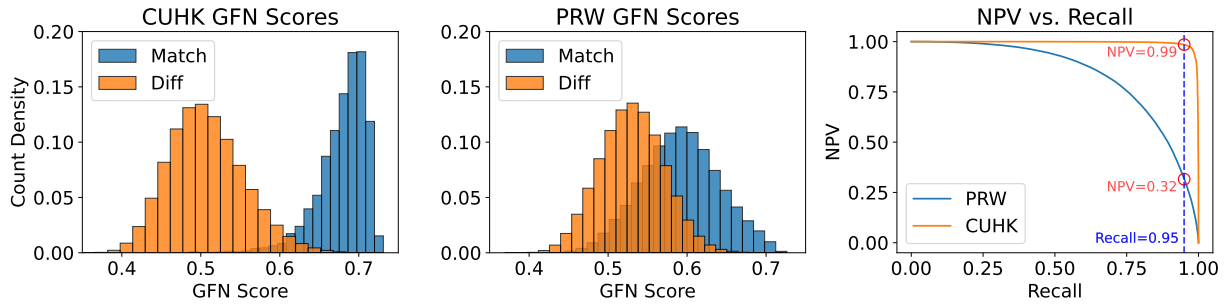


Figure 3.8: GFN score histograms for the CUHK-SYSU (left) and PRW test sets (middle). Matches and non-matches (Diffs) are shown for queries in the gallery size 4,000 set for CUHK-SYSU, and the full gallery for PRW (6,111 scenes). In the plot on the right, we show negative predictive value (NPV) vs. recall for each dataset, based on the score histograms.

	Shared		GFN		Detection	
	Backbone	Query Emb.	Scene Emb.	GFN Scores	RPN	R-CNN($\times 2$)
CUHK Time (%)	33.7	<0.1	5.3	<0.1	19.2	41.8
	33.7		5.3		61.0	
PRW Time (%)	36.9	<0.1	5.3	<0.1	16.1	41.7
	36.9		5.3		57.8	

Table 3.9: Percent computation time averaged per query of shared feature extraction, GFN, and detection on the CUHK-SYSU (gallery size 4,000) and PRW (gallery size full) test sets.

CUHK-SYSU, 57.8% time is saved by filtering scenes with the GFN. For PRW, this value is lower, at 13.6%, as expected from Fig. 3.8 (middle), but non-negligible. We also show in Fig. 3.8 (right) negative predictive value (NPV) vs. recall for the two datasets, making it clear that CUHK-SYSU can filter significantly more scenes without taking a hit in performance.

3.4.6 Qualitative Analysis

Qualitative examples are shown for both CUHK-SYSU and PRW in Fig. 3.9. All examples show cases where the baseline model top-1 match is incorrect, but the GFN-modified match for the same example is correct. We highlight examples where global scene context has an obvious vs. a more subtle impact, and where the query and scene camera ID are the same or different: In the top-left and middle-left, subtle contextual clues (formal wear) help correct the predicted box. In the bottom-left, an obvious contextual clue (interior of same building)

CUHK Time Saved (TS)

Recall (%)	50	75	95	99	100
<i>CUHK Time Saved</i>					
NPV (%)	100.0	99.8	98.5	91.2	0.0
mAP (%)	47.9	72.2	91.6	95.4	96.4
TS (%)	58.8	58.7	57.8	53.1	0.0
<i>PRW Time Saved</i>					
NPV (%)	93.6	75.5	31.6	11.5	0.0
mAP (%)	29.1	43.8	55.4	57.7	58.3
TS (%)	51.4	40.3	13.6	1.4	0.0

Table 3.10: Time saved (TS) from using the GFN for scene filtering on the CUHK and PRW datasets, for different recall requirements. Negative predictive value (NPV) and mAP are also shown to give a holistic view of performance implications at each recall setting.

corrects the prediction, despite a 180° change in viewpoint of the person. In the top-right, the false positive and correct match look nearly identical, and the correct box is from the same camera view. In the middle-right, the false positive and correct match have the same shirt and hairstyle, and the correct box is from a different camera view. In the lower-right, the false positive appears to be a mistake in the ground truth (should be true positive), but the GFN “helped” by up-weighting a more contextually similar scene.

3.4.7 Feature Visualization

In Fig. 3.10, we show pooled features for the backbone `conv4` layer and GFN `conv5` layer, which is green in Fig. 3.5. These features are shown for two separate scenes in CUHK-SYSU. The feature visualization shows that `conv4` focuses on people, while `conv5` focuses on scene context. This is not surprising, because the `conv4` layer comes from the backbone, and is responsible for producing high quality person embeddings, where as the GFN `conv5` layer is optimized just for the GFN, and naturally learns to focus on scene features to produce an accurate scene representation. Since features from the two modules are combined to produce the final scene embedding, the scene embedding is able to accurately represent both person and context content for a given scene.



Figure 3.9: Retrieval examples (CUHK-SYSU left, PRW right) from the baseline model where application of the GFN score corrected the top-1 result. The query box is shown in yellow, a false positive gallery match in red, and a true positive gallery match in blue. In each scene, the white box in the lower right duplicates the person of interest for easier comparison between scenes.

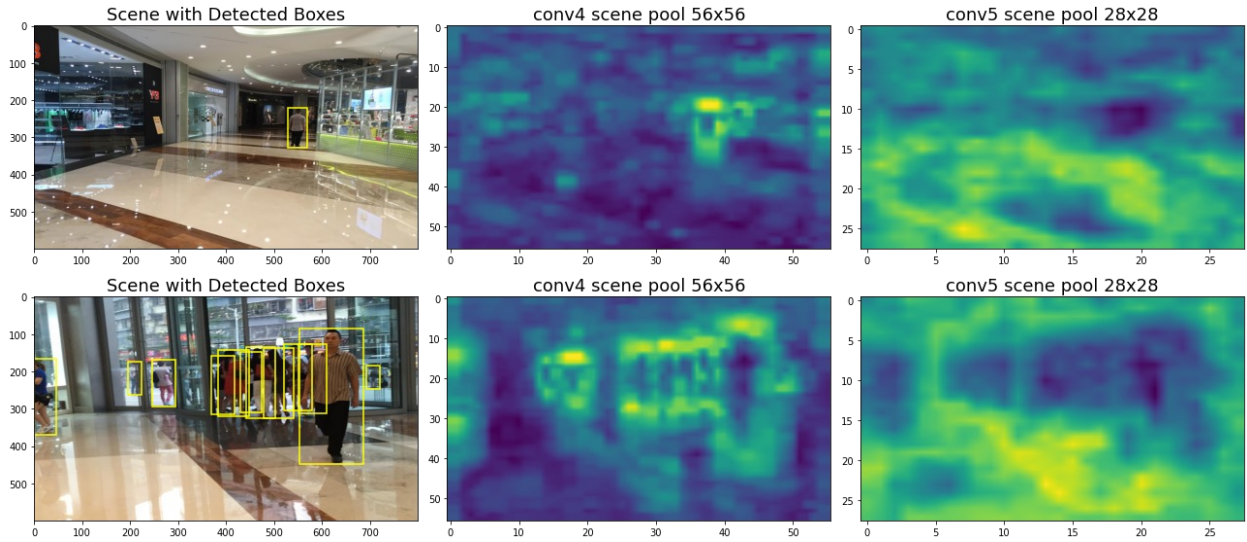


Figure 3.10: Pooled scene features from the (backbone) conv4 and (GFN) conv5 layers show that conv4 focuses on people, while conv5 focuses on scene context.

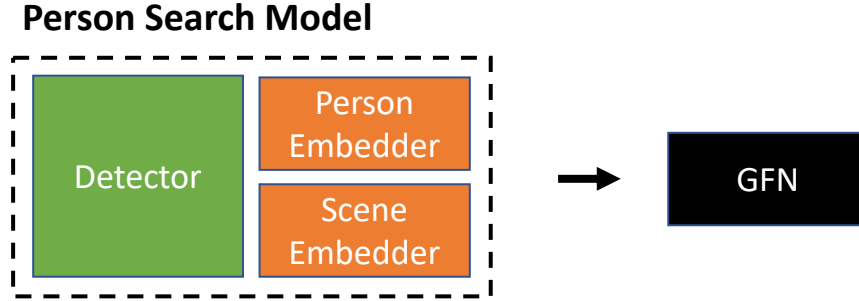


Figure 3.11: Diagram showing that any system with modules for detection, person embedding, and scene embedding can be combined with the GFN to perform scene filtering and score weighting.

3.5 Discussion

We describe the one-step vs. two-step LIR modularization, which splits LIR into separate tasks for detection and retrieval, and compare a range of implications for choosing between these two model types.

In addition, we describe and demonstrate a novel modularization of LIR called the Gallery Filter Network, which improves accuracy and efficiency of person search models. We show that the GFN can efficiently filter gallery scenes under certain conditions, and that it benefits scoring for detects in scenes which are not filtered. We show that the GFN is robust under a range of different conditions by testing on different retrieval sets, including cross-camera, occluded, and low-resolution scenarios. In addition, we show that the benefit given by GFN score-weighting increases as gallery size increases.

Separately, we develop the base SeqNeXt person search model, which has significant performance gains over the original SeqNet model. We offer a corresponding training recipe to train efficiently with improved regularization, using an aggressive cropping strategy. Taken together, the SeqNeXt+GFN combination yields a significant improvement over other state-of-the-art methods. Finally, we note that the GFN is not specific to SeqNeXt, and can be easily combined with other person search models. This is shown in Fig. 3.11: any model with separate modules for detection, person embedding, and scene embedding can be used with the GFN. Further still, the GFN can extend to other identity types beyond persons, such as vehicles, animals, and generic object instances, given the appropriate data.

Comparison with CBGM The GFN module is similar to the Context Bipartite Graph Matching (CBGM) method from [111] in that both methods use context from the query and gallery scenes to improve prediction ranking, although the GFN is used at inference-time only, and does not need to be trained. CBGM is more explicit, in that it directly attempts to match detected person boxes in the query and gallery scenes, at the expense of requiring sensitive hyperparameters: the number of boxes to use from each scene for the matching. The authors

found that very different values for these parameters were optimal for the CUHK-SYSU vs. PRW datasets, and did not provide a clear methodology for their selection besides test set performance. In contrast, we use the exact same GFN configuration for both datasets during training and inference, selected separately based on validation data, and found it to robustly improve performance for both.

Chapter 4

Algorithmic Efficiency

In this chapter, we discuss algorithmic efficiency in LIR by comparing different families of object detectors, and comparing query-centric vs. object-centric approaches. We consider time complexity and actual costs of different approaches, in addition to training time and performance metrics.

The outline of this chapter is as follows: first, we compare efficiency of different detector types in Sec. 4.1, then we compare query-centric and object-centric models in Sec. 4.2, and the majority of the chapter is devoted to discussion of methods and experiments of our novel model, the **Swap Path Net (SPNet)**, in Sec. 4.3 and Sec. 4.4 respectively. SPNet can swap between query-centric and object-centric pathways, while using the same weights. We close with a discussion in Sec. 4.5.

We carefully ensure reproducibility by providing the code, model configs, and installation instructions required to reproduce all experiments, which can be accessed in the GitHub repository¹.

4.1 Comparing Detector Efficiency

We begin with a comparison of time complexity and training time of the three detector types introduced in Chapter 2. This is critical for understanding which detector is the best choice for LIR. We study Faster R-CNN in Sec. 4.1.1, DETR in Sec. 4.1.2, and Pix2Seq in Sec. 4.1.3, followed by a comparative cost analysis in Sec. 4.1.4. We use the function T for time complexity.

In order to make the comparison fair, we make a few simplifications and assumptions, which will hold when replacing with other standard components without loss of generality:

- All models use the same feature backbone, which has an output feature map of size $O(HWD)$, for an image with size $H \times W \times C$ and embedding dimension D , where H and W stand for height and width and C stands for the number of channels for the

¹<https://github.com/LukeJaffe/spnet>

image. We omit the variable D from the analysis, to focus mainly on the effect of image size paired with each detector.

- We make the substitution $L = HW$, since H, W are always paired.
- The feature backbone has fixed inference time c_{bb} , and we do not consider it for the analysis.
- The number of queries for DETR is the same as the max number of detections for Pix2Seq and Faster R-CNN. We call this variable B for boxes. Note that while $B \ll L$, typical constant factors bring terms containing them closer together for some model types, which is why we include $O(B+L)$ terms in our time complexity analysis.

4.1.1 Faster R-CNN

For Faster R-CNN, the time complexity depends linearly on the sequence length. The cost of the RPN depends on L , and the cost of the box head depends on B . Though the box head cost is typically much greater per proposal than the RPN head, the time complexity is technically dominated by L since $B \ll L$ for Faster R-CNN.

$$T(L) = O(L)$$

4.1.2 DETR

For DETR, the time complexity depends on the transformer encoder and decoder. The cost of self-attention on L image features is simply L^2 , and the cost of cross-attention between queries and image features is LB .

$$T(L, B) = O(L^2 + LB)$$

4.1.3 Pix2Seq

For Pix2Seq, the cost of the decoder is dependent on the square of the max number of boxes B , but this cost is even more significant in practice, because it must be paid sequentially via the sequential decoding.

$$T(L, B) = O(L^2 + LB^2)$$

4.1.4 Cost Analysis

In Tab. 4.1, we compare training and inference costs vs. performance for three detectors on the MS-COCO 2017 dataset. We include data for two Faster R-CNN training schedules, 3x

Model	Time Complexity	Training Epochs	Parameters	FPS	Detector AP
Faster R-CNN (3x)	$O(L)$	36	42M	26	40.2
Faster R-CNN (9x)	$O(L)$	109	42M	26	42.0
DETR	$O(L^2 + LB)$	500	41M	28	42.0
Pix2Seq	$O(L^2 + LB^2)$	300	37M	3.5	43.0

Table 4.1: Comparison of training and inference cost vs. performance for three detectors. Values from Sparse R-CNN paper [66]. Models evaluated on a NVIDIA Tesla V100 GPU to determine FPS (frames per second).

which has 36 training epochs, and 9x which has 109 training epochs. Each training epoch is a single pass through the full dataset, so the comparison is equivalent between models.

The most important takeaways from Tab. 4.1 are that: 1) all models have roughly the same parameter count and detector performance (AP), 2) It is much faster to train Faster R-CNN than the other models, 3) DETR has similar inference speed to Faster R-CNN, and 4) Pix2Seq has very slow inference speed, as expected from the sequential decoding.

We note that there have been numerous improvements to DETR and other set-based models, such as those discussed in Chapter 2, and that these improvements reduce the training time to Faster R-CNN levels, while other parameters in the table remain similar. The main innovation these variants of DETR introduced is some version of spatially-referenced features. Mostly, they force DETR query-feature cross attention to attend to specific regions of the image feature map, similar to the inductive bias of traditional anchors.

The true cost of Pix2Seq is masked by the time complexity, since the problem is that the cost must be paid sequentially in the sequential decoding, not merely that more FLOPs are required. The authors of Pix2Seq have suggested numerous methods of speeding up inference, including the multi-query trick, quantization, distillation, and hardware optimization. Still, there is a fundamental barrier for Pix2Seq in decoding scenes with many objects, so it is better suited to tasks such as visual grounding, *e.g.* “where is the horse in this image”? This type of task is possible using later models which inherit from Pix2Seq such as Florence-2 [68].

We conclude that Faster R-CNN, and anchor-based methods in general, compare favorable in performance and efficiency with set-based and sequence-based counterparts. This is also supported by the fact that many changes to DETR and other set-based models seek to replicate the spatial-referenced property of anchor-based models, as discussed in Chapter 2.

4.2 Query-Centric and Object-Centric Models

Most person search models implement an *object-centric* (OC) approach [110]–[115], in which we detect persons independent of any query, then afterwards compare queries to detected persons.

	Time Complexity		
Model	Detection	Embedding	Comparison
Object-centric	$O(G)$	$O(GB)$	$O(GBQ)$
Query-centric	$O(GQ)$	$O(GQ)$	$O(GQ^2)$

Table 4.2: Time complexity of QC vs. OC models based in the number of queries Q , the number of gallery scenes G , and the number of boxes kept per scene after filtering B .

By contrast, *query-centric* (QC) person search models [118]–[121] use query person images during the detection process, producing proposals tailored to each query, but significantly increasing time complexity of inference computation.

In query-centric detection, we detect matches to a query object in a gallery scene, vs. standard *object-centric detection*, where we detect a predetermined set of objects in a scene. **Inference Time Complexity.** Time complexity of inference using query-centric and object-centric models is best understood based on the number of queries Q , the number of gallery scenes G , and the number of boxes kept, B , per scene after filtering. The comparison is shown in Tab. 4.2, which uses the assumption that one or zero boxes are kept per scene with a query-centric model: one simply selects the top-scoring box since there can only be one or zero matches to the query.

In addition, we must consider the cost of detection c_{det} , cost of computing embeddings c_{emb} , and the cost of embeddings comparison c_{cmp} which roughly has the relationship $c_{\text{det}} \gg c_{\text{emb}} \gg c_{\text{cmp}}$. The upshot is that is that the cost of detection is much greater for query-centric models since it has a dependency on Q , and therefore LIR in general is much less efficient for large Q . Because of this, query-centric detection and LIR should only be done when there is another compelling reason for them, such as: 1) significantly improved performance, 2) tasks that can only be done in a query-centric context such as Referring Expression Comprehension (REC), Referring Expression Segmentation (RES), 3) tasks with small or constant Q such as zero-shot detection as in UNINEXT [8], or 4) beneficial pre-training, as we will show in Chapter 5.

4.3 Methods

In this section, we detail the SPNet, including both the details of its architecture in Sec. 4.3.1 and training objectives in Sec. 4.3.2. As mentioned, SPNet can swap between query-centric and object-centric pathways, while using the same weights. While we will focus on using each pathway separately in this chapter, we will show how the model benefits from pre-training in one mode and fine-tuning in the other mode in Chapter 5.

4.3.1 SPNet Description

SPNet Forward Pass Overview. A diagram detailing components of SPNet is shown in Fig. 4.1, with the full SPNet architecture shown in the center, details about the Box Embedder shown in the top left, the Cascade Loop shown in the top right, and the SP Block shown on the bottom. Model blocks are outlined with dashed lines, while sub-blocks are within these blocks and outlined with solid lines. The subscripts q, d, a stand for “query”, “detection”, and “anchor” respectively. The superscript “reid” means the embedding or loss is used for re-id, and the superscript “det” means the embedding is used for detection. SPNet takes Gallery images as input, and outputs detected boxes b_d with corresponding class logits z_d and re-id embeddings x_d^{reid} . SPNet also takes a Query scene as input with a corresponding ground truth person box b_q , and outputs the re-id embedding for that box x_q^{reid} . The query re-id embedding x_q^{reid} is compared to gallery re-id embeddings x_d^{reid} via cosine similarity to produce re-id scores, which are used to rank predictions for retrieval. Image features (C_3 - C_5) are learned by the backbone and refined by the FPN (P_3 - C_7), and then fed to the Box Embedder, with either query boxes b_q or detected boxes b_d , to learn re-id embeddings $x^{\text{reid}}, x^{\text{det}}$. FPN features are also fed to the Anchor Embedder to learn initial anchor embeddings x_a^0 .

When SPNet uses multiple refinement steps, we call this the Cascade Loop following Cascade R-CNN [19]. In the Cascade Loop, predicted boxes $b_d^{(t)}$ from stage t are fed back into the Box Embedder to produce anchor embeddings $x_a^{(t)}$ for the next step $t + 1$. The outputs $z_a^{(t)}, x_o^{(t)}, z_d^{(t)}$, and $\hat{r}_d^{(t)}$ are also updated during each Cascade Loop step. The detector re-id embeddings x_d^{reid} are computed from the detected boxes coming out of the final Cascade Loop stage.

SPNet Forward Pass Details. More specifically, query and gallery images are fed through a standard vision backbone to produce feature maps from successive `conv` blocks (C_3 - C_5). These features pass through a five-layer Feature Pyramid Network (FPN) [17], following the RetinaNet structure [55]. The Anchor Embedder, a single `conv2d` layer, takes FPN features (P_3 - P_7) at each scale and produces the initial anchor embeddings x_a^0 corresponding to each anchor box. Simultaneously, FPN P_3 features [110] and query boxes are fed to the Box Embedder, which pools a 14×14 feature map using RoI Align [53], and feeds it to a duplicate of the backbone `conv5` block, followed by an additional linear head to produce query embeddings x_q . The Box Embedder can be optionally duplicated to produce separate query embeddings for re-id x_q^{reid} and detection x_q^{det} , or shared *i.e.* $x_q^{\text{reid}} = x_q^{\text{det}}$, shorthanded as x_q . Using a shared Box Embedder has a regularizing effect by pushing x_q to comply with both re-id and detection losses, while duplicating the Box Embedder gives more capacity for each task, which can lead to overfitting. Embeddings x_a , and x_q for QC, then enter the SP Block, where they may pass through either the QC or OC pathway, depicted in Fig. 4.2.

