

支持深度学习的视觉数据库管理系统研究进展*

丁光耀^{1,2}, 徐辰^{1,2}, 钱卫宁^{1,2}, 周傲英^{1,2}



¹(华东师范大学 数据科学与工程学院, 上海 200062)

²(上海市大数据管理系统工程研究中心(华东师范大学), 上海 200062)

通信作者: 徐辰, E-mail: cxu@dase.ecnu.edu.cn

摘要: 计算机视觉因其强大的学习能力, 在各种真实场景中得到了广泛应用. 随着数据库的发展, 利用数据库中成熟的数据管理技术来处理视觉分析应用, 已成为一种日益增长的研究趋势. 图像、视频和文本等多模态数据的相互融合处理, 也促进了视觉分析应用的多样性和准确性. 近年来, 因深度学习的兴起, 支持深度学习的视觉分析应用开始受到广泛关注. 然而, 传统的数据库管理技术在深度学习场景下面临着复杂视觉分析语义难以表达、应用执行效率低等问题. 因此, 支持深度学习的视觉数据库管理系统得到了广泛关注. 综述了目前视觉数据库管理系统的研究进展: 首先, 总结了视觉数据库管理系统在不同层面上面临的挑战, 包括编程接口、查询优化、执行调度和数据存储; 其次, 分别探讨了上述4个层面上的相关技术; 最后, 对视觉数据库管理系统未来的研究方向进行了展望.

关键词: 深度学习; 视觉分析; 数据库管理系统

中图法分类号: TP311

中文引用格式: 丁光耀, 徐辰, 钱卫宁, 周傲英. 支持深度学习的视觉数据库管理系统研究进展. 软件学报, 2024, 35(3): 1207-1230. <http://www.jos.org.cn/1000-9825/7075.htm>

英文引用格式: Ding GY, Xu C, Qian WN, Zhou AY. Survey on Vision Database Management Systems Supporting Deep Learning. Ruan Jian Xue Bao/Journal of Software, 2024, 35(3): 1207-1230 (in Chinese). <http://www.jos.org.cn/1000-9825/7075.htm>

Research Progress on Vision Database Management Systems Supporting Deep Learning

DING Guang-Yao^{1,2}, XU Chen^{1,2}, QIAN Wei-Ning^{1,2}, ZHOU Ao-Ying^{1,2}

¹(School of Data Science and Engineering, East China Normal University, Shanghai 200062, China)

²(Shanghai Engineering Research Center on Big Data Management (East China Normal University), Shanghai 200062, China)

Abstract: Computer vision has been widely used in various real-world scenarios due to its powerful learning ability. With the development of databases, there is a growing trend in research to exploit mature data management techniques in databases for vision analytics applications. The integration and processing of multimodal data, including images, video and text, promotes diversity and improves accuracy in vision analytics applications. In recent years, due to the popularization of deep learning, there has been a growing interest in vision analytics applications that support deep learning. Nevertheless, traditional database management techniques in deep learning scenarios suffer from the issues such as lack of semantics for vision analytics and inefficiency in application execution. Hence, vision database management systems that support deep learning have been widely studied. This study reviews the progress of vision database management systems. First, this study summarizes the challenges faced by vision database management systems in different dimensions, including programming interface, query optimization, execution scheduling, and data storage. Second, this study discusses the technologies in each of these four dimensions. Finally, the study investigates the future research directions of vision database management systems.

* 基金项目: 国家自然科学基金(62272168); 上海市自然科学基金(23ZR1419900)

本文由“面向多模态数据的新型数据库技术”专题特约编辑彭智勇教授、高云君教授、李国良教授、许建秋教授推荐.