QC Pathway. In the QC pathway, shown in Fig. 4.2a, we compute offset embeddings x_o from the *Offset Layer*, by simply subtracting each anchor embedding from the query embedding: $x_o = x_q - x_a$. We show why this is a good choice of f in Sec. 4.3.2, though other f could be explored. Offset embeddings x_o are passed to a *Logits Layer*, explained in Sec. 4.3.2,

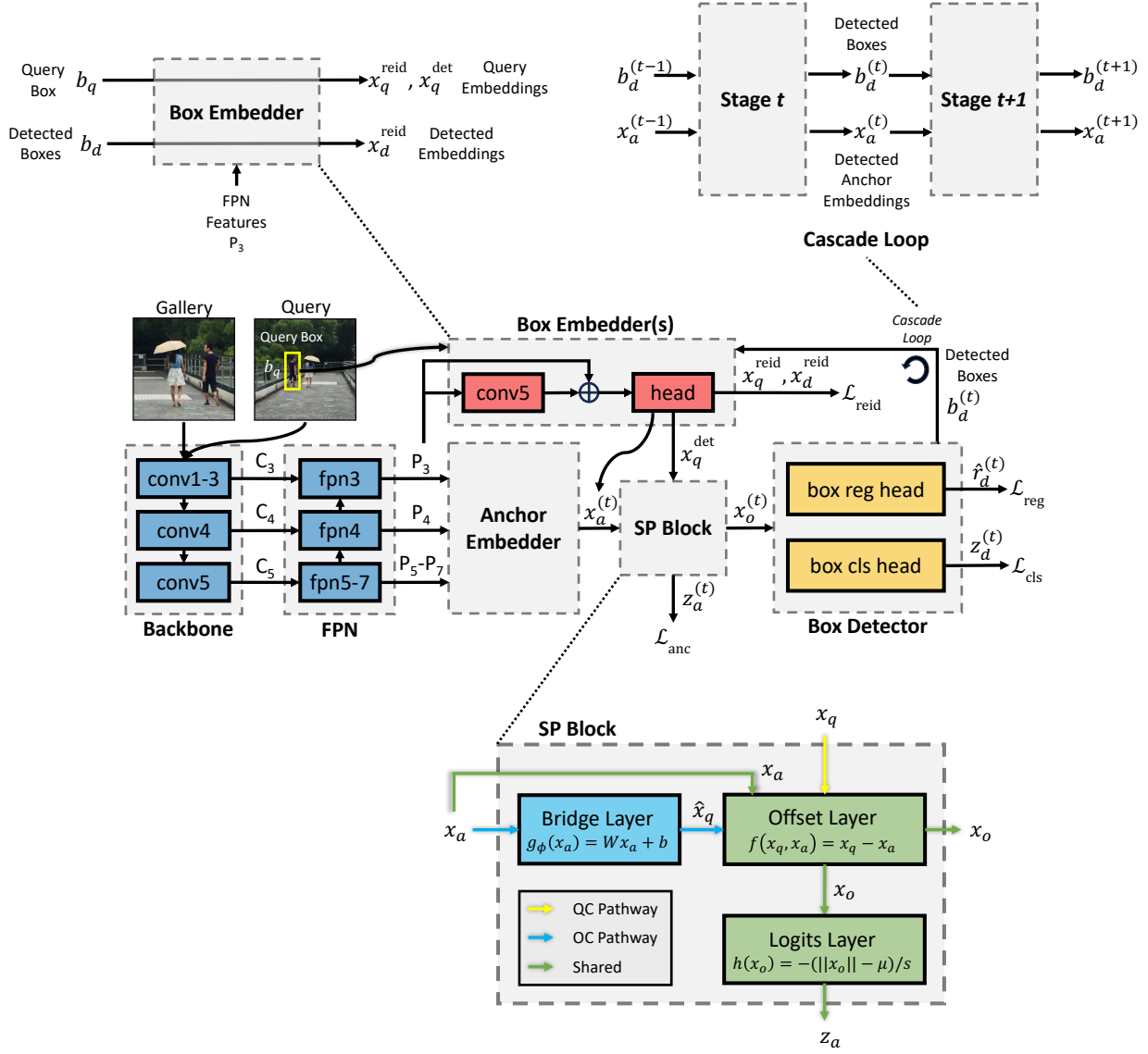


Figure 4.1: Full SPNet architecture is shown in the center, with details about the Box Embedder shown in the top left, the Cascade Loop in the top right, and the SP Block on the bottom.

to produce classifier predictions z_a for each anchor box *i.e.*, asking the question: does this anchor box match the query? These z_a are used similarly to classifier logits in a standard Faster R-CNN Regional Proposal Network (RPN) [20]: they filter a large number of anchors *e.g.*, $> 100,000$ down to $< 1,000$, which are then refined to produce more accurate boxes.

OC Pathway. In the OC pathway, shown in Fig. 4.2b, we do not have knowledge of any query embedding x_q , so we instead compute a pseudo-query embedding $\hat{x}_q$ using the *Bridge Layer*, by default a single affine layer $g_\phi(x_a) = Wx_a + b$ with learnable parameters $\phi = \{W, b\}$. By mimicking the QC pathway instead of predicting localization offsets and logits directly from x_a , we improve transfer performance between the two pathways. Crucially, the layer g_ϕ is the only difference between the QC and OC models, meaning that all SPNet weights, aside from ϕ , can be trained in one mode, and ported to the other, or vice versa.

Box Prediction. Both pathways output offset embeddings x_o that are passed to a separate Box Detector module, which performs the box refinement. The **box reg head** is a 4-layer MLP with input x_o which predicts offsets $\hat{r}_d$ between anchor boxes and matching ground truth boxes. The **box cls head** uses the same Logits Layer as the SP Block to produce class logits z_d . Predicted boxes b_d are fed back to the Box Embedder (same as that used to produce x_q^{reid}) to produce predicted re-id embeddings x_d^{reid} . The x_q^{reid} from queries are compared to x_d^{reid} from gallery images for ranking.

Cascade Loop. We can iteratively refine box accuracy and scores by repeating the same basic prediction structure in a loop, referred to as the *Cascade Loop*, originating from the Cascade R-CNN detector [19], and also done in the SeqNet [111] and SeqNeXt [3] person search models. To do so, we take the predicted boxes and embeddings from iteration t , and use them in place of the anchor boxes and anchor embeddings x_a in iteration $t + 1$. The SP Block, Box Detector, and Box Embedder are all duplicated for each round, meaning that we use modules with the same architecture, but do not share weights of the same module between rounds.

4.3.2 Losses

In this subsection, we detail the classification, box regression, re-id, and feature consistency losses used by SPNet.

Classifier Losses. For the offset embeddings x_o shown in Fig. 4.1, it is critical to define a loss function which enforces a consistent relationship between query embeddings x_q and anchor embeddings x_a , while performing the classification regression. The goal is to regress query-anchor matches to 1 and non-matches to 0. For this task, we model the anchor loss after the Norm-Aware Embedding concept from [108], and the Focal Loss from [55].

Recall the offset embedding is defined as the difference between query and anchor embeddings: $x_o = x_q - x_a$ with $x_o \in \mathbb{R}^d$. Assign $\omega = \|x_o\|$. In the Norm-Aware Embedding work [108], classification logits are computed from embedding norms using a Batch Norm layer [205]. To avoid use of unstable batch statistics, we instead compute a fixed rescaling by modeling $x_o \sim \mathcal{N}(0, I_d)$, where I_d is the $d \times d$ identity matrix, which implies $\omega \sim \text{Chi}(d)$. Then we can standardize ω using the mean, μ , and variance, s^2 , of the Chi distribution:

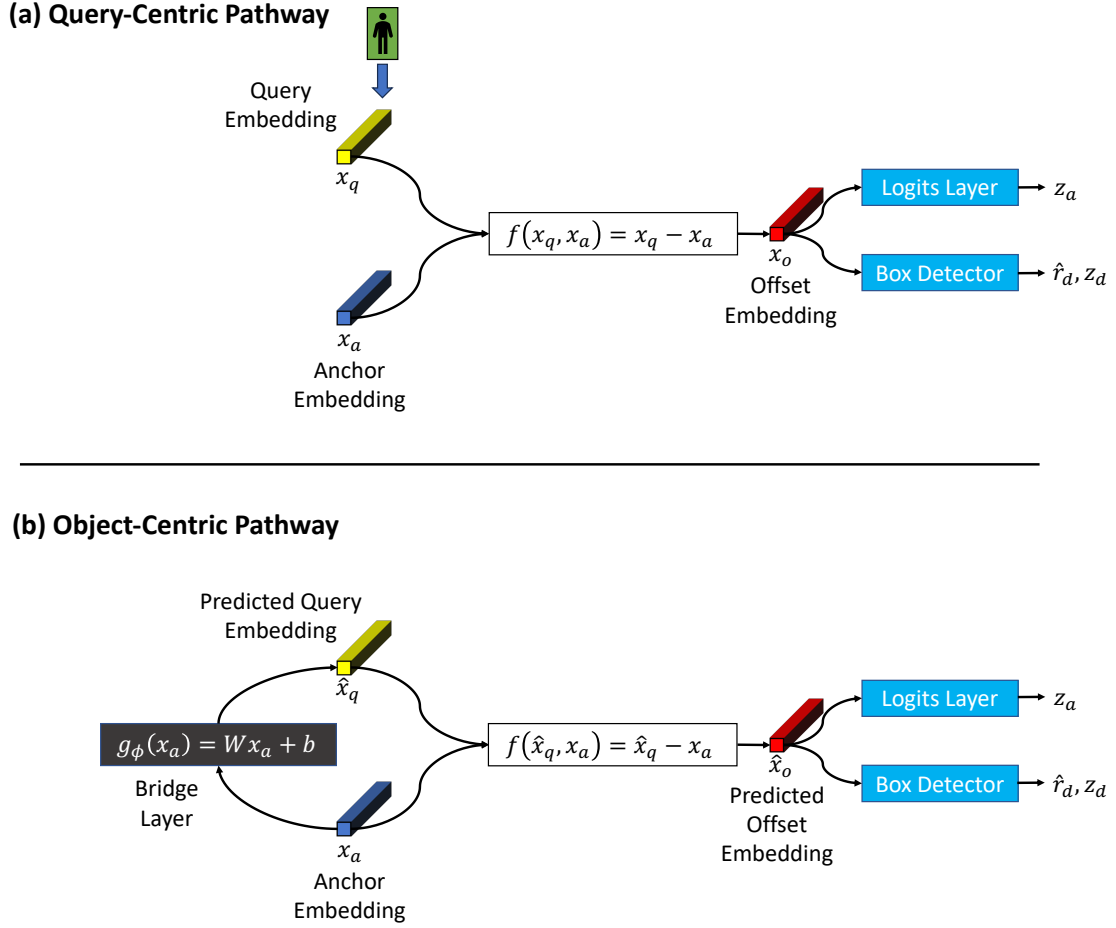


Figure 4.2: Query-centric and object-centric pathways of the Swap Path Block. The two pathways are nearly identical, but the query-centric pathway assumes input of a query person image crop, whereas the object-centric pathway predicts a pseudo-query embedding using the Bridge Layer.

$$\tilde{z} = \frac{w - \mu}{s}, \quad z = -\tilde{z} \quad (4.1)$$

with

$$\mu = \sqrt{2} \frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)}, \quad s = \sqrt{d - \mu^2} \quad (4.2)$$

where Γ is the gamma function.

Finally, we compute class probabilities $p = \sigma(z)$, using the logistic sigmoid, denoted by σ . We note for $d \gg 1$, $\mu \approx \sqrt{d - \frac{1}{2}}$ and $s \approx \sqrt{\frac{1}{2}}$.

For anchor logits z_a , we compute a loss averaged across all anchors $\mathcal{L}_{\text{anc}}$, applying the Focal Loss defined in [55], with hyperparameters $\alpha_{\text{FL}} = 0.5, \gamma_{\text{FL}} = 1$. For cascade layer class logits z_d , we compute a loss $\mathcal{L}_{\text{cls}}$ averaged across predicted boxes using the unweighted Cross Entropy Loss.

The *FixedNorm* transformation in Eq. (4.1) has two key properties: 1) For $x_o \sim \mathcal{N}(0, I_d)$, $\mathbb{E}[z] = 0, \text{Var}[z] = 1$, independent of d . In addition, $\lim_{d \rightarrow \infty} z \sim \mathcal{N}(0, 1)$ due to the Central Limit Theorem. Although the distribution of x_o is rarely unit-normal and changes during optimization, this framing gives us a reasonable choice of shifting and scaling parameters which works well in practice, does not require learnable parameters or hyperparameters, and holds independent of d .

2) For x_q similar to x_a i.e., $\|x_o\|$ small, z is larger, and for x_q dissimilar to x_a i.e., $\|x_o\|$ large, z is smaller. Intuitively, when a query embedding matches a given anchor embedding according to box IoU overlap, we want the two to be more similar, and when they do not match, we want them to be more different. The relationship is also critical to co-optimizing with the re-id loss on x_q and has a nice relationship with the box regression loss.

An ablation comparing use of BatchNorm vs. our FixedNorm, and with vs. without negation of $\tilde{z}$, is included in Sec. 4.4.

Box Regression Loss. For bounding box offset regression, we use the generalized IoU (GIoU) loss [71], shown as $\mathcal{L}_{\text{reg}}$. This loss has a beneficial interaction with the classifier loss: for matching query and anchor boxes (high IoU/GIoU), the target box regression offsets should be small, corresponding to $\|x_o\|$ small. This allows the model to learn that the magnitude of $\|x_o\|$ correlates directly to the size of predicted box offsets.

Re-id Loss. For the re-id loss, we use the standard online instance matching (OIM) loss from [1].

Feature Consistency Loss. We also add an optional feature consistency loss specifically for QC training, which improves alignment between query and anchor embeddings, x_q and x_a . This loss minimizes the mean squared error between a given x_a and the query feature extracted for the same anchor box, which we call x_q^a . In each batch, we randomly sample a set of anchor boxes n_a to extract both x_a and x_q^a .

$$\mathcal{L}_{\text{fc}} = \frac{1}{n_a} \sum_{i=1}^{n_a} \|x_{a_i} - x_q^{a_i}\|_2^2 \quad (4.3)$$

Final Loss. The final loss is simply the unweighted sum of all described losses:

$$\mathcal{L} = \mathcal{L}_{\text{reid}} + \mathcal{L}_{\text{anc}} + \sum_{t=1}^{\mathcal{T}+1} \mathcal{L}_{\text{reg}}^{(t)} + \mathcal{L}_{\text{cls}}^{(t)} \quad (4.4)$$

where $\mathcal{T}$ is the number of cascade stages, with $\mathcal{L}_{\text{fc}}$ added optionally.

4.4 Experiments and Analysis

In this section, we describe the datasets and evaluation mechanism used in Sec. 4.4.1, implementation details of the model and optimization in Sec. 4.4.2, compare query-centric and object-centric training in Sec. 4.4.3, show state-of-the-art results compared to other person search models in Sec. 4.4.4, and study the choice of classifier logits in Sec. 4.4.5.

4.4.1 Datasets and Evaluation

Datasets. We fine-tune models on the two standard person search benchmark datasets: CUHK-SYSU [1] and PRW [100].

The CUHK-SYSU dataset has 18k scenes with 95k annotations, split among 23k boxes for 8.5k known identities, and 72k boxes for unknown identities. The scenes in CUHK-SYSU come from a mixture of handheld device photos taken on city streets, and scenes from TV shows and films. The standard test retrieval scenario for CUHK-SYSU uses 2,900 queries with 100 scenes in each gallery.

The PRW dataset has 12k scenes with 43k annotations, split among 34k boxes for 1k known identities, and 9k boxes for unknown identities. The scenes from PRW come from six fixed cameras installed at the Tsinghua University campus in Beijing. The standard test retrieval scenario for PRW uses 2,057 queries with all 6,112 test scenes in the gallery.

Evaluation. We evaluate all models by measuring fine-tuning performance once on the standard retrieval test scenario for CUHK-SYSU or PRW.

To measure performance, we use standard detection performance metrics of recall@0.5 IoU and average precision@0.5 (AP@0.5) IoU, indicating predictions with an overlap of > 0.5 IoU with a ground truth box are considered matches. For re-id evaluation, we use the standard metrics of mean average precision (mAP) and top-1 accuracy (top-1). We distinguish between re-id measured on ground truth boxes as GT mAP and GT top-1, vs. re-id measured on detected boxes as mAP and top-1. When mAP and top-1 are measured on detected boxes, we consider these to be metrics of *person search*. Measuring ground truth re-id performance is equivalent to using a perfect object detector, and helps us understand re-id performance independent of detector quality.

Config. Name	Cascade Steps	Shared Heads	Re-id Dim
SPNet-S	0	✓	128
SPNet-L	2		2048

Config. Name	Backbone		Post-Backbone		LWF Loss	Image Size
	LR	WD	LR	WD		
Pre-train	1e−5	0	1e−4	1e−3	✓	512
Fine-tune	1e−4	5e−4	1e−4	5e−4		512 or 1024

Table 4.3: SPNet architecture (top) and layer optimization (bottom) hyperparameter configurations. LR = Learning Rate, WD = Weight Decay, LWF = Learning without Forgetting [183].

4.4.2 Implementation Details

Model Configurations. We perform experiments with two variants of the SPNet model: SPNet-S(mall) and SPNet-L(arge), with differences shown in Tab. 4.3. In Tab. 4.3, Post-Backbone refers to any weights after the backbone in Fig. 4.1, and Re-id Dim is the dimension used for re-id embeddings x^{reid} . Unless otherwise stated, SPNet-S uses a ConvNeXt-Tiny [202] backbone, and SPNet-L uses a ConvNeXt-Base backbone. In all model versions, a 4-layer MLP is applied to offset embeddings x_o to produce box regression offsets, while classification logits are computed directly from $\|x_o\|$. The training batch size is 8, with $k = 2$ augmentations per image during pre-training.

Optimization. For all experiments, we train for 30 epochs, using the AdamW optimizer [206], a cosine-annealed schedule, and linear warmup following [206], [207]. For all experiments, the SPNet backbone is initialized using ImageNet-1k classifier weights. For the re-id loss, we use the OIM objective described in Sec. 4.3.2 with temperature 1/30, momentum 0.5, and queue length 5,000 for CUHK and 500 for PRW, the standard settings. We use the Fine-tune config from Tab. 4.3, with image size 512×512 used for SPNet-S, and image size 1024×1024 used for SPNet-L.

Anchor Sampling. To reduce differences between QC and OC pre-training from anchor sampling during detection, we ensure that the same number of anchors are optimized in each batch between QC and OC trials, 2,048 by default. For the OC case, this is calculated as number of images per batch $\times$ number of anchors per image. For the QC case, this is calculated as number of images per batch $\times$ number of query-image pairs $\times$ number of anchors per query-image pair.

Query-image pairs constitute the pairing of a given query embedding x_q with anchor embeddings x_a from a given scene image. With the total number of query-image pairs equal to number of queries $\times$ number of images, this can quickly exceed memory limitations if all

Method	CUHK-SYSU		PRW	
	mAP	top-1	mAP	top-1
QC	80.8	87.7	54.0	86.7
QC-FC	91.1	92.8	54.4	86.2
OC	91.9	93.2	52.7	86.6

Table 4.4: Comparison of QC and OC training for SPNet-S, with optional feature consistency loss (-FC).

pairs are used, so some subsampling procedure is required. We select pairs in the following order up to a fixed number, 32 by default: 1) queries are selected for images which they are not present in, but share a label with a box in the image, 2) queries are selected for images which they are present in, and 3) queries are selected for images which they are neither present in, nor share a label with any boxes in.

Both QC and OC sampling methods attempt to balance the number of positive and negative samples, but there are usually more negative samples in practice due to sparsity of positives and typical hyperparameter selection.