收稿时间: 2023-07-17; 修改时间: 2023-09-05; 采用时间: 2023-10-24; jos 在线出版时间: 2023-11-08

CNKI 网络首发时间: 2023-12-25

Key words: deep learning; vision analytics; database management system

计算机视觉是一门涉及图像和视频处理、模式识别和机器学习的领域,其主要目标在于使计算机具备理解和解释视觉相关数据的能力,以模拟人类视觉系统的功能.因其对数据的准确理解,计算机视觉实现了对图像检测识别、视频动作预测以及语义分割等任务的成熟支持^[1].随着成像设备和传感器的普及,视觉数据的规模快速增长,类型也呈现多样化(包括图像、视频和文本等).事实上,计算机视觉分析应用所需的数据往往不再局限于单一的图像或视频数据,而是呈现出多模态趋势.这些多种类型数据的互补融合,能够使应用获得更准确的查询结果.因此,为了充分管理和利用多模态数据以执行各类视觉分析应用,一些多媒体数据库管理系统逐渐崭露头角^[2].如图 1 所示,这些系统包括图像数据库管理系统(image database management system)^[3,4]和视频数据库管理系统(video database management system)^[5,6].本文将这两类系统统称为视觉数据库管理系统(vision database management system).

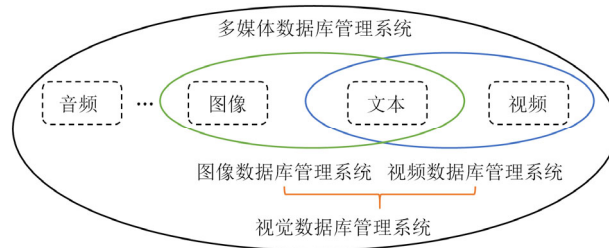


图 1 视觉数据库管理系统的范围

如图 2 所示,根据视觉分析应用的发展历程,我们将其划分为两个阶段.

- 首先,在 20 世纪末,数据管理研究人员开始认识到计算机视觉在查询视觉数据方面的潜力,并趋向于结合数据库中的数据管理技术来处理视觉分析应用.这一阶段的视觉分析应用主要侧重于根据特征相似度来进行图像检索^[7],而这些特征则通常利用传统的机器学习技术(如支持向量机 SVM)来获取^[8].根据所需特征的不同,可将图像检索分为基于内容和基于语义两类.其中:基于内容的图像检索仅利用图像或视频数据中的颜色和纹理等视觉特征来进行检索,具有代表性的系统包括加州大学开发的 Chabot^[3]和 IBM 公司开发的商业化系统 QBIC^[9];基于语义的图像检索则在此基础上进一步利用了图像表达的语义特征(如人们对图像内容的理解等),因此,该应用不仅需要图像或视频数据,还需要包含图像语义的文本信息(如研究人员根据理解手动为图像添加的注释等),为了支持这类应用,IBM 公司推出了 SVI^[10],哥伦比亚大学也提出了 SVT^[11].
- 其次,随着深度学习的大放异彩,视觉分析应用也经历了重大变革.深度神经网络以其卓越的准确度,使得这一阶段的视觉分析应用开始聚焦于如何使用深度学习来应对更多的场景,包括简单的对象识别或复杂的智能交通等^[12].简单的对象识别等应用仅需要图像或视频数据即可完成对人、车等目标物的检测与提取.斯坦福大学开发的 NoScope^[13]和密西根大学开发 Tahoma^[14]则主要面向这类应用.然而在复杂场景下,单一的图像或视频数据可能无法完整描述场景特征^[1],例如汽车传感器不仅可以生成图像或者视频,还包含与之匹配的时间戳、车速信息等.因此,复杂场景下的视觉分析应用通常需要利用这些多模态特征数据来满足应用需求.以智能交通应用为例,利用多位置摄像头视频和时间戳等信息,视觉分析技术可以实现车辆的自动驾驶和交通流量的优化.一些系统如斯坦福大学研发的 BlazeIt^[15]和佐治亚理工学院设计的 EVA^[6]逐渐开始涌现,以处理这类应用.

本文主要关注支持深度学习的视觉数据库管理系统.目前,这类系统的相关研究技术已日趋成熟,系统数量众多.现有综述^[16]主要从深度学习系统优化的角度将现有系统分类为计算和存储优化两个方向.其中,计算优化进一步划分为模型简化、数据量减少等优化方向,但这种分类方式无法涵盖数据库管理系统中与计算相关的查询优化等方面.此外,存储优化也细分为存储空间和数据解码优化,同样无法涵盖数据库管理系

统中与存储相关的索引等方面. 文献[17]则主要聚焦于单独介绍每个系统的应用场景和解决方案, 未深入归纳各个系统之间的联系与差异. 综上, 本文旨在总结视觉数据库管理系统的数据库查询和管理技术.