4.4.3 Query-Centric vs. Object-Centric Training

In Tab. 4.4, we compare QC and OC training for SPNet-S. We can see the OC model vastly outperforms the QC model on CUHK-SYSU, and that the QC model is moderately better on PRW. When we add in the feature consistency loss (QC-FC), the gap between QC and OC training is largely closed, with QC-FC mAP on CUHK-SYSU being more than 10% higher than plain QC, and QC-FC remaining close to QC on PRW. This shows that the QC relationship $x_o = x_q - x_a$ holds during training, but that it generalizes poorly in some cases without ensuring that x_q and x_a are extracted from consistent feature spaces. It is likely QC and QC-FC differ much more on the CUHK-SYSU dataset than PRW because there is much greater image variability in CUHK-SYSU, which consists of real photos, movies, TV shows. This creates more incentive for the Box Embedder to diverge significantly from the Anchor Embedder.

4.4.4 Comparison with State-of-the-art

In Tab. 4.5, we compare performance of SPNet with recent state-of-the-art person search models. We show that SPNet-L (OC) bests the previous best SeqNeXt model of Chapter 3 in mAP for the PRW dataset by 1.3%, and is the best or near-best in all metrics. In addition, SPNet-S (QC-FC) is competitive with other QC models, achieving significantly greater mAP on the PRW dataset.

Method	Backbone	CUHK-SYSU		PRW	
		mAP	top-1	mAP	top-1
<i>Query-Centric (QC)</i>					
IGPN [120]	ResNet50	90.3	91.4	47.2	87.0
TCTS [121]	ResNet50	93.9	95.1	46.8	87.5
QEEPS [118]	ResNet50	88.9	89.1	37.1	76.7
QGN [119]	ResNet50	93.7	94.4	46.7	82.9
SPNet-S (QC-FC)	ConvNeXt-T	91.1	92.8	54.4	86.2
<i>Object-Centric (OC)</i>					
IDE [100]	ResNet50	-	-	20.5	48.3
MGTS [101]	VGG16	83.0	83.7	32.6	72.1
CLSA [102]	ResNet50	87.2	88.5	38.7	65.0
RDLR [103]	ResNet50	93.0	94.2	42.9	70.2
OIM [1]	ResNet50	75.5	78.7	21.3	49.4
NPSM [104]	ResNet50	77.9	81.2	24.2	53.1
APNet [105]	ResNet50	88.9	89.3	41.9	81.4
HOIM [106]	ResNet50	89.7	90.8	39.8	80.4
BINet [107]	ResNet50	90.0	90.7	45.3	81.7
NAE+ [108]	ResNet50	92.1	92.9	44.0	81.1
DKD [109]	ResNet50	93.1	94.2	50.5	87.1
AlignPS [110]	ResNet50	94.0	94.5	46.1	82.1
SeqNet+CBGM [111]	ResNet50	94.8	95.7	47.6	87.6
COAT+CBGM [112]	ResNet50	94.8	95.2	54.0	89.1
MHGAM [115]	ResNet50	94.9	95.9	47.9	88.0
PSTR [113]	PVTv2-B2	95.2	96.2	56.5	89.7
SeqNeXt	ConvNeXt-B	96.1	96.5	57.6	89.5
SPNet-L (OC)	ConvNeXt-B	95.9	96.6	58.9	89.7

Table 4.5: Standard performance metrics mAP and top-1 accuracy on the benchmark CUHK-SYSU and PRW datasets are compared for state-of-the-art *QC* and *OC* models. +CBGM indicates using Context Bipartite Graph Matching from [111].

Method	CUHK-SYSU		PRW	
	mAP	top-1	mAP	top-1
BatchNorm Logits (+)	90.4	91.9	49.8	84.3
BatchNorm Logits (-)	91.2	92.0	42.2	82.2
FixedNorm Logits (+)	91.1	92.2	52.6	86.3
FixedNorm Logits (-)	91.9	93.2	52.7	86.6

Table 4.6: Comparison of BatchNorm Logits from [108] to proposed FixedNorm Logits for CUHK-SYSU and PRW datasets, for SPNet-S (OC). (+) vs. (-) indicates whether positive or negative embedding norm was used to compute logits.

4.4.5 Classifier Logits

To validate our FixedNorm logits over the BatchNorm logits from [108], we compare both for the baseline model, with results shown in Tab. 4.6. The FixedNorm method achieves greater performance for all metrics, especially on the PRW dataset, where mAP exceeds the BatchNorm method by more than 10%. This is likely due in part to unbalanced batch statistics, which are dominated by negative samples, unlike the original Norm-Aware Embedding use case. Tab. 4.6 also shows that negating the norm after scaling is beneficial, especially for the FixedNorm logits, validating the rationale discussed in Sec. 4.3.2.

4.5 Discussion

In this chapter, we performed an analysis of detector types, showing that anchor-based detectors such as Faster R-CNN are still viable in the age of set-based and sequence-based detectors.

We proposed and validated Swap Path Net (SPNet), an end-to-end model for person search, which supports query-centric (QC) and object-centric (OC) modes of operation. We showed that the model can perform both QC and OC training using the same set of weights, and that when the QC feature consistency loss is employed, QC and OC modes achieve comparable results. In addition, we showed that SPNet-L (OC) matches or exceeds the SeqNeXt model of Chapter 3 in most person search metrics.

Finally, we showed that QC person search is much less efficient than OC person search, though it achieves comparable performance. In Chapter 5, we will explore how the QC mode of SPNet can also be used for effective weakly-supervised and self-supervised pre-training.

Chapter 5

Data Efficiency

In this chapter, we discuss data efficiency in LIR by way of studying pre-training algorithms for person search, and training in the presence of missing labels. In general, pre-training of models enables fine-tuning for downstream tasks more quickly for a given set of data and generalization requirements. In some cases, pre-training can result in a level of fine-tuning performance that could not otherwise be achieved with the fine-tuning dataset. In fact, pre-training can benefit both few-shot and large-scale fine-tuning, and is considered essential for both in modern training schedules.

The overview of this chapter is as follows: first we discuss pre-training for person search and the motivation for our method in Sec. 5.1, then we cover the details of our pre-training pretext task and corresponding objective functions in Sec. 5.2, and showcase several ablation studies designed to analyze the effect of different training configurations, loss functions, and model architecture choices in Sec. 5.3. We close with a discussion in Sec. 5.4.

5.1 Pre-training for Person Search

In all person search models, it is typical to initialize the feature backbone using weights pre-trained for classification on the *ImageNet-1k* dataset [204]. Only recently have approaches emerged considering pre-training backbones for person search on annotated person data [123], [129], even then only considering cropped bounding boxes from the *LUPerson* [130] or *Market* [80] datasets.

This leaves two key issues unresolved in the pursuit of effective pre-training for person search: 1) current backbone pre-training is done using a pretext task unrelated to the person search task, and 2) all parameters beyond the backbone are randomly initialized. Even pre-training on cropped person bounding boxes is insufficient, because it does not initialize weights for the detection task. Therefore, a model is needed which can simultaneously pre-train for detection and re-id, while achieving domain transfer from a pre-training dataset that is potentially very different from the target dataset.

In this chapter, we develop a novel model which addresses these challenges by pre-training

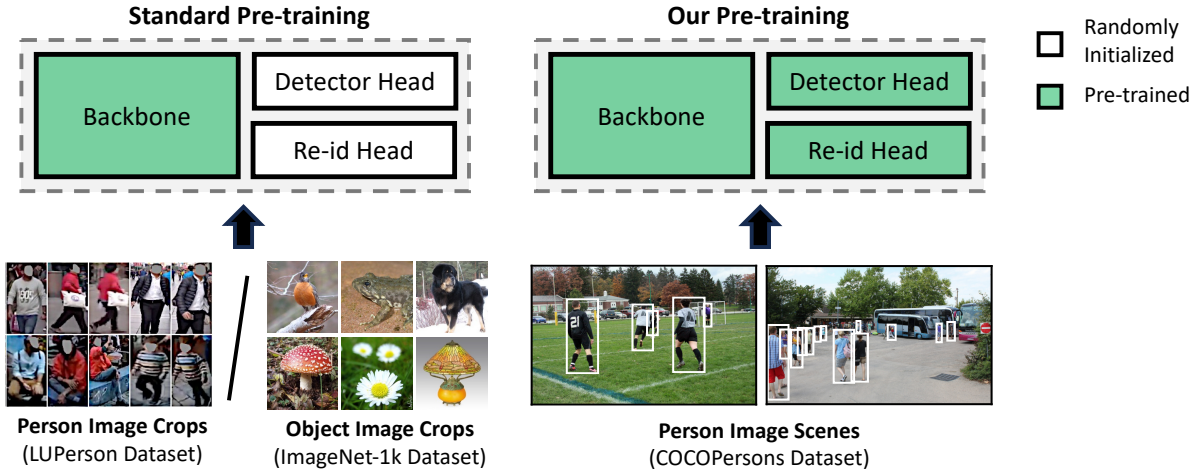


Figure 5.1: Standard person search model pre-training approach (left) vs. our approach (right). (LUPerson images from paper [130])

using the *query-centric detection* pretext task. The query-centric approach is not only resilient to missing annotations, but also can train using arbitrary box annotations, since it operates only on relative similarity between images, and not absolute definitions of what constitutes an “object”. We use the **Swap Path Net (SPNet)**, described in Chapter 4, because it can swap between query-centric and object-centric pathways, while using the same weights. SPNet is capable of pre-training using *weakly-labeled* person bounding boxes, which simply means we do not know whether person bounding boxes in different scenes correspond to the same person or not. SPNet is also capable of fully-self-supervised pre-training, meaning that it can train on scene images with no labels at all.

We visualize our pre-training approach vs. typical backbone-only pre-training in Fig. 5.1: the standard person search model pre-training approach, shown left, pre-trains only backbone weights using either ImageNet-1k classification or person image crops from e.g., LUPerson. Our approach, shown right, in the weakly-supervised case, initializes all model weights, including the backbone, detector head, and re-id head, using full person scenes with multiple annotated persons.

We show that when SPNet is pre-trained in the query-centric mode, and fine-tuned in the object-centric mode, it achieves state-of-the-art performance on the benchmark *CUHK-SYSU* [1] and *Person Re-identification in the Wild (PRW)* [100] person search datasets. In particular, we demonstrate that weakly-supervised pre-training on the *COCOPersons* dataset [64], [195] using our method is far more effective and efficient than recent backbone-only pre-training alternatives for unsupervised person re-id [129], [169], [173]. We also demonstrate self-supervised pre-training on ImageNet1k, and show that both weakly-supervised and self-supervised pre-training can be effective in different circumstances. We carefully ensure reproducibility by providing the code, model configs, and installation instructions required to

reproduce all experiments, which are included in corresponding GitHub repository¹.

5.2 Methods

In this section, we describe the pre-training pretext task for weakly-supervised and self-supervised cases in Sec. 5.2.1, and the losses used during pre-training and fine-tuning in Sec. 5.2.2.

5.2.1 Pretext Task

The goal of our pre-training pretext task is to initialize nearly all model weights to optimize for transfer to the downstream person search task. In Fig. 5.2, we show the QC detection pretext task compared to standard OC detection between two augmentations of the same base image, shown in Image 1a and 1b. We use Image 1a as the query and Image 1b as the gallery. One person box is correctly annotated, denoted with b_q, b_g , while the other is not *i.e.* the annotation is missing. Note that anchor box₁ $b_a^{(1)}$ overlaps ground truth box b_g while anchor box₂ $b_a^{(2)}$ does not. Anchor embeddings x_a are used to compute box offsets $\hat{r}$ and anchor probabilities p either directly (OC) or relatively using $x_o = x_q - x_a$ (QC). We do not compute box offsets $\hat{r}$ for $b_a^{(2)}$ because it does not match any ground truth.

In OC detection, the goal is to regress each anchor class probability p to 1 if the anchor matches a ground truth box b_g *e.g.*, $b_a^{(1)}$, or 0 if it does not according to box overlap ≥ 0.5 IoU, *e.g.*, $b_a^{(2)}$. For matching anchors, we also regress box offsets $\hat{r}^{(1)}$ to targets $r^{(1)}$. In QC detection, the regression targets are the same, but predictions for each anchor are computed only *relative* to a query, explained by the QC vs. OC pathways in Chapter 4. QC detection results in more robust optimization, because it can better handle cases like $b_a^{(2)}$ in Fig. 5.2, which overlaps a person who is missing a ground truth annotation. Intuitively, QC detection allows us to learn salient features for transfer to person search without rigidly defining positive and negative detections.

We experiment with training for three levels of data annotation, enumerated in Fig. 5.3 as 1) Fully-supervised: person boxes annotated, identities tracked across images, 2) weakly-supervised: person boxes annotated, identities *not* tracked across images, 3) self-supervised: all boxes generated randomly. The key distinction is between *what* is annotated: person boxes for FS and WS, random boxes for SSL, and whether identities are *tracked* between images: yes for FS, no for WS and SSL. Since identities are not tracked between images for WS and SSL data, we generate correspondence between images using data augmentation: we know that a box in an image corresponds to the same box with potentially different coordinates in an augmented version of that image. Augmentations consist of scaling, cropping, and horizontal flipping.

¹<https://github.com/LukeJaffe/spnet>

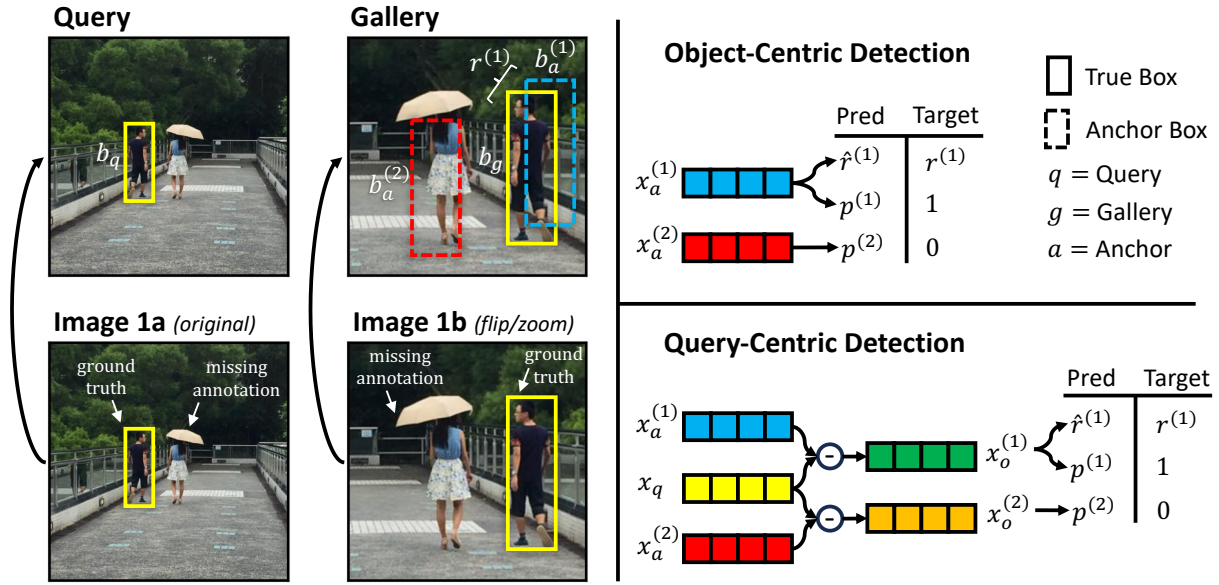


Figure 5.2: Visual comparison of object-centric (OC) and query-centric (QC) detection tasks between two augmentations of the same base image, shown in Image 1a and 1b). We use Image 1a as the query and Image 1b as the gallery. One person box is correctly annotated, denoted with b_q , b_g , while the other is not *i.e.* the annotation is missing. Note that anchor box₁ $b_a^{(1)}$ overlaps ground truth box b_g while anchor box₂ $b_a^{(2)}$ does not. Anchor embeddings x_a are used to compute box offsets $\hat{r}$ and anchor probabilities p either directly (OC) or relatively using $x_o = x_q - x_a$ (QC). We do not compute box offsets $\hat{r}$ for $b_a^{(2)}$ because it does not match any ground truth.

For QC detection on weakly-labeled data, each image in a batch is augmented twice, with each augmented image taking a turn as both query and gallery. Annotated boxes in the query image are compared against the corresponding box in the gallery image, and assigned to anchors with overlap ≥ 0.5 IoU. Self comparisons are allowed as well, in which the query and gallery are the same image.

For all cases, OC optimization treats annotated boxes as true objects with the same anchor matching criterion as QC, and pushes anchor scores to 0 or 1 as shown in Fig. 5.2. We also augment batch images k times for the OC case to fairly compare with QC, though it is not necessary.

For the OC SSL case, the implication is clear: the optimization is ill-posed. In the limit, we are teaching the model that every conceivable anchor box is both positive and negative, and that box regression offsets should be pushed towards every different target within the matching range. Since the QC objective enforces regression of anchors only relative to each query, it should optimize even in this use case. This hypothesis is tested in Sec. 5.3.7.

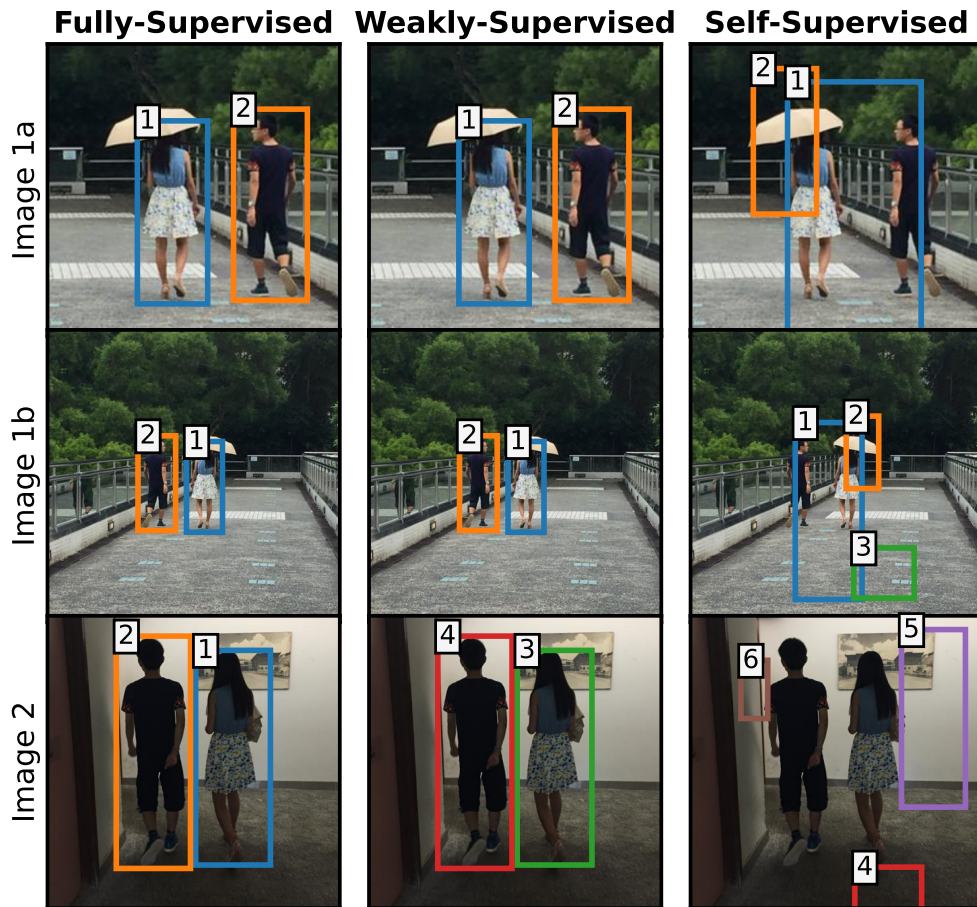


Figure 5.3: Augmentation and labeling for three cases in the CUHK-SYSU dataset. Numbers/colors indicate identity labels. Two different base images shown, Images 1 and 2, with Image 1 augmented twice, shown in 1a and 1b. Annotation levels: 1) Fully-supervised: person boxes annotated, identities tracked across images, 2) weakly-supervised: person boxes annotated, identities *not* tracked across images, 3) self-supervised: all boxes generated randomly, correspondence comes only from augmentations of the same image.

5.2.2 Losses

In this subsection, we describe all re-id loss functions and auxiliary loss functions used with SPNet for pre-training and fine-tuning, beyond the SPNet losses described in Chapter 4.

Re-id Losses. For our re-id losses, we use two variations of the normalized temperature-scaled cross-entropy loss [73], [75]. We use the terms *positive pair* and *negative pair* to refer to two embeddings with the same label or different labels respectively.

We define the probability for positive pair (x, x^+) under the contrastive objective as

$$p(x, x^+) = \frac{s_\tau(x, x^+)}{s_\tau(x, x^+) + \sum_{x^- \in \chi_x^-} s_\tau(x, x^-)} \quad (5.1)$$

With the full contrastive loss expressed as

$$\mathcal{L}_{\text{reid}} = -\log \sum_{x \in \chi} \sum_{x^+ \in \chi_x^+} p(x, x^+) \quad (5.2)$$

using $s_\tau(u, v) \triangleq \exp(\text{sim}(u, v)/\tau)$ and $\text{sim}(u, v) \triangleq u \cdot v / \|u\| \|v\|$, $u, v \in \mathbb{R}^d$. χ denotes the set of all x , χ_x^+ the set of all positive samples x^+ for x , and χ_x^- the set of all negative samples x^- for x . To examine variations of this loss, we simply vary the definitions of χ_x^+ and χ_x^- .

Fine-Tuning Re-id Loss. During fine-tuning, we use the standard online instance matching (OIM) loss from [1].

Pre-Training Re-id Loss. During pre-training, we use a variation of the momentum contrast loss (MoCo) from [154]. Similar to MoCo, we define an encoder network which is updated via gradient descent, and a momentum network, which is updated as a moving average of the encoder. We call embeddings from the encoder network x_e and embeddings from the momentum network x_m . Define $\bar{x}_m$ as the mean of all predicted box embeddings with corresponding box having $\text{IoU} \geq 0.7$ with a given ground truth box. We store embeddings $\bar{x}_m$ in a queue during training. Then, we form positive pairs $\chi_{x_e}^+$ with all embeddings $(x_e, \bar{x}_m^+)$ and negative pairs $\chi_{x_e}^-$ with all embeddings $(x_e, \bar{x}_m^-)$.

We are effectively comparing current embeddings to moving average cluster centroids [123], in the limit where the cluster consists of only one person image. In absence of cluster pseudo-labels similar to those used in [123] and other weakly-supervised methods, it is critical to have a representative embedding to compare against aside from the image itself. As in [114], we found it useful to utilize proposals with $\text{IoU} \geq 0.7$, but did not find it beneficial to weight them by IoU in the re-id loss, as in that work.

SSL Auxiliary Loss. For SSL pre-training on ImageNet-1k, we experiment with the option of an additional auxiliary loss $\mathcal{L}_{\text{aux}}$, which can improve intra-image feature compactness, in opposition to the other losses. In short, the problem is that traditional ImageNet-1k classifier and SSL pre-training mechanisms tend to push intra-image features closer together, while our losses tend to push them further apart, by forcing distinction between intra-image regions. The result is that our pre-training procedure has the potential to reduce the feature quality that is learned from ImageNet-1k pre-training, and potentially harm fine-tuning performance. To address this issue, we can simply add back in a loss which pushes intra-image features

close together again by treating crops from the same image as having the same label, similar to SimCLR [73] or MoCo [154]. In our case, we choose a triplet margin loss, with the margin parameter set to 0. This simply enforces features within an image to be closer to one another than they are to features from other images. The loss does not necessarily conflict with the other SPNet losses, because intra-image feature clusters can still be large, and have large intra-feature distance, so long as they are not too close to other image features.

Learning without Forgetting Loss. When pre-training from a backbone which has already been pre-trained, it is often desirable to explicitly retain features from the original pre-training while updating weights for the new objective. In *Learning without Forgetting* (LWF) [183], an additional knowledge distillation loss [141] is optimized to retain the predictions of the pre-training model during fine-tuning. While many other objectives have been developed since this original strategy, it has been recently reaffirmed in [184], and we compare its use to other options in our ablation studies.

Since we initialize our model backbone from ImageNet-1k classifier weights before pre-training, the idea is that this loss will help preserve useful features learned during the classifier pre-training. This loss $\mathcal{L}_{\text{LWF}}$ is simply added to the other losses when used, using temperature $T = 2$ as in [183]. To compute this loss, we leave on the original ImageNet1k classifier head, although it is not used for our target task.

Final Loss. The final loss for both pre-training and fine-tuning is simply the unweighted sum of all described losses:

$$\mathcal{L} = \mathcal{L}_{\text{reid}} + \mathcal{L}_{\text{anc}} + \sum_{t=1}^{\mathcal{T}+1} \mathcal{L}_{\text{reg}}^{(t)} + \mathcal{L}_{\text{cls}}^{(t)} \quad (5.3)$$

where $\mathcal{T}$ is the number of cascade stages, and $\mathcal{L}_{\text{LWF}}$ may be optionally added during pre-training.

5.3 Experiments and Analysis

In this section, we describe the datasets and evaluation procedures used in Sec. 5.3.1, implementation details of the model and optimization methods in Sec. 5.3.2, a comparison of our best results with SOTA alternatives in Sec. 5.3.3, a comparison of our pre-training methods with pre-training alternatives in Sec. 5.3.4, several ablation studies for weakly-supervised pre-training in Sec. 5.3.5, an ablation study with limited samples in Sec. 5.3.6, and self-supervised pre-training results in Sec. 5.3.7.

5.3.1 Datasets and Evaluation

Datasets. We perform weakly-supervised pre-training on the train partition of the COCOPersons dataset [195], which contains 64k scenes with 257k person box annotations. COCOPersons is the subset of MS-COCO2017 [64] images containing at least one person

annotation; non-person annotations are ignored. For self-supervised pre-training, we use the train partition of ImageNet1k (IN1K), without any labels or annotations. We fine-tune models on the two standard person search benchmark datasets: CUHK-SYSU [1] and PRW [100].

Evaluation. We evaluate model pre-training performance using three strategies: 1) out-of-the-box (OOTB) performance, in which models are evaluated directly on the target dataset, without any fine-tuning, 2) linear probing (LP) [181], in which only the Bridge Layer is fine-tuned, and 3) full fine-tuning (FT), in which all model layers are fine-tuned. We evaluate all models by measuring fine-tuning performance once on the standard retrieval test scenario for CUHK-SYSU or PRW. While we compare QC vs. OC pre-training, fine-tuning is always OC. In Sec. 5.3.5, we show that QC fine-tuning is less effective than OC fine-tuning.

We will show that the unique architecture of the QC SPNet allows for detection optimization by analyzing linear probing, even when the pre-training is fully self-supervised. While linear probing is useful as an analysis tool on its own as shown in Sec. 5.3.7, we do not find it beneficial to perform linear probing prior to full fine-tuning as suggested in [182].

To measure performance, we use standard detection performance metrics of recall@0.5 IoU and average precision@0.5 (AP@0.5) IoU, indicating predictions with an overlap of > 0.5 IoU with a ground truth box are considered matches. For re-id evaluation, we use the standard metrics of mean average precision (mAP) and top-1 accuracy (top-1). We distinguish between re-id measured on ground truth boxes as GT mAP and GT top-1, vs. re-id measured on detected boxes as mAP and top-1. Measuring ground truth re-id performance is equivalent to using a perfect object detector, and helps us understand re-id performance independent of detector quality.

5.3.2 Implementation Details

Model Configurations. We perform experiments with two variants of the SPNet model: SPNet-S(mall) and SPNet-L(arge), with differences shown in Tab. 5.1. In Tab. 5.1, Shared Heads means the same Box Embedder from Fig. 4.1 is used for both detection and re-id, and if it not checked in the table, the Box Embedder is duplicated for detection and re-id. As before, Post-Backbone refers to any weights after the backbone in Fig. 4.1. Unless otherwise stated, SPNet-S uses a ConvNeXt-Tiny [202] backbone, and SPNet-L uses a ConvNeXt-Base backbone. We also measure performance of the SPNet-L model with the Gallery Filter Network (GFN) module from [3], since the SeqNeXt model from [3] achieved previous SOTA results using the GFN.

Self-Supervised Training Recipe. For self-supervised pre-training on IN1k, we randomly generate 10 bounding boxes per image at the start of each training epoch, with side length sampled log-uniformly in range 8 to 128 pixels, and aspect ratio sampled log-uniformly in range $1/3$ to 3. We enforce boxes in the same image to have overlap IoU < 0.1 .

Optimization. For all experiments, we pre-train and fine-tune for 30 epochs by default, using the AdamW optimizer [206], a cosine-annealed schedule, and linear warmup following [206], [207].

Pre-training Optimization. For the pre-training re-id loss, we use the MoCo objective described in Sec. 5.2.2 with a queue length of 65,536, momentum of 0.9999, and temperature $\tau = 0.1$. We use the Pre-train config from Tab. 5.1 for learning rate and weight decay settings, with the Learning without Forgetting (LWF) loss [183].

We train for 5 epochs for randomly-initialized SSL trials, and 1-3 epochs for IN1k-initialized SSL trials. Further training for the IN1k classifier initialization case was found to degrade existing features.

In addition, we consider two forms of weight initialization for the backbone: 1) random initialization and 2) initialization with IN1k classifier pre-trained weights. We primarily study the second case with IN1k classifier pre-training, since it is common practice for person search and highly effective for most natural image datasets. This is the default initialization unless otherwise stated, and we refer to our pre-training as *secondary* in this case for clarity.

We also experiment with four variations of layer freezing and learning rates for the backbone and post-backbone modules, with results shown in Sec. 5.3.5 Fig. 5.4. We note that QC outperforms OC pre-training for nearly all configurations, and that the LWF loss did not significantly impact results.

Fine-tuning Optimization. For the fine-tuning re-id loss, we use the OIM objective described in Sec. 5.2.2 with temperature $1/30$, momentum 0.5, and queue length 5,000 for CUHK and 500 for PRW (standard settings). We use the Fine-tune config from Tab. 5.1, with image size 512×512 used for SPNet-S, and image size 1024×1024 used for SPNet-L.

Linear Probing. We implement linear probing by optimizing only the affine Bridge Layer in the SP Block, shown in Fig. 4.1, while other model weights are frozen. This version of linear probing is not convex as in the original formulation [181], but still converges quickly using the same optimization and augmentation used for fine-tuning. All linear probing runs train for 5 epochs using the baseline fine-tune optimization configuration.

Image Augmentation. We adopt three methods of image augmentation, which all consist of combinations of scaling, cropping, and horizontal flipping. For consistency, we label these here using the configuration name used in the code. We call the standard augmentation for person search *window resize* as in [3], abbreviated by **wrs**. This consists of scaling the image to fit in a window with shortest side length 900 and longest side length 1,500. **wrs** augmentation is used only for evaluation. In **rrc2** augmentation, we first perform **wrs** scaling, then randomly select from two types of crops, followed by random horizontal flip: 1) fixed size square random crop containing at least one bounding box in the image, chosen at random, and 2) random sized crop containing all bounding boxes in the image, that is then resized to a fixed square size. These square crop sizes are 512×512 during pre-training and SPNet-S fine-tuning, and 1024×1024 during SPNet-L fine-tuning. Crop size is one of the most important parameters for controlling memory usage. **rrc2** augmentation is used for all fine-tuning runs. **rrc_scale** augmentation is the same as **rrc2** augmentation, except we scale randomly in the range $0.5\times$ to $2\times$ instead of **wrs** scaling. **rrc_scale** augmentation is used for all pre-training runs.

Hardware and Training Time. All models were trained using a single A100 GPU with 82GB VRAM. Pre-training and fine-tuning times are shown in Tab. 5.2, with the QC pre-training taking 30 hours for SPNet-S and 46.5 hours for SPNet-L. By comparison, PASS

Config. Name	Cascade Steps	Shared Heads	Re-id Dim
SPNet-S	0	✓	128
SPNet-L	2		2048

Config. Name	Backbone		Post-Backbone		LWF Loss	Image Size
	LR	WD	LR	WD		
Pre-train	1e−5	0	1e−4	1e−3	✓	512
Fine-tune	1e−4	5e−4	1e−4	5e−4		512 or 1024

Table 5.1: SPNet architecture (top) and layer optimization (bottom) hyperparameter configurations. LR = Learning Rate, WD = Weight Decay, LWF = Learning without Forgetting [183].

Model	Backbone	Pre-train (QC)	Pre-train (OC)	Fine-tune (OC)	
		COCOPersons	COCOPersons	CUHK-SYSU	PRW
SPNet-S	ConvNeXt-Tiny	30.0h	24.5h	2.5h	1.5h
SPNet-L	ConvNeXt-Base	46.5h	40.5h	9.0h	4.5h

Table 5.2: Training times for person search model variants on a single A100 GPU. Pre-training and fine-tuning are each done for 30 epochs on the respective datasets.

[173] pre-trains for 100 epochs on 8 A100 GPUs, which takes 60-120 hours depending on the backbone. SOLIDER [129] and LUPerson [169] do not report pre-training times, but they pre-train for 110 and 200 epochs respectively, on 8 V100 GPUs. In addition, LUPerson, PASS, and SOLIDER each pre-train on the much larger LUPerson dataset with 4.18M images vs. our 64k images in the COCOPersons dataset.

5.3.3 Comparison with State-of-the-art

In Tab. 5.3, we compare performance of SPNet with recent state-of-the-art person search models. We show that SPNet-L, with QC pre-training on COCOPersons, matches or exceeds other models in most metrics for the CUHK-SYSU and PRW datasets, e.g., improving mAP by +1.7% on PRW over previous SOTA LEAPS [114] and +1.6% over alternate pre-training procedure SOLIDER. When adding the GFN, in SPNet-L+GFN, performance improves on the CUHK-SYSU dataset to SOTA, exceeding previous SOTA SeqNeXt+GFN by +0.2% in both mAP and top-1, though performance drops for mAP on PRW.

In addition, we show that the QC pre-training itself on SPNet-L adds 0.5% mAP for

Person Search Model	Backbone	Pre-training Method	Pre-training Dataset	CUHK-SYSU		PRW	
				mAP	top-1	mAP	top-1
<i>Two-step (QC)</i>							
IGPN [120]	ResNet50	Classifier	ImageNet-1k	90.3	91.4	47.2	87.0
TCTS [121]	ResNet50	Classifier	ImageNet-1k	93.9	95.1	46.8	87.5
<i>Two-step (OC)</i>							
IDE [100]	ResNet50 [198]	Classifier	ImageNet-1k	-	-	20.5	48.3
MGTS [101]	VGG16	Classifier	ImageNet-1k	83.0	83.7	32.6	72.1
CLSA [102]	ResNet50	Classifier	ImageNet-1k	87.2	88.5	38.7	65.0
RDLR [103]	ResNet50	Classifier	ImageNet-1k	93.0	94.2	42.9	70.2
<i>One-step (QC)</i>							
QEEPS [118]	ResNet50	Classifier	ImageNet-1k	88.9	89.1	37.1	76.7
QGN [119]	ResNet50	Classifier	ImageNet-1k	93.7	94.4	46.7	82.9
<i>One-step (OC)</i>							
OIM [1]	ResNet50	Classifier	ImageNet-1k	75.5	78.7	21.3	49.4
NPSM [104]	ResNet50	Classifier	ImageNet-1k	77.9	81.2	24.2	53.1
APNet [105]	ResNet50	Classifier	ImageNet-1k	88.9	89.3	41.9	81.4
HOIM [106]	ResNet50	Classifier	ImageNet-1k	89.7	90.8	39.8	80.4
BINet [107]	ResNet50	Classifier	ImageNet-1k	90.0	90.7	45.3	81.7
NAE+ [108]	ResNet50	Classifier	ImageNet-1k	92.1	92.9	44.0	81.1
DKD [109]	ResNet50	Classifier	ImageNet-1k	93.1	94.2	50.5	87.1
AlignPS [110]	ResNet50	Classifier	ImageNet-1k	94.0	94.5	46.1	82.1
SeqNet [111]	ResNet50	Classifier	ImageNet-1k	93.8	94.6	46.7	83.4
SeqNet+CBGM [111]	ResNet50	Classifier	ImageNet-1k	94.8	95.7	47.6	87.6
SeqNet [129]	Swin-S [200]	Classifier	ImageNet-1k	91.1	92.0	45.8	82.9
SeqNet [129]	Swin-S	SOLIDER	LUPerson	95.3	95.7	59.4	87.0
COAT [112]	ResNet50	Classifier	ImageNet-1k	94.2	94.7	53.3	87.4
COAT+CBGM [112]	ResNet50	Classifier	ImageNet-1k	94.8	95.2	54.0	89.1
MHGAM [115]	ResNet50	Classifier	ImageNet-1k	94.9	95.9	47.9	88.0
PSTR [113]	PVTv2-B2 [201]	Classifier	ImageNet-1k	95.2	96.2	56.5	89.7
SeqNeXt [3]	ConvNeXt-B	Classifier	ImageNet-1k	96.1	96.5	57.6	89.5
SeqNeXt+GFN [3]	ConvNeXt-B	Classifier	ImageNet-1k	96.4	97.0	58.3	92.4
LEAPS [114]	PVTv2-B2	Classifier	ImageNet-1k	96.4	96.9	59.5	89.7
SPNet-L	ConvNeXt-B	Classifier	ImageNet-1k	95.9	96.6	58.9	89.7
SPNet-L	ConvNeXt-B	Ours-QC	COCOPersons	96.4	97.0	61.2	90.9
SPNet-L+GFN	ConvNeXt-B	Ours-QC	COCOPersons	96.6	97.2	60.5	92.0

Table 5.3: Comparison of SOTA models for the benchmark CUHK-SYSU and PRW datasets. +GFN indicates using the Gallery Filter Network from [3]. +CBGM indicates using Context Bipartite Graph Matching from [111].

CUHK-SYSU and 2.3% mAP for PRW. This shows that the benefit from query-centric pre-training extends to larger models which perform better, even when performance statistics are nearly saturated, as on CUHK-SYSU and PRW.

Person Search Model	Backbone	Pre-training Method	Pre-training Dataset	CUHK-SYSU		PRW	
				mAP	top-1	mAP	top-1
SPNet-S	ConvNeXt-T	None	None	69.8	73.4	22.3	69.1
SPNet-S	ConvNeXt-T	Classifier	ImageNet-1k	91.9	93.2	52.7	86.6
SPNet-S	ConvNeXt-T	Ours-QC	COCOPersons	93.3	94.1	55.6	89.5
SPNet-S	ResNet50	None	None	77.6	80.1	22.3	70.2
SPNet-S	ResNet50	LUPerson [169]	LUPerson	89.8	91.1	47.4	85.3
SPNet-S	ResNet50	Classifier	ImageNet-1k	89.5	90.9	48.6	86.1
SPNet-S	ResNet50	Ours-QC	COCOPersons	90.2	91.8	52.0	87.7
SPNet-S	Swin-T	None	None	60.3	61.9	18.0	66.3
SPNet-S	Swin-T	SOLIDER [129]	LUPerson	83.0	84.8	36.2	77.9
SPNet-S	Swin-T	Classifier	ImageNet-1k	89.5	90.5	46.3	84.0
SPNet-S	Swin-T	Ours-QC	COCOPersons	92.4	93.4	51.7	87.9
SPNet-S	ViT-S/16	None	None	27.4	27.6	<0.01	2.6
SPNet-S	ViT-S/16	PASS [173]	LUPerson	31.1	29.7	<0.01	2.1
SPNet-S	ViT-S/16	Classifier	ImageNet-1k	66.1	68.1	<0.01	4.3
SPNet-S	ViT-S/16	Ours-QC	COCOPersons	87.4	88.8	23.7	68.4

Table 5.4: Comparison of performance gained from pre-training SPNet using our method vs. unsupervised person re-id backbone pre-training.

Pre-train Method	CUHK-SYSU			PRW		
	mAP	GT mAP	AP@0.5	mAP	GT mAP	AP@0.5
None	69.8	85.8	47.8	22.3	28.7	66.8
Classifier	91.9	95.8	83.2	52.7	56.6	88.8
Ours-OC	92.8	96.3	83.9	54.5	57.6	89.1
Ours-QC	93.3	95.9	84.9	55.6	58.7	88.9

Table 5.5: Person search, re-id, and detection metrics for pre-training method comparison on SPNet-S with ConvNeXt-Tiny backbone.

Method	CUHK-SYSU		PRW	
	mAP	top-1	mAP	top-1
Baseline	91.9	93.2	52.7	86.6
Re-id Only	91.5	93.0	51.9	87.4
OC Re-id GT	92.4	93.7	55.7	89.1
QC Re-id GT	93.2	94.3	55.1	88.5
OC Original	92.8	94.1	54.5	88.8
QC Original	93.3	94.1	55.6	89.5

Table 5.6: Comparison of different settings of the re-id pre-training loss for fine-tuning performance. For *re-id GT*, only GT box embeddings are used to compute re-id loss, and for *re-id only*, we optimize only the re-id loss (with GT boxes) and no other losses. In original trials, both GT and detected box embeddings are used to compute re-id loss.