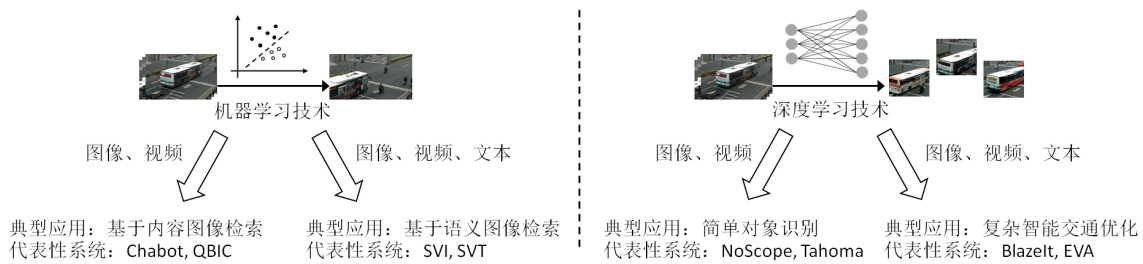


图2 视觉分析应用发展历程

本文第1节分析视觉数据库管理系统面临的挑战以及相关技术. 第2-5节关注现有视觉数据库管理系统在编程接口、查询优化、执行调度、数据存储这4个层面的相关技术. 第6节探讨视觉数据库管理系统未来的发展方向和趋势. 第7节总结全文.

1 视觉数据库管理系统简介

在本节中, 我们将从一个典型的视觉分析应用出发, 来介绍视觉数据库管理系统处理应用时的层次结构, 并据此分析视觉数据库管理系统所面临的挑战与相关技术分类.

图3展示了一个视觉分析技术在交通视频数据中的应用案例. 在该案例中, 用户希望通过文本信息中的时间戳属性以及视频数据来获取下午2点后的公交车车辆情况, 以进行交通流量优化等应用. 其中, 系统执行查询的具体流程可分为以下4个部分.

- (1) 系统为用户提供简易且高效的编程接口. 系统在接收用户查询后, 根据语法规则将其转换为语法树, 并在此基础上生成相应的查询计划.
- (2) 系统通过调整计划中算子的实现和执行顺序等方式来优化查询计划, 以减少查询执行时间.
- (3) 系统中的调度器根据当前数据分布、资源使用情况具体调度优化后的执行计划.
- (4) 系统在查询计划实施过程中, 从存储层读取数据, 包括原始数据、中间数据和一些必要的元数据.

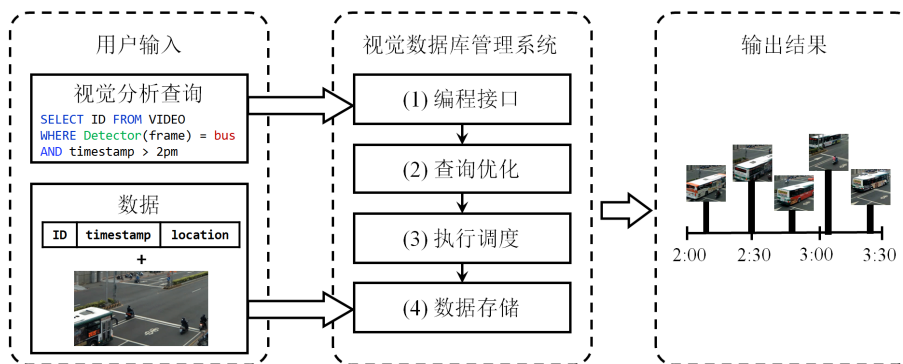


图3 视觉分析应用示例

因此, 视觉数据库管理系统涵盖了编程接口、查询优化、执行调度和数据存储这4个层面. 本文将总结这4个层面所面临的挑战及其对应的相关技术.