Config. Name	Ablation Name	Cascade Steps		Shared Heads		Re-id Dim
SPNet-S	c0-share-d128	0		✓		128
-	c0-sep-d128	0				128
-	c2-share-d128	2		✓		128
-	c2-sep-d128	2				128
SPNet-L	c2-sep-d2048	2				2048

Config. Name	Ablation Name	Backbone		Post-Backbone		LWF Loss
		LR	WD	LR	WD	
-	Uniform LR	1e−4	1e−3	1e−4	1e−3	
-	Mixed LR	1e−5	0	1e−4	1e−3	
Pre-train	Mixed LR + LWF	1e−5	0	1e−4	1e−3	✓
-	Frozen-BB	Frozen	Frozen	1e−4	1e−3	
Fine-tune	Fine-Tune	1e−4	5e−4	1e−4	5e−4	

Table 5.7: SPNet architecture (top) and layer optimization (bottom) hyperparameter configurations. The Ablation Name is given by the number of Cascade Steps, followed by whether Shared Heads are used, and ending with the Re-id Dim. LR = Learning Rate, WD = Weight Decay.

5.3.4 Comparison with Pre-training Alternatives

In Tab. 5.4, we compare our QC pre-training approach to backbone-only pre-training approaches for unsupervised person re-id which published model weights: SOLIDER [129], LUPerson [169], and PASS [173]. All comparisons are done for the SPNet-S model, though the backbone architecture varies based on which method we compare against, based on the weights they made available. We also compare against ImageNet-1k classifier backbone weights, and random backbone initialization, listed in Tab. 5.4 as None.

We find that our QC pre-training approach outperforms all alternative initialization strategies, in some cases by wide margins. Surprisingly, we also find that ImageNet-1k classifier backbone pre-training alone exceeds nearly all unsupervised person re-id backbone pre-training methods for the SPNet-S model. This is despite these methods showing far superior fine-tuning for supervised person re-id alone vs. ImageNet-1k weights in their respective papers. The original LUPerson backbone [169] is the only backbone which exceeds ImageNet-1k with SPNet-S, and only on CUHK-SYSU, but this backbone still does not match our QC pre-training. This is an unexpected result for the SOLIDER pre-training method, which did exceed ImageNet-1k classifier pre-training for the SeqNet person search model, as shown in Tab. 5.3, but not for SPNet. This is likely due to a mismatch of the SOLIDER pre-training objective in creating effective features for the feature pyramid network in SPNet, caused by exacerbating scale misalignment of features [110]. ImageNet-1k weights, in contrast, are more general, and pair better with the feature pyramid network out-of-the-box.

Likewise, some backbone architectures, such as ViT-S/16 do not naturally pair well with the feature pyramid network in SPNet. This results in an inability to optimize for the PRW dataset for most settings, but still we find that the QC pre-training initializes the model to a point where it can successfully fine-tune in this case. The result is that our method achieves 68.4% PRW top-1 while the next highest, ImageNet-1k, achieves only 4.3% top-1.

Finally, we conclude from the Hardware and Training Times in Sec. 5.3.2 and the results comparison in Tabs. 5.3 and 5.4 that our method achieves much greater pre-training efficiency for the person search problem than the described alternatives.

5.3.5 Weakly-Supervised Ablation Studies

In this subsection, we detail several ablation studies designed to analyze the performance of weakly-supervised pre-training, followed by supervised fine-tuning. Our goal is to understand relative performance of query-centric and object-centric training, the importance of the re-id loss, model architecture choices, and details of how to load/use pre-trained weights for fine-tuning.

Query-Centric vs. Object-Centric Baseline. In Tab. 5.5, we compare QC vs. OC pre-training to ImageNet-1k Classifier pre-training, and random backbone initialization, denoted as None. The comparison is done for SPNet-S with a ConvNeXt-Tiny backbone, and we show person search mAP in addition to metrics of re-id, GT mAP, and detection, AP@0.5. This helps us understand the contribution of pre-training to detection, re-id, and

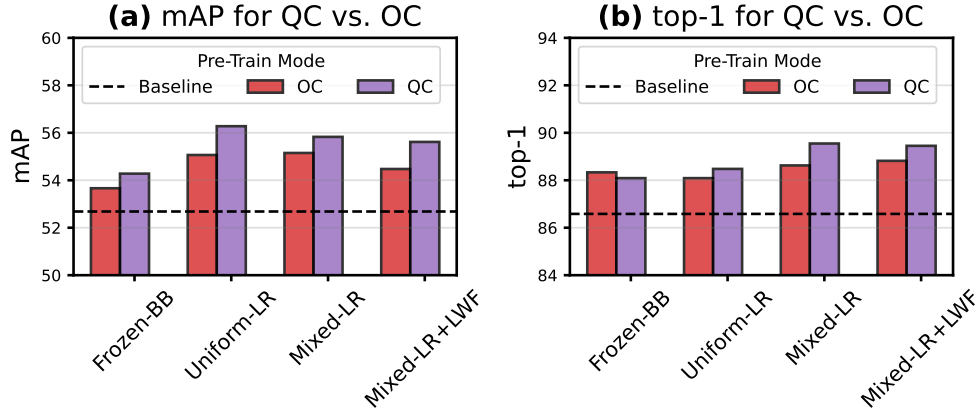


Figure 5.4: Comparison of SPNet fine-tune performance on PRW with QC vs. OC pre-training for varying layer optimization hyperparameters. The corresponding configurations are described in Tab. 5.7.

the combined person search problem. We show both QC and OC pre-training improve over ImageNet-1k Classifier pre-training for all metrics, and that QC is superior to OC pre-training for most metrics. Most importantly, we show that QC pre-training exceeds OC pre-training for person search mAP on both datasets: +0.5% on CUHK-SYSU, +1.1% on PRW, though whether re-id or detection benefit more depends on the dataset. Finally, we show that all pre-training vastly exceeds random initialization, denoted as None.

Re-id Pre-training. In Tab. 5.6, we compare different methods of handling the re-id loss during pre-training. The goal is to isolate the effect of the re-id loss on pre-training, for both QC and OC methods. We found that pre-training only the re-id loss, without optimizing the other losses, was not beneficial for all statistics.

We found that QC pre-training using detected boxes with $\text{IoU} \geq 0.7$ performed best on balance. When only ground truth (GT) boxes were used to compute the re-id loss, and no detected boxes, we found that QC pre-training was impaired more than OC pre-training.

Query-Centric vs. Object-Centric Pre-training. In Fig. 5.4, we compare query-centric vs. object-centric pre-training for fine-tuning on the PRW dataset for four configurations, given in Tab. 5.7 (bottom). All models with pre-training significantly outperform the baseline in mAP and top-1 accuracy, and nearly all QC models outperform the corresponding OC model. One exception is the Frozen Backbone model for top-1 accuracy, showing that the query-centric pre-training performs better when the backbone is optimized in addition to later layers.

Query-Centric vs. Object-Centric Fine-tuning. While we primarily explore OC fine-tuning (OC-FT) and evaluation in this work, our framework supports QC fine-tuning (QC-FT) and evaluation as well. QC fine-tuning is done using the same procedure as QC pre-training, with $k = 2$ augmentations per image, but using the `rrc2` augmentation method.

Method	CUHK-SYSU		PRW	
	mAP	top-1	mAP	top-1
OC-FT	91.9	93.2	52.7	86.6
OC-FT+PT	93.3	94.1	55.6	89.5
<i>Gain from PT</i>	+1.4	+0.9	+2.9	+2.9
QC-FT	80.8	87.7	54.0	86.7
QC-FT+PT	77.4	85.4	54.1	87.1
<i>Gain from PT</i>	-3.4	-2.3	+0.1	+0.4

Table 5.8: Comparison of QC and OC fine-tuning (-FT) for baseline model vs. COCOPersons QC pre-trained model (+PT).

In Tab. 5.8, we compare QC-FT and OC-FT for models with and without COCOPersons QC pre-training. While the QC-FT model outperforms the OC-FT model on PRW top-1 without PT, OC-FT performance is drastically higher for mAP and top-1 on CUHK-SYSU, by >10% mAP. In addition, the OC-FT model benefits significantly from QC pre-training for both datasets and metrics, while the QC-FT model benefits only slightly on PRW and is actually harmed on CUHK-SYSU.

This suggests QC fine-tuning is more impacted by the distribution shift between pre-training and fine-tuning datasets, at least for the shared embedding head model. It also suggests the QC pre-train to OC fine-tune transition has a regularizing effect, limiting the effect of overfitting on the pre-training dataset.

Model Architecture. To understand the model architecture design space, we examine variations in number of cascade steps, shared vs. separate embeddings heads, and size of the re-id embedding dimension. In Fig. 5.5, we compare the configurations described in Tab. 5.7 (top). We show that the cascaded model with two steps is better than the base model, using separate embedding and re-id heads results in better fine-tuning performance, and larger re-id embedding dimension is beneficial. In addition, all architecture configurations benefit from pre-training, although the cascaded models benefit less.

Cascaded Weight Loading. When loading weights from pre-trained SPNet into a larger version with multiple cascade stages, we have to make a choice about how to load weights into layers not utilized during pre-training. We explore two simple options: loading weights only into layers that were pre-trained, and duplicating weights into equivalent layers in each cascade stage, shown by *PT* vs. *PT-Rep* in Fig. 5.5. For the *c2-sep-d2048* configuration, explained in Tab. 5.7, the PT-Rep method of weight loading outperforms PT, showing that the benefits of the pre-trained weight initialization extend to cascade stages, even though the pre-trained model was trained without cascading.

Anchor Probability Visualization. In Fig. 5.6, we visually compare anchor probabilities

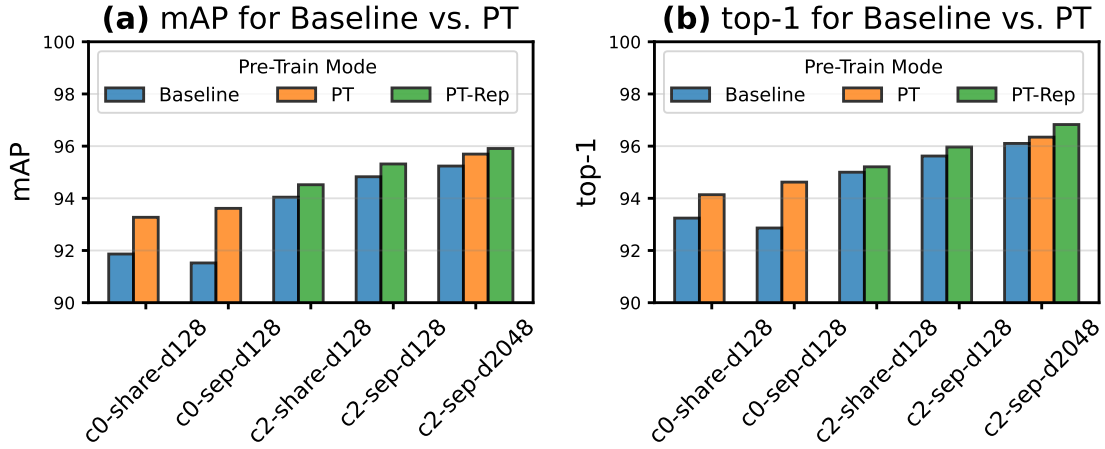


Figure 5.5: Comparison of SPNet fine-tune performance on CUHK-SYSU with and without QC pre-training for varying architecture hyperparameters. The configuration names, labeling the x-axis in the plots, are described in Tab. 5.7. PT=pre-trained, PT-Rep=pre-trained with replicated weight loading for cascade layers.

for the classifier (ImageNet-1k) pre-trained model vs. the weakly-supervised (WS) (COCOPersons) pre-trained model and the supervised (CUHK-SYSU) fine-tuned model (all QC). While the classifier model assigns high probability to regions containing persons, it is not as precise as the WS or supervised models. Only the supervised model is able to assign high probabilities to the matching region, with low probabilities to all other regions. This shows we are able to learn a model which produces good anchor candidates in the WS setting, but full supervision or fine-tuning is still needed for precise retrieval with different view angles and occlusions.

5.3.6 Limited Sample Ablation

We hypothesize that QC pre-training should learn more useful features than OC pre-training in the weakly-labeled scenario with sparse labels. Specifically, we focus on the case where we want to utilize images that have multiple people present, but only some fraction of them are annotated. This has applications both for manual labeling, in which persons may be missed, or not all persons are annotated by design, and also for automatically-annotated labeling, in which a detector is used to label imagery, but may have low recall with true GT boxes.

To model this scenario, we use the subset of COCOPersons with at least two known annotated persons in each image, which comprises 39k out of the full 64k images. Then, we remove some fraction of annotations from the dataset, always leaving at least one annotated person in each image, resulting in images with missing annotations, as shown in Fig. 5.2. We create successive partitions in increments of 20% of annotations removed, with each smaller partition being a subset of all larger partitions.

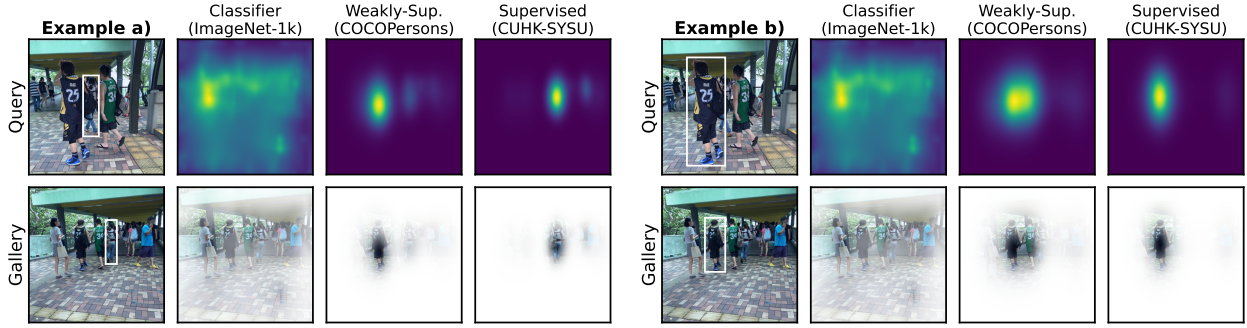


Figure 5.6: Visualization of anchor probabilities for three models and two different (a, b) query-gallery pairs from the same image pair, all from the CUHK-SYSU dataset. In the first column of each example, original images are shown, with the query box and matching gallery box outlined. In the following columns, all anchor probabilities for the gallery image are plotted for each model. Top row: probabilities shown as heatmap. Bottom row: probabilities shown as image alpha channel.

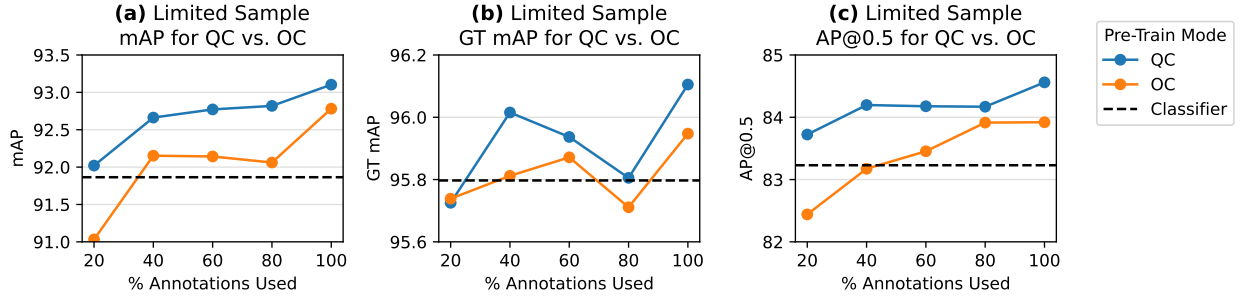


Figure 5.7: Plots of retrieval and detector stats for the CUHK-SYSU dataset for COCOPersons pre-trained QC and OC models fine-tuning with limited samples.

The results are shown in Fig. 5.7, where we compare how QC and OC evaluation are affected by the fraction of annotations which remain in the pre-training dataset. Results are compared to the baseline ImageNet-1k classifier backbone initialization, shown by the black dashed line. For all plots, we show that QC pre-training exceeds OC pre-training in all sample regimes for full fine-tuning, with the exception of GT mAP for 20% annotations used, where the two are tied. Even when only 20% of samples are used, QC pre-training offers a small benefit to fine-tuning for person search (mAP in plot (a)), where OC pre-training actually harms fine-tuning performance by forcing the model to learn that most ground truth person boxes are actually background. This is reinforced by the comparable relationship with detector performance shown in plot (c).

5.3.7 Self-Supervised Training

As explained in Sec. 5.2.1, we expect that only the QC model can train on SSL data, while the OC model fails to optimize. This is borne out in Fig. 5.8, where training loss is plotted for both QC and OC for the weakly-supervised (WS) and self-supervised (SSL) cases. While re-id loss optimization follows a similar trend for all cases, the detection losses differ drastically. As expected, we see that the SSL OC box regression and classification losses plateau nearly-immediately, while the SSL QC losses decrease steadily. By comparison, the WS losses for both OC and QC methods decrease steadily, showing that the SSL OC plateau is caused by inability of the OC objective to learn from randomly-generated annotations.

We show results of SSL pre-training on IN1k for two use cases: 1) pre-training from randomly initialized weights, shown in Tab. 5.9 and 2) (secondary) pre-training from weights initialized by IN1k classifier pre-training, shown in Tab. 5.10. For both weight initializations, linear-probed GT re-id performance is similar between QC and OC pre-trained models, but linear-probed detector AP is nonzero only for the QC pre-trained model, reaching 34.2% on CUHK-SYSU and 57.7% on PRW (for IN1k classifier init.). Likewise for random initialization in Tab. 5.9, linear-probed GT re-id performance is similar between QC and OC pre-trained models, but linear-probed detector AP is nonzero only for the QC pre-trained model, reaching 11.0% on CUHK-SYSU and 33.1% on PRW. This reaffirms our conclusions from observing Fig. 5.8.

In addition, Tab. 5.9 shows that the disparity between QC and OC pre-training translates to the full fine-tuning case, with QC outperforming OC by 3% for mAP on both datasets, and by more than 15% compared to the baseline.

While QC performance matches or exceeds OC performance in all statistics for both weight initializations, neither QC nor OC pre-training alone improved over the baseline for the IN1k classifier init. case. However, QC pre-training with the auxiliary SSL loss (QC-aux) does generate better performance than IN1k classifier init. for mAP and AP@0.5 on CUHK-SYSU, and nearly matches for GT mAP. Further research is needed on the optimization objective to determine an approach which captures the benefits of secondary pre-training, without eroding the already beneficial IN1k classifier weights.

5.4 Discussion

We have shown that the SPNet model benefits from QC pre-training and OC fine-tuning, and that pre-training can be done using either weakly-labeled person bounding boxes, or fully-unlabeled scenes via self-supervised learning. We evaluated performance of the model by pre-training on the ImageNet1k and COCOPersons datasets, and fine-tuning on the CUHK-SYSU and PRW person search datasets. We showed that 1) pre-training provides a significant boost to performance for all model variants, 2) QC pre-training benefits fine-tuning more than OC pre-training, 3) the model with QC pre-training achieves SOTA fine-tuning performance on CUHK-SYSU and PRW, and 4) QC pre-training performs better than OC

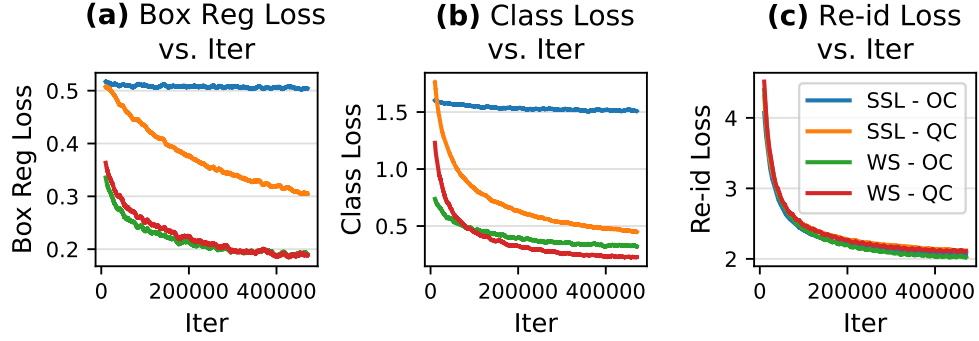


Figure 5.8: Training loss plots for QC and OC models for weakly-supervised (WS) pre-training on COCOPersons (COCO) vs. self-supervised (SSL) pre-training on IN1k.

<i>Weight Initialization: Random</i>							
Method	CUHK-SYSU			PRW			
	mAP	GT mAP	AP@0.5	mAP	GT mAP	mAP	AP@0.5
<i>Linear-Probed</i>							
Baseline	0.0	15.6	0.0	0.0	1.0	0.0	0.0
SSL (OC)	0.2	56.6	0.0	0.0	6.4	0.0	0.0
SSL (QC)	24.8	55.4	11.0	3.2	6.0	33.1	33.1
<i>Fine-Tuned</i>							
Baseline	69.8	85.8	47.8	22.3	28.7	66.8	66.8
SSL (OC)	82.4	90.0	72.4	34.5	38.3	85.2	85.2
SSL (QC)	85.7	91.2	75.4	37.5	40.9	85.5	85.5

Table 5.9: Results on CUHK-SYSU and PRW for IN1k SSL pre-trained SPNet models. Models start from random weight initialization prior to SSL pre-training. In the top half, results for linear probing are shown, and in the bottom half, results for full fine-tuning are shown.

<i>Weight Initialization: IN1k Classifier</i>							
Method	CUHK-SYSU			PRW			
	mAP	GT mAP	AP@0.5	mAP	GT mAP	mAP	AP@0.5
<i>Linear-Probed</i>							
Baseline	0.0	23.5	0.0	0.1	2.2		0.2
SSL (OC)	3.5	64.7	0.4	0.0	7.4		0.0
SSL (QC)	48.4	63.6	34.2	5.4	7.5		57.7
<i>Fine-Tuned</i>							
Baseline	91.9	95.8	83.2	52.7	56.6		88.8
SSL (OC)	91.3	95.5	82.5	50.6	54.2		88.6
SSL (QC)	91.5	95.5	83.6	52.4	55.7		88.6
SSL (QC-aux)	92.4	95.7	83.8	52.2	55.6		88.7

Table 5.10: Results on CUHK-SYSU and PRW for IN1k SSL pre-trained SPNet models. Models start from IN1k classifier pre-training, prior to SSL secondary pre-training. In the top half, results for linear probing are shown, and in the bottom half, results for full fine-tuning are shown.

pre-training when there are missing person labels in a scene. Finally, we showed that our end-to-end person search pre-training method is more effective and efficient than backbone-only pre-training alternatives.

Since the pre-training methods that we have discussed are not specific to person search, they can be readily applied to other domains. We suggest that the weakly-supervised pre-training method is useful for other identity search tasks, such as vehicle search and generic object search. The self-supervised pre-training is even more versatile, and could be used to pre-train for any LIR task, or even to pre-train only the vision backbone for arbitrary vision tasks. The ability of SPNet to enable pre-training for QC and OC tasks, while having faster optimization than non-anchor-based detector types, is a key reason why it is a good candidate for a universal model of LIR. This aspect of SPNet will be discussed further in Chapter 6.

Chapter 6

Conclusions and Future Work

We begin with a discussion of potential future work that explores how we can achieve a unified model of LIR using current approaches in Sec. 6.1. Our goal is to find a framing that can solve different LIR problems with the same model weights, while retaining inductive biases which improve efficiency of optimization and inference. After considering unified LIR under model types which are already designed for image localization, we will discuss how generalist sequence-to-sequence models can be adapted for LIR in Sec. 6.2. Finally, we will summarize the main findings of this dissertation in Sec. 6.3, and discuss societal impacts of the work in Sec. 6.4.

6.1 A Unified Framework of LIR

In this section, we discuss how to unify the problems in LIR as a single problem in Sec. 6.1.1, and how to solve the unified problem with a single corresponding model in Sec. 6.1.2. Then, we show how the closest existing model to addressing unified LIR, called UNINEXT [8], can be extended to better perform identity search in Sec. 6.1.3. Finally, we show how our SPNet model can be adapted as an alternative unified model of LIR with a superior inductive bias for localization in Sec. 6.1.4.

6.1.1 Unified Problem

The common thread of the four problem types visualized in Fig. 1.5, is that each localizes some type of prompt within one or more images, with optional post-processing such as tracking or ranking. This combined problem setting for localized image retrieval is visualized in Fig. 6.1.

6.1.2 Unified Model

A generic query-centric unified model which can solve the unified problem is visualized in Fig. 6.2. The model takes a prompt and image as input, and localizes the prompt in the

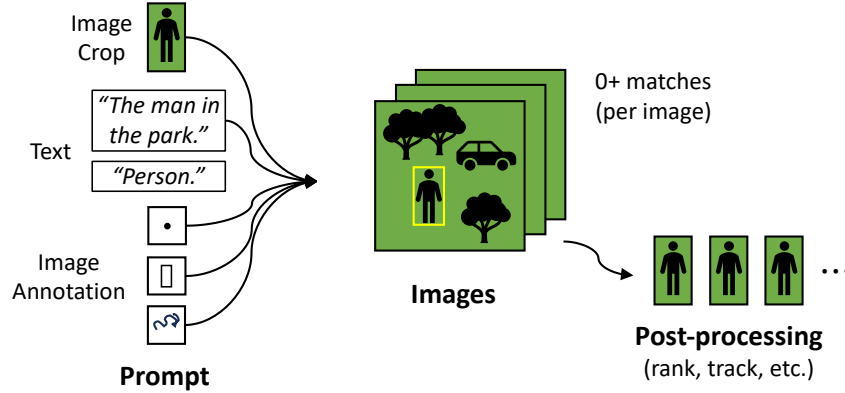


Figure 6.1: In the unified problem definition of localized image retrieval, image, text, annotation, or other prompts are used to localize one or more matches in a set of images, with optional post-processing.

image by encoding it is an embedding which is used by a detector module. As discussed in Chapter 1, each output localization has a corresponding embedding which enables computation of similarity and ranking.

It should be clear that our anchor-based query-centric model introduced in Chapter 4 can be trivially extended to this more general case simply by adding encoders for additional data types beyond image queries. Off-the-shelf text encoders can be borrowed from models such as BERT [208], GPT [209], or CLIP [33]. Annotations on the image itself can be encoded in a manner similar to SAM [5] and SEEM [6]. In addition, any other data which is reasonable to encode as an embedding could be used as well.

One key insight is that the object detection problem can be solved in a query-centric manner by using class name text embeddings as object classes. This can be made efficient by allowing these class text embeddings to exchange information, and performing detection on all classes collectively, as in the UNINEXT model [8].

6.1.3 Extending UNINEXT

Recent frameworks such as *universal instance perception model of the next generation* (UNINEXT) [8] offer an elegant approach to LIR problem unification by posing several problems as a variant of prompt localization. UNINEXT has most of the tools needed for unified LIR, including the ability to encode and utilize arbitrary prompts / queries, and output localizations in the form of segmentation masks, with corresponding embeddings. What UNINEXT lacks is integration of the type of identity-level learning and retrieval that we expect from *e.g.*, a model of person search. Therefore, while UNINEXT demonstrates performance for object tracking, it is not ideal in its current form to be used for identity search.

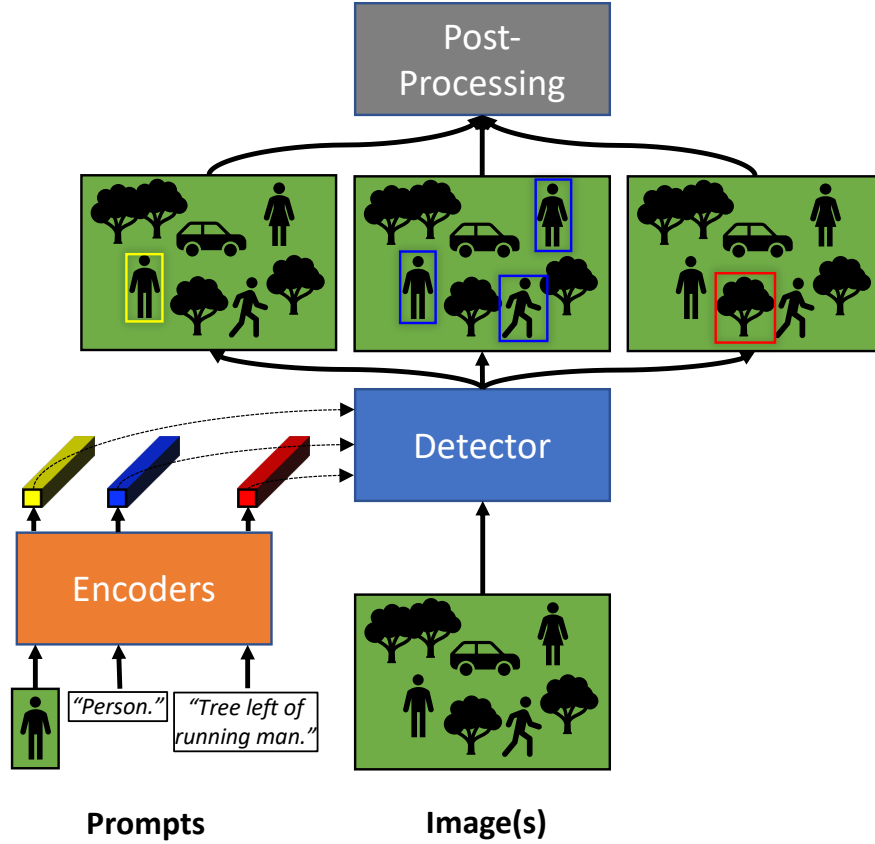


Figure 6.2: A query-centric detector which can localize arbitrary prompts, by encoding prompts with modality-specific encoders.

One approach to producing an improved unified model, then, would be to add identity-level learning to UNINEXT, using the lessons we have learned from person search. The simplest way to do so would be to use the person search OIM loss, and simply train this loss in conjunction with the existing UNINEXT loss terms.

However, the model would likely perform better if it accounted for the relationship between objects and object instances / identities. We suggest three potential methods for this task, based on box embeddings [144], [145], Poincaré embeddings [146] mentioned in Chapter 2, and a baseline of simply using separate embedding spaces for object and identity embeddings.

The way that UNINEXT currently implements class embeddings is to encode them as text prompts, then combine the text embedding with the target image features, using cross-attention, to produce a fused class representation embedding. Instance embeddings are learned for each decoded object, corresponding to the specific object and its localization in the target image. The mechanism UNINEXT uses to match prompts to output embeddings is to learn an association between the two with a focal loss, to account for the possibility of

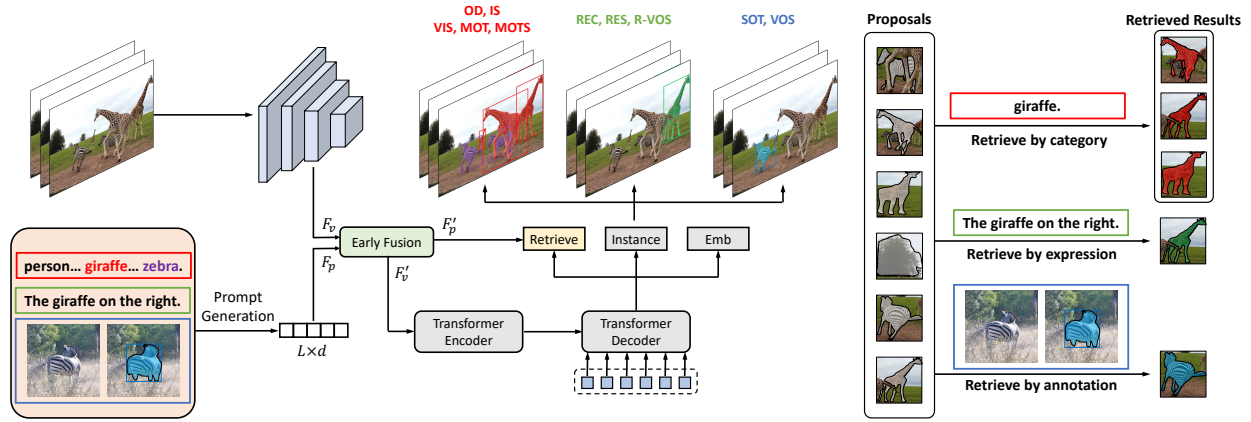


Figure 6.3: Figure from [8] showing the UNINEXT model architecture. UNINEXT encodes text and vision prompts, performs early fusion of prompts with image features, and combines fused features with decoder queries to segment images for various tasks.

many negatives. The result is that UNINEXT instance embeddings are effectively learned to be instantiations of a given class embedding dependent upon the current image. While this is shown to be effective for object tracking, it would not likely extend well to cases where two instances have significantly different visual appearance, *i.e.* generic identity search. Tracking is an easier case of identity search in that there is typically high visual similarity between instances in neighboring frames.

Separate Embedding Spaces UNINEXT effectively learns separate class and instance embedding spaces, in that class embeddings are derived from BERT [208] text tokens, and instance embeddings are associated between frames using a contrastive loss, intended for video tasks. However, no identity-level tasks like person search are trained on, so no comprehensive identity-level losses like OIM are used. Since the BERT text embeddings are fixed during UNINEXT training, it would be reasonable to simply add person search fine-tuning as one of the fine-tuning tasks. This would significantly improve recognition of intra-class person identity embeddings, and would compliment the inter-frame instance loss.

Box Embeddings A more principled approach would be to introduce a region-based representation like box embeddings. Box embeddings allow learning of hierarchical relationships between entities by encoding entities as geometric regions, such that boxes can intersect or fully be contained within one another, and have various volumes. By encoding object classes as textual box embeddings, we could augment UNINEXT by adding a loss which forces instance representations to lie within the corresponding class box, instead of effectively matching it exactly. This would pair better with an identity-level loss like OIM by allowing greater intra-class variance, so identity prototype embeddings can spread out more within the embedding space.

Poincaré Embeddings Finally, we consider another approach using Poincaré embeddings,

which are designed to encode hierarchical representations by embedding data into hyperbolic space. This allows for encoding of tree-like structures, while approximately preserving distances. In addition, no explicit modeling of tree structure is required to learn hierarchical relationships with Poincaré embeddings, as the structure forms implicitly during training, if it is represented in the data.

To use Poincaré embeddings in UNINEXT and other similar models, we need to replace any embedding relationships of similarity and distance with their equivalents, and replace the optimization, *e.g.* SGD, with its Riemannian SGD equivalent, using a package such as ¹. This would require re-training of any auxiliary networks used for embedding extraction, such as BERT for text embeddings.

6.1.4 Unified LIR with SPNet

By augmenting SPNet for LIR, we can build a model which solves all the desired tasks, with potentially improved training and pre-training. Similar to SEEM and UNINEXT, SPNet has the ability to perform both query-centric and object-centric localization, but using an anchor-based framing instead of a set-based framing. What SPNet is missing, as presented in Chapters 4 and 5, is auxiliary modules for prompt encoding and instance segmentation. To encode different prompt modalities, we can simply adopt prompt encoders from SAM, SEEM, and UNINEXT. Instance segmentation can be performed using a Mask R-CNN approach, with a separate mask prediction branch in parallel with the box prediction and re-id embedding branches. In addition, models like SPNet are uniquely suited to tracking because they can perform track initialization in the object-centric mode, and track known identities in the query-centric mode. Similar to UNINEXT, tracking can be accomplished simply by associating detected instances in neighboring frames, as in [210], [211].

6.2 Generalist Models

A current trend in the deep learning literature broadly is the use of generalist models which can input and output heterogeneous data types. In the context of visual recognition, this type of modeling is a step beyond models such as SAM, SEEM, and UNINEXT, in that the output is more diverse, and can include other localizations such as keypoints, arbitrary polygons, and textual output, *i.e.* captions. These models operate by developing a token vocabulary which can express text, task commands, and image positions, and using a sequence-to-sequence encoder-decoder structure, which solves an array of tasks using sequences of input tokens to produce sequences of output tokens. In Pix2Seqv2 [67], Pix2Seq is extended by allowing output of segmentation masks, object keypoints, and image captions. The Florence-2 model [68] can also take more expressive text as input, allowing for describing what is occurring in specific image sub-regions. Unified-IO [69] can solve even more tasks, including depth estimation, relationship detection, and image inpainting.

¹<https://github.com/facebookresearch/poincare-embeddings>

One of the few tasks these generalist models are missing is dedicated handling of image retrieval, and specifically identity-level re-id. These models struggle with this type of output because it requires fine-grained visual features, and cannot be easily expressed with existing tokens for text and image position information.

In Sec. 6.2.1, we discuss how the Pix2Seq model can be adapted to support person search, and in Sec. 6.2.2, we consider how to adapt generic generalist sequence-to-sequence models for person search.

6.2.1 Pix2Seq for Person Search

We propose a potential method of sequential prediction for person search based on Pix2Seq [23]. While Pix2Seq has already been extended to other seq2seq models which expand its capabilities beyond bounding box and class prediction, no one has yet developed a Pix2Seq variant for person search or other identity-level recognition tasks. We suggest that recent methods of person featurization can be used to construct a token vocabulary capable of richly representing person features, and enabling token-level identity feature prediction.

Learning Person Features To learn high quality person features, we can simply train an existing model for person re-id which produces a spatial feature map. The size of the feature map will determine the number of tokens comprising a person representation, and there is a tradeoff between accurately representing persons, and the efficiency of predicting them, especially in the sequence prediction formulation. As an example, we can use a $h \times w$ feature map, which will cause us to represent each person with $h \cdot w$ tokens. The re-id head of the model will decode this feature map into a single embedding, which will be fed to the re-id loss.

A more advanced approach, tailored to person features specifically, could involve the use of part-based features. Common approaches include dividing image crops into horizontal strips as in [105], or learning visual token groupings corresponding to body regions as in [129].

Tokenizing Features After training the re-id model to convergence, we can extract all feature maps from our training set, and perform dimensionality reduction *e.g.*, with dual PCA, to produce the top- k features which encode the most variance. The value of k represents the size of our token vocabulary, and its selection creates a tradeoff between how well we can represent all the persons in the dataset, and how difficult it will be to learn which tokens represent a given person for the sequence prediction model.

Refining Tokens After selecting k tokens from the initial re-id model, we can optionally refine our tokens to better represent the re-id training data, using an approach similar to the Vector Quantized-Variational AutoEncoder (VQ-VAE) [212], [213]. The VQ-VAE extends the original Variational AutoEncoder (VAE) [214], [215] by introducing a learnable quantized codebook, which forces the generated images to be produced from a discrete space of tokens instead of from a continuous space. This method can be naturally repurposed for our application by replacing the image generation aspect with the re-id embedding head. We can simply swap in the learnable quantized codebook to our already trained re-id model, then initialize the codebook vectors using our selected k tokens. Then, we can fine-tune the

model to better represent the data using only the k learnable vectors, which will become our new token set.

Predicting Tokens To augment Pix2Seq for predicting person tokens, we add the k person feature tokens to our token vocabulary, then define the ordering of output predictions:

$[y_{min}, x_{min}, y_{max}, x_{max}, \text{class}, \text{person_token}_1, \text{person_token}_2, \dots, \text{person_token}_{h \times w}]$

Person tokens will be predicted using softmax probabilities according to their position in the token vocabulary, just like the other tokens. We note the use of quantized image tokens combined with autoregressive learning with a transformer decoder has been previously explored in [216], though in that case still for image generation like VQ-VAE.

Decoding Tokens Finally, to decode the person embedding corresponding to a given token sequence, we simply feed the tokens back to our trained re-id model embedding head. The resulting re-id embedding can be used for retrieval as normal.

Auxiliary Loss Depending on the number of tokens per person, the size of the token vocabulary, and the quantity of data for sequence model training, it is possible that the softmax token prediction mechanism would be insufficient to learn the same rich person features in the rest of the model that the re-id model learned. To augment this process, we can add an additional contrastive loss between token predictions and tokens, to force the sequence model to explicitly learn the token features.

6.2.2 Extending to Generalist Models

Finally, we consider how to combine instance sequence learning with generalist sequence prediction models. In fact, the seq2seq paradigm is so well-equipped for this type of integration, that it would be as easy as adding the tokens from our proposed “Pix2Seq for person search” model to the generalist model token vocabulary, and adding a new prompt token to perform identity search. This showcases a core benefit of this type of model, in that we only need to add a small additional decoder to decode the task output, and no other expensive modules.

6.3 Summary of Main Findings

In this dissertation, we introduce the concept of localized image retrieval as a unifying mechanism for several different problems in visual recognition, including variants of identity search, prompt localization, object detection, and tracking. We analyze efficiency of LIR through the lens of person search, investigating the areas of modular efficiency, algorithmic efficiency, and data efficiency. In doing so, we develop new models which achieve state-of-the-art results on the standard person search benchmark datasets CUHK-SYSU and PRW, and show how these models can be used for LIR more broadly. We show how existing unified models can solve most problems in LIR, but that they do not yet address identity search, and we suggest how this can be added to existing models, and how to adapt SPNet for unified LIR. Finally, we show that generalist sequence-to-sequence models have the capability to solve the mentioned problem types in LIR, including several additional problems in recognition and

generation, and that identity-level recognition should be explicitly added into these models as well via a new tokenization mechanism.