1.1 研究挑战

根据对视觉数据库管理系统的层次划分, 其面临的挑战可具体分为如下4个方面.

- 在编程接口层面, 如何扩展 SQL 语言来表达视觉分析语义, 从而降低编程难度、拓展优化空间. 传统的视觉分析算法通常使用命令式编程语言, 如 Python、C 语言等, 执行逻辑需要用户自行定义. 一方面, 尽管这类编程接口足够灵活, 但这也要求用户自行了解视觉数据库管理系统的相关原理(如任务执行流程、CNN 模型选择等), 加大了用户使用的门槛. 相反地, SQL 语言隐藏了视觉分析计算的细节, 提供了高度抽象的编程接口. 这意味着用户只需声明地给出数据之间的逻辑关系, 降低了用户编程难度. 另一方面, 编程接口越接近于通用编程语言, 系统可优化的机会越少. 通用编程语言的即时执行特性, 使得系统通常会按照用户定义的执行顺序来执行查询. 然而, 由于算法中的复杂逻辑, 用户提供的执行顺序通常是次优的. 此时, SQL 编程接口的惰性执行特性则便于系统收集查询信息, 并利用其优化器自动对查询执行进行优化. 但 SQL 语言受语法结构的限制, 仅适合编写简单的算法逻辑, 无法完整表达视觉分析应用中常见的准确度要求、计数统计等语义. 由此可见, 视觉数据库管理系统的编程接口需要扩展 SQL 语言来降低编程难度、拓展优化空间.
- 在查询优化层面, 如何面向视觉分析负载优化查询改写以及生成最优计划. 在传统数据库管理系统中, 查询优化是必不可少的一环. 然而, 相较于传统数据库管理系统, 复杂的视觉分析算法导致生成的执行计划中算子类型多, 算子实现方式与查询计划结构也更为复杂. 举例来说, 一个查询(如图像分类、交通检测等)可能包含多个预处理和推理算子. 一方面, 传统的计划等价变换规则如谓词下推等, 在视觉分析查询中作用有限; 另一方面, 查询中每个算子存在多个配置(如模型、帧率、分辨率等参数的组合)供选择, 不同配置对执行性能和表现可能有很大的影响. 一个次优的执行计划可能会大幅降低执行效率, 甚至会由于存储空间不足等问题导致系统崩溃. 因此, 视觉数据库管理系统需要探究查询优化以生成高效的执行计划.
- 在执行调度层面, 如何优化查询的并行执行以及查询的资源分配, 以提高资源利用率或降低资源消耗. 在深度学习领域, 通用深度学习系统都已发展成熟. 视觉数据库管理系统大多无须重新设计深度学习算法引擎, 而是利用已有的通用系统进行模型计算. 然而, 视觉分析查询存在多样的查询配置, 查询在不同配置下的计算与资源需求也会有所不同. 一方面, 在大数据应用中, 单机上的优化方法已经不能满足大规模视觉分析, 系统需要在分布式环境下优化并行执行; 另一方面, 面对多个不同配置的查询, 系统需要在可用资源上调度这些查询以提高整体性能. 因此, 系统需要结合查询特征来优化查询执行与资源分配.
- 在数据存储层面, 如何节省数据的存储空间以及提升数据读取效率. 除了计算过程外, 数据库管理系统还需要考虑应用特点对数据存储进行优化. 在视觉分析应用中, 非结构化的视觉数据拥有多样的组织方式. 更重要的是, 视觉数据在时间与空间上的局部性使得数据之间相互关联, 盲目存储所有数据可能会造成冗余存储. 此时, 需要系统能够自适应地选择合适的存储方式以节省存储空间. 在数据读取方面, 传统数据库管理系统中的采样、索引、缓存机制都没有充分考虑视觉分析应用与数据存储方式的多样性, 可能导致数据读取效率低下. 由此可见, 系统在数据存储时需要考虑数据特性来降低数据的存储空间并提高系统的读取效率.