6.4 Societal Impact

Lastly, we discuss potential societal impacts of person search, and of LIR more broadly. It is important to consider the potential negative impact of person search models specifically, since they are ready-made for surveillance applications. This is highlighted by the PRW dataset being entirely composed of surveillance imagery, and the CUHK-SYSU dataset containing many street-view images of pedestrians.

We consider three potential advantages of advancing person search research, and doing so in an open format. First, that person search models can be used for beneficial applications, including public safety, aiding in finding missing persons, and for newly-emerging autonomous systems that interact with humans, *e.g.*, automated vehicles. Second, it allows the research community to understand how the models work at a granular level, and therefore benefits the potential for counteracting negative uses when the technology is abused. Third, it allows government regulatory bodies to have full visibility into what the models can actually do, and how best to impose limitations on what should be available, if some subset of models are deemed harmful to society.

For the broader problem of LIR, and its many subproblems, there are a wide array of applications across the entire domain of visual recognition. Identity search, object discovery, and instance tracking are critical for applications in industrial automation, automated vehicles, and wearable devices. We must also acknowledge that any improvements in LIR are likely to improve person search and tracking as well, so the mentioned caveats apply.

Bibliography

- [1] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, “Joint Detection and Identification Feature Learning for Person Search,” en, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 3376–3385, ISBN: 978-1-5386-0457-1.
- [2] Y. Xu, B. Ma, R. Huang, and L. Lin, “Person Search in a Scene by Jointly Modeling People Commonness and Person Uniqueness,” en, in *Proceedings of the 22nd ACM international conference on Multimedia*, Orlando Florida USA: ACM, Nov. 2014, pp. 937–940, ISBN: 978-1-4503-3063-3.
- [3] L. Jaffe and A. Zakhori, “Gallery Filter Network for Person Search,” in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, ISSN: 2642-9381, Jan. 2023, pp. 1684–1693.
- [4] L. Jaffe and A. Zakhori, “Query-Centric Pre-training for End-to-end Person Search,” *(under review) submitted to: the 18th European Conference on Computer Vision ECCV*, 2024.
- [5] A. Kirillov, E. Mintun, N. Ravi, *et al.*, *Segment Anything*, arXiv:2304.02643 [cs], Apr. 2023.
- [6] X. Zou, J. Yang, H. Zhang, *et al.*, “Segment Everything Everywhere All at Once,” en, in *Thirty-seventh Conference on Neural Information Processing Systems*, Nov. 2023.
- [7] X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang, *SegGPT: Segmenting Everything In Context*, arXiv:2304.03284 [cs], Apr. 2023.
- [8] B. Yan, Y. Jiang, J. Wu, *et al.*, “Universal Instance Perception As Object Discovery and Retrieval,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 325–15 336.
- [9] Z. Yang, B. Gong, L. Wang, W. Huang, D. Yu, and J. Luo, “A Fast and Accurate One-Stage Approach to Visual Grounding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4683–4693.
- [10] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg, “Modeling Context in Referring Expressions,” en, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 69–85, ISBN: 978-3-319-46475-6.

- [11] C. Zhu, Y. Zhou, Y. Shen, *et al.*, “SeqTR: A Simple Yet Universal Network for Visual Grounding,” en, in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 598–615, ISBN: 978-3-031-19833-5.
- [12] Z. Yang, J. Wang, Y. Tang, K. Chen, H. Zhao, and P. H. S. Torr, “LAVT: Language-Aware Vision Transformer for Referring Image Segmentation,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 155–18 165.
- [13] L. Ye, M. Ročan, Z. Liu, and Y. Wang, “Cross-Modal Self-Attention Network for Referring Image Segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 502–10 511.
- [14] A. Botach, E. Zheltonozhskii, and C. Baskin, “End-to-End Referring Video Object Segmentation With Multimodal Transformers,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4985–4995.
- [15] S. Seo, J.-Y. Lee, and B. Han, “URVOS: Unified Referring Video Object Segmentation Network with a Large-Scale Benchmark,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 208–223, ISBN: 978-3-030-58555-6.
- [16] J. Wu, Y. Jiang, P. Sun, Z. Yuan, and P. Luo, “Language As Queries for Referring Video Object Segmentation,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4974–4984.
- [17] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jul. 2017, pp. 936–944.
- [18] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2019, pp. 9626–9635.
- [19] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving Into High Quality Object Detection,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ISSN: 2575-7075, Jun. 2018, pp. 6154–6162.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” en, *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 213–229, ISBN: 978-3-030-58452-8.

- [22] M. Minderer, A. Gritsenko, A. Stone, *et al.*, “Simple Open-Vocabulary Object Detection,” en, in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., vol. 13670, Series Title: Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2022, pp. 728–755, ISBN: 978-3-031-20079-3 978-3-031-20080-9.
- [23] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. Hinton, “Pix2seq: A Language Modeling Framework for Object Detection,” en, in *International Conference on Learning Representations*, Oct. 2021.
- [24] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLACT: Real-Time Instance Segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9157–9166.
- [25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path Aggregation Network for Instance Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.
- [26] Z. Tian, C. Shen, and H. Chen, “Conditional Convolutions for Instance Segmentation,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 282–298, ISBN: 978-3-030-58452-8.
- [27] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “SOLO: Segmenting Objects by Locations,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 649–665, ISBN: 978-3-030-58523-5.
- [28] H. K. Cheng, Y.-W. Tai, and C.-K. Tang, “Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation,” in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 11 781–11 794.
- [29] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, “Video Object Segmentation Using Space-Time Memory Networks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9226–9235.
- [30] N. Xu, L. Yang, Y. Fan, *et al.*, *YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark*, arXiv:1809.03327 [cs], Sep. 2018.
- [31] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, “One-Shot Object Detection with Co-Attention and Co-Excitation,” in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.
- [32] X. Wang, T. Huang, J. Gonzalez, T. Darrell, and F. Yu, “Frustratingly Simple Few-Shot Object Detection,” en, in *Proceedings of the 37th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Nov. 2020, pp. 9919–9928.

- [33] A. Radford, J. W. Kim, C. Hallacy, *et al.*, “Learning Transferable Visual Models From Natural Language Supervision,” en, in *Proceedings of the 38th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jul. 2021, pp. 8748–8763.
- [34] J. Wu, Y. Jiang, S. Bai, W. Zhang, and X. Bai, “SeqFormer: Sequential Transformer for Video Instance Segmentation,” en, in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 553–569, ISBN: 978-3-031-19815-1.
- [35] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-Convolutional Siamese Networks for Object Tracking,” en, in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds., Cham: Springer International Publishing, 2016, pp. 850–865, ISBN: 978-3-319-48881-3.
- [36] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High Performance Visual Tracking With Siamese Region Proposal Network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.
- [37] Y. Wu, J. Lim, and M.-H. Yang, “Object Tracking Benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [38] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking Without Bells and Whistles,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.
- [39] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, *MOT16: A Benchmark for Multi-Object Tracking*, arXiv:1603.00831 [cs], May 2016.
- [40] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking,” en, *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, Nov. 2021.
- [41] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking Objects as Points,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 474–490, ISBN: 978-3-030-58548-8.
- [42] L. Ke, X. Li, M. Danelljan, Y.-W. Tai, C.-K. Tang, and F. Yu, “Prototypical Cross-Attention Networks for Multiple Object Tracking and Segmentation,” in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 1192–1203.
- [43] P. Voigtlaender, M. Krause, A. Osep, *et al.*, “MOTS: Multi-Object Tracking and Segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7942–7951.

- [44] Z. Xu, W. Yang, W. Zhang, X. Tan, H. Huang, and L. Huang, "Segment as Points for Efficient and Effective Online Multi-Object Tracking and Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6424–6437, Oct. 2022, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [45] S. Hwang, M. Heo, S. W. Oh, and S. J. Kim, "Video Instance Segmentation using Inter-Frame Communication Transformers," in *Advances in Neural Information Processing Systems*, vol. 34, Curran Associates, Inc., 2021, pp. 13 352–13 363.
- [46] Y. Wang, Z. Xu, X. Wang, *et al.*, "End-to-End Video Instance Segmentation With Transformers," en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8741–8750.
- [47] L. Yang, Y. Fan, and N. Xu, "Video Instance Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5188–5197.
- [48] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, ISSN: 1063-6919, vol. 1, Jun. 2005, 886–893 vol. 1.
- [49] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [50] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [51] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
- [52] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective Search for Object Recognition," en, *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, Sep. 2013.
- [53] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, Feb. 2020, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [54] W. Liu, D. Anguelov, D. Erhan, *et al.*, "SSD: Single Shot MultiBox Detector," en, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2016, pp. 21–37, ISBN: 978-3-319-46448-0.
- [55] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

- [56] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [57] H. Law and J. Deng, “CornerNet: Detecting Objects as Paired Keypoints,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.
- [58] A. Newell, Z. Huang, and J. Deng, “Associative Embedding: End-to-End Learning for Joint Detection and Grouping,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [59] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint Triplets for Object Detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6569–6578.
- [60] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, “Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9759–9768.
- [61] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [62] H. W. Kuhn, “The Hungarian method for the assignment problem,” en, *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [63] M. J., “Algorithms for the Assignment and Transportation Problems,” *Society for Industrial and Applied Mathematics*, vol. 15, pp. 196–210, 1962.
- [64] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, “Microsoft COCO: Common Objects in Context,” en, in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 740–755, ISBN: 978-3-319-10602-1.
- [65] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, *Deformable DETR: Deformable Transformers for End-to-End Object Detection*, arXiv:2010.04159 [cs], Mar. 2021.
- [66] P. Sun, R. Zhang, Y. Jiang, *et al.*, “Sparse R-CNN: End-to-End Object Detection With Learnable Proposals,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 454–14 463.
- [67] T. Chen, S. Saxena, L. Li, T.-Y. Lin, D. J. Fleet, and G. E. Hinton, “A Unified Sequence Interface for Vision Tasks,” en, *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 333–31 346, Dec. 2022.
- [68] B. Xiao, H. Wu, W. Xu, *et al.*, *Florence-2: Advancing a Unified Representation for a Variety of Vision Tasks*, en, Nov. 2023.

- [69] J. Lu, C. Clark, R. Zellers, R. Mottaghi, and A. Kembhavi, “UNIFIED-IO: A Unified Model for Vision, Language, and Multi-modal Tasks,” en, in *The Eleventh International Conference on Learning Representations*, Sep. 2022.
- [70] P. Wang, A. Yang, R. Men, *et al.*, “OFA: Unifying Architectures, Tasks, and Modalities Through a Simple Sequence-to-Sequence Learning Framework,” en, in *Proceedings of the 39th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Jun. 2022, pp. 23 318–23 340.
- [71] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.
- [72] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality Reduction by Learning an Invariant Mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, ISSN: 1063-6919, vol. 2, Jun. 2006, pp. 1735–1742.
- [73] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” en, in *Proceedings of the 37th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Nov. 2020, pp. 1597–1607.
- [74] A. v. d. Oord, Y. Li, and O. Vinyals, *Representation Learning with Contrastive Predictive Coding*, arXiv:1807.03748 [cs, stat], Jan. 2019.
- [75] K. Sohn, “Improved Deep Metric Learning with Multi-class N-pair Loss Objective,” in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016.
- [76] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep Hypersphere Embedding for Face Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 212–220.
- [77] H. Wang, Y. Wang, Z. Zhou, *et al.*, “CosFace: Large Margin Cosine Loss for Deep Face Recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.
- [78] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.
- [79] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 1063-6919, Jun. 2015, pp. 815–823.
- [80] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable Person Re-identification: A Benchmark,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Dec. 2015, pp. 1116–1124.

- [81] Y. Bai, J. Jiao, W. Ce, *et al.*, “Person30K: A Dual-Meta Generalization Network for Person Re-Identification,” *en*, 2021, pp. 2123–2132.
- [82] S. Yu, S. Li, D. Chen, R. Zhao, J. Yan, and Y. Qiao, “COCAS: A Large-Scale Clothes Changing Person Dataset for Re-Identification,” *en*, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 3397–3406, ISBN: 978-1-72817-168-5.
- [83] X. Liu, W. Liu, H. Ma, and H. Fu, “Large-scale vehicle re-identification in urban surveillance videos,” in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, ISSN: 1945-788X, Jul. 2016, pp. 1–6.
- [84] Y. Lou, Y. Bai, J. Liu, S. Wang, and L. Duan, “VERI-Wild: A Large Dataset and a New Method for Vehicle Re-Identification in the Wild,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3235–3243.
- [85] H. Liu, Y. Tian, Y. Yang, L. Pang, and T. Huang, “Deep Relative Distance Learning: Tell the Difference Between Similar Vehicles,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2167–2175.
- [86] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, “Deep Learning for Person Re-Identification: A Survey and Outlook,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2872–2893, Jun. 2022, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [87] Y. Huang, Z.-J. Zha, X. Fu, R. Hong, and L. Li, “Real-World Person Re-Identification via Degradation Invariance Learning,” *en*, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 14 072–14 082, ISBN: 978-1-72817-168-5.
- [88] R. Yu, Z. Dou, S. Bai, Z. Zhang, Y. Xu, and X. Bai, “Hard-Aware Point-to-Set Deep Metric for Person Re-identification,” *en*, in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11220, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 196–212, ISBN: 978-3-030-01269-4 978-3-030-01270-0.
- [89] Y.-T. Cao, J. Wang, and D. Tao, “Symbiotic Adversarial Learning for Attribute-Based Person Search,” *en*, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12359, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 230–247, ISBN: 978-3-030-58567-9 978-3-030-58568-6.
- [90] Y. Zou, X. Yang, Z. Yu, B. V. K. V. Kumar, and J. Kautz, “Joint Disentangling and Adaptation for Cross-Domain Person Re-Identification,” *en*, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12347, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 87–104, ISBN: 978-3-030-58535-8 978-3-030-58536-5.

- [91] L. He and W. Liu, “Guided Saliency Feature Learning for Person Re-identification in Crowded Scenes,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12373, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 357–373, ISBN: 978-3-030-58603-4 978-3-030-58604-1.
- [92] K. Han, Y. Huang, Z. Chen, L. Wang, and T. Tan, “Prediction and Recovery for Adaptive Low-Resolution Person Re-Identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12371, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 193–209, ISBN: 978-3-030-58573-0 978-3-030-58574-7.
- [93] M. Ye, J. Shen, D. J. Crandall, L. Shao, and J. Luo, “Dynamic Dual-Attentive Aggregation Learning for Visible-Infrared Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12362, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 229–247, ISBN: 978-3-030-58519-8 978-3-030-58520-4.
- [94] C. Luo, C. Song, and Z. Zhang, “Generalizing Person Re-Identification by Camera-Aware Invariance Learning and Cross-Domain Mixup,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12360, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 224–241, ISBN: 978-3-030-58554-9 978-3-030-58555-6.
- [95] Z. Zhuang, L. Wei, L. Xie, *et al.*, “Rethinking the Distribution Gap of Person Re-identification with Camera-Based Batch Normalization,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12357, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 140–157, ISBN: 978-3-030-58609-6 978-3-030-58610-2.
- [96] S. Liao and L. Shao, “Interpretable and Generalizable Person Re-identification with Query-Adaptive Convolution and Temporal Lifting,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12356, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 456–474, ISBN: 978-3-030-58620-1 978-3-030-58621-8.
- [97] Y. Zhai, Q. Ye, S. Lu, M. Jia, R. Ji, and Y. Tian, “Multiple Expert Brainstorming for Domain Adaptive Person Re-Identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12352, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 594–611, ISBN: 978-3-030-58570-9 978-3-030-58571-6.
- [98] S. Zhao, C. Gao, J. Zhang, *et al.*, “Do Not Disturb Me: Person Re-identification Under the Interference of Other Pedestrians,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12351, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 647–663, ISBN: 978-3-030-58538-9 978-3-030-58539-6.

- [99] K. Zhu, H. Guo, Z. Liu, M. Tang, and J. Wang, “Identity-Guided Human Semantic Parsing for Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12348, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 346–363, ISBN: 978-3-030-58579-2 978-3-030-58580-8.
- [100] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, “Person Re-identification in the Wild,” en, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 3346–3355, ISBN: 978-1-5386-0457-1.
- [101] D. Chen, S. Zhang, W. Ouyang, J. Yang, and Y. Tai, “Person Search via a Mask-Guided Two-Stream CNN Model,” en, in *Computer Vision - ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 764–781, ISBN: 978-3-030-01234-2.
- [102] X. Lan, X. Zhu, and S. Gong, “Person Search by Multi-Scale Matching,” en, in *Computer Vision - ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 553–569, ISBN: 978-3-030-01246-5.
- [103] C. Han, J. Ye, Y. Zhong, *et al.*, “Re-ID Driven Localization Refinement for Person Search,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2019, pp. 9813–9822.
- [104] H. Liu, J. Feng, Z. Jie, *et al.*, “Neural Person Search Machines,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2017, pp. 493–501.
- [105] Y. Zhong, X. Wang, and S. Zhang, “Robust Partial Matching for Person Search in the Wild,” en, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 6826–6834, ISBN: 978-1-72817-168-5.
- [106] D. Chen, S. Zhang, W. Ouyang, J. Yang, and B. Schiele, “Hierarchical Online Instance Matching for Person Search,” en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 10 518–10 525, Apr. 2020, Number: 07.
- [107] W. Dong, Z. Zhang, C. Song, and T. Tan, “Bi-Directional Interaction Network for Person Search,” en, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 2836–2845, ISBN: 978-1-72817-168-5.
- [108] D. Chen, S. Zhang, J. Yang, and B. Schiele, “Norm-Aware Embedding for Efficient Person Search,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2020, pp. 12 612–12 621.

- [109] X. Zhang, X. Wang, J.-W. Bian, C. Shen, and M. You, “Diverse Knowledge Distillation for End-to-End Person Search,” en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3412–3420, May 2021, Number: 4.
- [110] Y. Yan, J. Li, J. Qin, *et al.*, “Anchor-Free Person Search,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2021, pp. 7686–7695.
- [111] Z. Li and D. Miao, “Sequential End-to-end Network for Efficient Person Search,” en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, pp. 2011–2019, May 2021, Number: 3.
- [112] R. Yu, D. Du, R. LaLonde, *et al.*, “Cascade Transformers for End-to-End Person Search,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7267–7276.
- [113] J. Cao, Y. Pang, R. M. Anwer, *et al.*, “PSTR: End-to-End One-Step Person Search With Transformers,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9458–9467.
- [114] Z. Dong, J. Cao, R. M. Anwer, J. Xie, F. Khan, and Y. Pang, *LEAPS: End-to-End One-Step Person Search With Learnable Proposals*, arXiv:2303.11859 [cs], Mar. 2023.
- [115] Y. Li, H. Xu, M. Bian, and J. Xiao, “Cross-scale global attention feature pyramid network for person search,” *Image and Vision Computing*, vol. 116, p. 104332, Dec. 2021.
- [116] P. Zheng, J. Qin, Y. Yan, *et al.*, “Global-local context network for person search,” *arXiv preprint arXiv:2112.02500*, vol. 8, 2021.
- [117] X. Chang, P.-Y. Huang, Y.-D. Shen, X. Liang, Y. Yang, and A. G. Hauptmann, “RCAA: Relational Context-Aware Agents for Person Search,” en, in *Computer Vision - ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11213, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 86–102, ISBN: 978-3-030-01239-7 978-3-030-01240-3.
- [118] B. Munjal, S. Amin, F. Tombari, and F. Galasso, “Query-Guided End-To-End Person Search,” en, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 811–820, ISBN: 978-1-72813-293-8.
- [119] B. Munjal, A. Flaborea, S. Amin, F. Tombari, and F. Galasso, “Query-guided networks for few-shot fine-grained classification and person search,” *Pattern Recognition*, vol. 133, p. 109049, Jan. 2023.
- [120] W. Dong, Z. Zhang, C. Song, and T. Tan, “Instance Guided Proposal Network for Person Search,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2020, pp. 2582–2591.

- [121] C. Wang, B. Ma, H. Chang, S. Shan, and X. Chen, “TCTS: A Task-Consistent Two-Stage Framework for Person Search,” en, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 11 949–11 958, ISBN: 978-1-72817-168-5.
- [122] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, ISSN: 2575-7075, Jun. 2018, pp. 7132–7141.
- [123] C. Jia, M. Luo, C. Yan, L. Zhu, X. Chang, and Q. Zheng, “Collaborative Contrastive Refining for Weakly Supervised Person Search,” *IEEE Transactions on Image Processing*, vol. 32, pp. 4951–4963, 2023, Conference Name: IEEE Transactions on Image Processing.
- [124] B.-J. Han, “Context-Aware Unsupervised Clustering for Person Search,” en, *Proceedings of BMVC*, 2021.
- [125] J. Wang, Y. Pang, J. Cao, H. Sun, Z. Shao, and X. Li, *Deep Intra-Image Contrastive Learning for Weakly Supervised One-Step Person Search*, en, arXiv:2302.04607 [cs], Feb. 2023.
- [126] Y. Yan, J. Li, S. Liao, *et al.*, “Exploring Visual Context for Weakly Supervised Person Search,” en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, pp. 3027–3035, Jun. 2022, Number: 3.
- [127] B. Wang, Y. Yang, J. Wu, G.-j. Qi, and Z. Lei, *Self-similarity Driven Scale-invariant Learning for Weakly Supervised Person Search*, en, arXiv:2302.12986 [cs], Feb. 2023.
- [128] C. Han, K. Su, D. Yu, *et al.*, “Weakly Supervised Person Search with Region Siamese Networks,” en, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 11 986–11 995, ISBN: 978-1-66542-812-5.
- [129] W. Chen, X. Xu, J. Jia, *et al.*, “Beyond Appearance: A Semantic Controllable Self-Supervised Learning Framework for Human-Centric Visual Tasks,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 050–15 061.
- [130] D. Fu, D. Chen, J. Bao, *et al.*, “Unsupervised Pre-Training for Person Re-Identification,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 750–14 759.
- [131] M. Caron, H. Touvron, I. Misra, *et al.*, “Emerging Properties in Self-Supervised Vision Transformers,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2021, pp. 9630–9640.
- [132] C. Jia, M. Luo, Z. Dang, *et al.*, *PSDiff: Diffusion Model for Person Search with Iterative and Collaborative Refinement*, arXiv:2309.11125 [cs], Sep. 2023.

- [133] S. Chen, P. Sun, Y. Song, and P. Luo, “DiffusionDet: Diffusion Model for Object Detection,” en, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 830–19 843.
- [134] A. Neubeck and L. Van Gool, “Efficient Non-Maximum Suppression,” in *18th International Conference on Pattern Recognition (ICPR’06)*, ISSN: 1051-4651, vol. 3, Aug. 2006, pp. 850–855.
- [135] H. Kim, S. Joung, I.-J. Kim, and K. Sohn, “Prototype-Guided Saliency Feature Learning for Person Search,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2021, pp. 4863–4872.
- [136] Y. Yan, Q. Zhang, B. Ni, W. Zhang, M. Xu, and X. Yang, “Learning Context Graph for Person Search,” en, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 2153–2162, ISBN: 978-1-72813-293-8.
- [137] J. Xiao, Y. Xie, T. Tillo, K. Huang, Y. Wei, and J. Feng, “IAN: The Individual Aggregation Network for Person Search,” en, *Pattern Recognition*, vol. 87, pp. 332–340, Mar. 2019.
- [138] C. Han, Z. Zheng, C. Gao, N. Sang, and Y. Yang, “Decoupled and Memory-Reinforced Networks: Towards Effective Feature Learning for One-Step Person Search,” en, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1505–1512, May 2021, Number: 2.
- [139] J. Yang and X. Wang, “Multi-Attention-Guided Cascading Network for End-to-End Person Search,” en, *Applied Sciences*, vol. 13, no. 9, p. 5576, Jan. 2023, Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.
- [140] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” en, in *International Conference on Learning Representations*, Jul. 2022.
- [141] G. Hinton, O. Vinyals, and J. Dean, *Distilling the Knowledge in a Neural Network*, arXiv:1503.02531 [cs, stat], Mar. 2015.
- [142] L. Wang and K.-J. Yoon, “Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3048–3068, Jun. 2022, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [143] C. Jia, M. Luo, Z. Dang, X. Chang, and Q. Zheng, “Towards Real-Time Person Search with Invariant Feature Learning,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.
- [144] L. Vilnis, X. Li, S. Murty, and A. McCallum, “Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 263–272.

- [145] T. Chheda, P. Goyal, T. Tran, *et al.*, “Box Embeddings: An open-source library for representation learning using geometric structures,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, H. Adel and S. Shi, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 203–211.
- [146] M. Nickel and D. Kiela, “Poincaré Embeddings for Learning Hierarchical Representations,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [147] L. Chen and J. Xu, *Sequential Transformer for End-to-End Person Search*, arXiv:2211.04323 [cs], Nov. 2022.
- [148] M. Fiaz, H. Cholakkal, R. M. Anwer, and F. S. Khan, “SAT: Scale-Augmented Transformer for Person Search,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 4820–4829.
- [149] G. M. van de Ven and A. S. Tolias, *Three scenarios for continual learning*, arXiv:1904.07734 [cs, stat], Apr. 2019.
- [150] P. Zhang, X. Bai, J. Zheng, and X. Ning, *Towards Fully Decoupled End-to-End Person Search*, arXiv:2309.04967 [cs], Sep. 2023.
- [151] B.-J. Han, K. Ko, and J.-Y. Sim, “End-to-End Trainable Trident Person Search Network Using Adaptive Gradient Propagation,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2021, pp. 905–913.
- [152] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, “Big Self-Supervised Models are Strong Semi-Supervised Learners,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 22 243–22 255.
- [153] O. Henaff, “Data-Efficient Image Recognition with Contrastive Predictive Coding,” in *Proceedings of the 37th International Conference on Machine Learning*, ISSN: 2640-3498, PMLR, Nov. 2020, pp. 4182–4192.
- [154] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 9726–9735, ISBN: 978-1-72817-168-5.
- [155] X. Chen, H. Fan, R. Girshick, and K. He, “Improved Baselines with Momentum Contrastive Learning,” Tech. Rep., Mar. 2020, Publication Title: arXiv e-prints ADS Bibcode: 2020arXiv200304297C Type: article.
- [156] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 9912–9924.

- [157] Z. Xie, Z. Zhang, Y. Cao, *et al.*, “SimMIM: A Simple Framework for Masked Image Modeling,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN: 2575-7075, Jun. 2022, pp. 9643–9653.
- [158] A. v. d. Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, *Conditional Image Generation with PixelCNN Decoders*, Number: arXiv:1606.05328 arXiv:1606.05328 [cs], Jun. 2016.
- [159] X. Chen, S. Xie, and K. He, “An Empirical Study of Training Self-Supervised Vision Transformers,” English, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE Computer Society, Oct. 2021, pp. 9620–9629, ISBN: 978-1-66542-812-5.
- [160] J.-B. Grill, F. Strub, F. Altché, *et al.*, “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 21 271–21 284.
- [161] X. Chen, M. Ding, X. Wang, *et al.*, *Context Autoencoder for Self-Supervised Representation Learning*, arXiv:2202.03026 [cs], May 2022.
- [162] Z. Ji, X. Zou, X. Lin, X. Liu, T. Huang, and S. Wu, “An Attention-Driven Two-Stage Clustering Method for Unsupervised Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12373, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 20–36, ISBN: 978-3-030-58603-4 978-3-030-58604-1.
- [163] D. Mekhazni, A. Bhuiyan, G. Ekladios, and E. Granger, “Unsupervised Domain Adaptation in the Dissimilarity Space for Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12372, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 159–174, ISBN: 978-3-030-58582-2 978-3-030-58583-9.
- [164] F. Zhao, S. Liao, G.-S. Xie, J. Zhao, K. Zhang, and L. Shao, “Unsupervised Domain Adaptation with Noise Resistible Mutual-Training for Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12356, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 526–544, ISBN: 978-3-030-58620-1 978-3-030-58621-8.
- [165] G. Chen, Y. Lu, J. Lu, and J. Zhou, “Deep Credible Metric Learning for Unsupervised Domain Adaptation Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12353, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 643–659, ISBN: 978-3-030-58597-6 978-3-030-58598-3.

- [166] Y. Lin, L. Xie, Y. Wu, C. Yan, and Q. Tian, “Unsupervised Person Re-Identification via Softened Similarity Learning,” en, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 3387–3396, ISBN: 978-1-72817-168-5.
- [167] X. Jin, C. Lan, W. Zeng, and Z. Chen, “Global Distance-Distributions Separation for Unsupervised Person Re-identification,” en, in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., vol. 12352, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 735–751, ISBN: 978-3-030-58570-9 978-3-030-58571-6.
- [168] D. Fu, D. Chen, J. Bao, *et al.*, “Unsupervised Pre-Training for Person Re-Identification,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 750–14 759.
- [169] D. Fu, D. Chen, H. Yang, *et al.*, “Large-Scale Pre-Training for Person Re-Identification With Noisy Labels,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2476–2486.
- [170] H. Luo, P. Wang, Y. Xu, *et al.*, *Self-Supervised Pre-Training for Transformer-Based Person Re-Identification*, arXiv:2111.12084 [cs], Nov. 2021.
- [171] Y. Cho, W. J. Kim, S. Hong, and S.-E. Yoon, “Part-Based Pseudo Label Refinement for Unsupervised Person Re-Identification,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7308–7318.
- [172] Z. Yang, X. Jin, K. Zheng, and F. Zhao, “Unleashing Potential of Unsupervised Pre-Training With Intra-Identity Regularization for Person Re-Identification,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 298–14 307.
- [173] K. Zhu, H. Guo, T. Yan, Y. Zhu, J. Wang, and M. Tang, “PASS: Part-Aware Self-Supervised Pre-Training for Person Re-Identification,” en, in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., ser. Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2022, pp. 198–214, ISBN: 978-3-031-19781-9.
- [174] A. Gokul, K. Kallidromitis, S. Li, *et al.*, *Refine and Represent: Region-to-Object Representation Learning*, arXiv:2208.11821 [cs], Aug. 2022.
- [175] C. Yang, Z. Wu, B. Zhou, and S. Lin, “Instance Localization for Self-supervised Detection Pretraining,” en, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 3986–3995, ISBN: 978-1-66544-509-2.
- [176] A. Ziegler and Y. M. Asano, “Self-Supervised Learning of Object Parts for Semantic Segmentation,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 502–14 511.

- [177] G. Huang, I. Laradji, D. Vázquez, S. Lacoste-Julien, and P. Rodríguez, “A Survey of Self-Supervised and Few-Shot Object Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [178] Z. Dai, B. Cai, Y. Lin, and J. Chen, “UP-DETR: Unsupervised Pre-training for Object Detection with Transformers,” en, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 1601–1610, ISBN: 978-1-66544-509-2.
- [179] A. Bar, X. Wang, V. Kantorov, *et al.*, “DETReg: Unsupervised Pretraining With Region Priors for Object Detection,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 605–14 615.
- [180] T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell, *Region Similarity Representation Learning*, Number: arXiv:2103.12902 arXiv:2103.12902 [cs], Aug. 2021.
- [181] G. Alain and Y. Bengio, *Understanding intermediate layers using linear classifier probes*, arXiv:1610.01644 [cs, stat], Nov. 2018.
- [182] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, *Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution*, arXiv:2202.10054 [cs], Feb. 2022.
- [183] Z. Li and D. Hoiem, “Learning without Forgetting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [184] G. Oren and L. Wolf, “In Defense of the Learning Without Forgetting for Task Incremental Learning,” en, in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada: IEEE, Oct. 2021, pp. 2209–2218, ISBN: 978-1-66540-191-3.
- [185] E. J. Hu, Y. Shen, P. Wallis, *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models,” en, in *International Conference on Learning Representations*, Jan. 2022.
- [186] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, *Scaling Laws for Neural Language Models*, arXiv:2001.08361 [cs, stat], Jan. 2020.
- [187] J. Hoffmann, S. Borgeaud, A. Mensch, *et al.*, “An empirical analysis of compute-optimal large language model training,” en, *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 016–30 030, Dec. 2022.
- [188] Z. Xie, Z. Zhang, Y. Cao, *et al.*, *On Data Scaling in Masked Image Modeling*, arXiv:2206.04664 [cs], Jun. 2022.
- [189] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling Vision Transformers,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.

- [190] OpenAI, J. Achiam, S. Adler, *et al.*, *GPT-4 Technical Report*, arXiv:2303.08774 [cs], Dec. 2023.
- [191] J. Betker, G. Goh, L. Jing, *et al.*, “Improving image generation with better captions,” *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, vol. 2, no. 3, p. 8, 2023.
- [192] L. Jaffe and A. Zakhor, *Gallery Filter Network for Person Search*, arXiv:2210.12903 [cs], Oct. 2022.
- [193] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” en, *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
- [194] S. Shao, Z. Li, T. Zhang, *et al.*, “Objects365: A Large-Scale, High-Quality Dataset for Object Detection,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2019, pp. 8429–8438.
- [195] S. Shao, Z. Zhao, B. Li, *et al.*, *CrowdHuman: A Benchmark for Detecting Human in a Crowd*, arXiv:1805.00123 [cs], Apr. 2018.
- [196] S. Zhang, R. Benenson, and B. Schiele, “CityPersons: A Diverse Dataset for Pedestrian Detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3213–3221.
- [197] Y. Pang, J. Cao, Y. Li, J. Xie, H. Sun, and J. Gong, “TJU-DHD: A Diverse High-Resolution Dataset for Object Detection,” *IEEE Transactions on Image Processing*, vol. 30, pp. 207–219, 2021, Conference Name: IEEE Transactions on Image Processing.
- [198] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” en, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778, ISBN: 978-1-4673-8851-1.
- [199] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, arXiv:2010.11929 [cs], Jun. 2021.
- [200] Z. Liu, Y. Lin, Y. Cao, *et al.*, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, ISSN: 2380-7504, Oct. 2021, pp. 9992–10 002.
- [201] W. Wang, E. Xie, X. Li, *et al.*, “PVT v2: Improved baselines with Pyramid Vision Transformer,” en, *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.
- [202] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.
- [203] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980 [cs], Jan. 2017.

- [204] O. Russakovsky, J. Deng, H. Su, *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” en, *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [205] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” en, in *Proceedings of the 32nd International Conference on Machine Learning*, ISSN: 1938-7228, PMLR, Jun. 2015, pp. 448–456.
- [206] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” en, in *International Conference on Learning Representations*, Feb. 2022.
- [207] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” en, in *International Conference on Learning Representations*, Nov. 2016.
- [208] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805 [cs], May 2019.
- [209] T. Brown, B. Mann, N. Ryder, *et al.*, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.
- [210] J. Pang, L. Qiu, X. Li, *et al.*, “Quasi-Dense Similarity Learning for Multiple Object Tracking,” en, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 164–173.
- [211] J. Wu, Q. Liu, Y. Jiang, S. Bai, A. Yuille, and X. Bai, “In Defense of Online Models for Video Instance Segmentation,” en, in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 588–605, ISBN: 978-3-031-19815-1.
- [212] A. van den Oord, O. Vinyals, and k. kavukcuoglu koray, “Neural Discrete Representation Learning,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [213] A. Razavi, A. van den Oord, and O. Vinyals, “Generating Diverse High-Fidelity Images with VQ-VAE-2,” in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.
- [214] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, arXiv:1312.6114 [cs, stat], Dec. 2022.
- [215] D. P. Kingma and M. Welling, “An Introduction to Variational Autoencoders,” *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019, arXiv:1906.02691 [cs, stat].
- [216] J. Yu, X. Li, J. Y. Koh, *et al.*, “Vector-quantized Image Modeling with Improved VQGAN,” en, in *International Conference on Learning Representations*, Oct. 2021.

- [217] A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin, “Albumentations: Fast and flexible image augmentations,” *Information*, vol. 11, no. 2, p. 125, Feb. 2020, arXiv:1809.06839 [cs].
- [218] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, *Tune: A Research Platform for Distributed Model Selection and Training*, arXiv:1807.05118 [cs, stat], Jul. 2018.

Appendix A

Supplementary Material: Modular Efficiency

A.1 Data Processing and Evaluation

We make publicly available our codebase¹, which includes instructions and config files needed to replicate all main experiments of the related paper [3]. For comparative purposes, we implicitly refer in the following subsections to the public codebases of OIM² [1], NAE³ [108], SeqNet⁴ [111], COAT⁵ [112], AlignPS⁶ [110], and PSTr⁷ [113].

A.1.1 Standardized Data Format

We produce an intermediate COCO-style [64] format for all partitions of the CUHK-SYSU and PRW datasets. In addition to standard COCO object metadata, we include `person_id` and `is_known` fields for persons, and a `cam_id` image field for performing cross-camera evaluation.

This standardization process made it straightforward to prepare new partitions of the data. In particular, we split the standard training sets into separate training and validation sets, and created some additional smaller debugging sets. This allowed us to pick hyperparameters without fitting to the test data.

We also standardize the format of retrieval partitions into three categories: 1) fully-specified format which encodes the exact gallery scenes to be used for each query 2) format which specifies queries only, and uses all scenes in the partition as the gallery and 3) format

¹<https://github.com/LukeJaffe/GFN>

²https://github.com/ShuangLI59/person_search

³<https://github.com/dichen-cd/NAE4PS>

⁴<https://github.com/serendipity/SeqNet>

⁵<https://github.com/Kitware/COAT>

⁶<https://github.com/daodaofr/AlignPS>

⁷<https://github.com/JialeCao001/PSTR>

	CUHK-SYSU				PRW			
Metadata	trainval	test	train	val	trainval	test	train	val
Scenes	11,206	6,978	8,964	2,242	5,704	6,112	4,563	1,141
Boxes	55,272	40,871	44,244	11,028	18,048	25,062	14,897	3,151
Known IDs	5,532	2,900	4,296	1,236	483	544	424	158
Known Boxes	15,085	8,345	11,623	3,462	14,907	19,127	12,125	2,782
Unknown Boxes	40,187	32,526	32,621	7,566	3,141	5,935	2,772	369

Table A.1: Dataset metadata showing how many scenes, boxes, and person IDs are in each partition.

which uses all possible queries, and all possible scenes as the gallery. We create the second and third formats because it is otherwise inefficient to fully-specify the “all” cases.

A.1.2 Training and Validation Sets

For both datasets, known identity sets between the train and test partitions are disjoint, making the standard evaluation an *open-set* retrieval problem. To construct the training and validation sets to mirror the open-set retrieval problem of the standard train-test divide, we build a graph based on which scenes share common person identities. Two nodes (scenes) have an edge between them if they share at least one person identity in common. In this way, we can easily split the CUHK-SYSU dataset into a set of connected components, and divide those components into two groups for train ($\sim 80\%$) and val ($\sim 20\%$).

Since the PRW dataset comprises video surveillance footage, this graph has the property that nearly every scene is connected to another scene via some common person identity. Therefore, we ignore the top 100 most common person identities when constructing the graph for PRW, resulting in a partition which is not quite open-set, but should exhibit similar generalization properties for the purpose of model development. For PRW, we also divide components into two groups for train ($\sim 80\%$) and val ($\sim 20\%$).

We rename the original train set to “trainval”, and all of our final experimental results in this paper are from models trained on the full trainval set, and tested on the full test set using the standard retrieval scenarios.

A.1.3 Partition Information

Metadata for the exact breakdown of known and unknown identities and boxes for each partition is given in Table A.1.

A.1.4 Evaluation Functions

Using these standardized partitions, we are able to use just one function for detection evaluation and one for retrieval evaluation, as opposed to separate functions for each dataset. This also makes it easier to add in method-specific metrics that can be immediately tested for all partitions.

We note that the current dataset releases for PRW and CUHK-SYSU have a small number (5 or less) of the following errors: duplicate bounding boxes in a single scene, repeated person ids in a single scene, and repeated gallery scenes in a retrieval partition. Although these issues are not handled correctly by the standard evaluation function, we exactly replicate the previous erroneous behavior in our new evaluation function to be certain the comparison against other methods is fair. We leave correction of the underlying data and evaluation function to future work.

A.1.5 Augmentation Code Structure

To make use of augmentation strategies in the albumentations library [217], we refactor evaluation to occur on the augmented data instead of the original data. This allows for easy inclusion of different resizing and cropping strategies which we make use of, in addition to a wealth of other augmentations, experimenting with which we leave to future work.

A.1.6 Config Format and Ray Tune

For running experiments with our code, we provide a YAML config format which is compatible with the Ray Tune library [218]. We specifically support the `tune.grid_search` functionality by parsing lists in the YAML file as inputs to this function. This makes it easy to run ablations with many variations using a single config file.