1.2 相关技术分类

如图 4 所示, 本节将视觉数据库管理系统的相关技术总结归纳为以下 4 个方面.

- 编程接口. 基于通用编程语言的编程接口灵活性高, 但也加大了视觉分析应用的编程难度. 与数据库类似, 为降低编程难度, 视觉数据库管理系统提出了基于 SQL 语言的编程接口. 首先, 视觉数据库管理系统通过在原有 SQL 语言上引入自定义函数以扩展编程接口, 例如, 面向分类、追踪等不同任务设计相应的自定义函数; 其次, 为了限制分析结果的准确度与误差等指标, 系统扩展了 SQL 的语法结构来表达这些信息. 同时, 为了适应复杂的均值统计、不同对象检索等查询, 系统还对 SQL 进行了各种扩展, 例如: 在均值统计场景中添加聚合操作以及均值计算的语法, 以更加灵活地调整查询的表达方式, 提高编程接口的表达能力.

- 查询优化. 视觉数据库管理系统中的查询优化问题与数据库中传统查询优化问题类似, 可分为逻辑计划优化和物理计划优化两方面: 首先, 在逻辑计划方面, 系统需要对等价计划转换方法进行优化以提升执行效率, 其中涉及利用过滤型或代替型专用模型在计划中插入有益算子, 以及利用重叠的视觉数据、模型的公共前缀等特性对计划中的算子中间结果进行合并重用; 其次, 在物理计划方面, 数据库中典型的优化方向包括算子实现、代价估计以及计划选择, 对应地, 视觉数据库管理系统利用压缩和级联等模型优化技术实现了高效的视觉分析算子. 同时, 考虑到算子的批次大小、部署环境(如 CPU, GPU)等因素, 系统进一步设计了算子代价估计方案, 包括对数据获取、预处理和推理步骤的代价估计. 此外, 鉴于视觉分析查询的准确度和性能指标通常相互制约, 为了实现两者的权衡, 系统提出了粗/细粒度计划选择等优化策略, 以确定最佳计划.

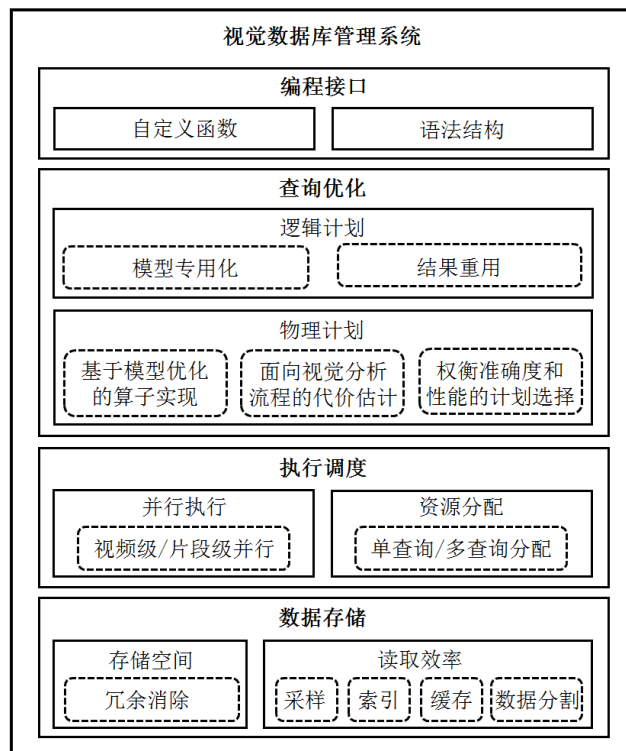


图4 视觉数据库管理系统相关技术分类

- 执行调度. 根据数据与查询规模、硬件环境的不同, 视觉数据库管理系统的执行调度优化也有所侧重. 在并行执行方面, 传统的视频级并行方式擅长处理各种查询却容易引起资源浪费; 相比之下, 片段级并行方式充分利用了查询对象可能仅在视频画面中短暂停留时间的特性, 能够有效地提高查询并行度、避免数据倾斜, 进而充分利用资源. 在资源分配方面, 面向单查询的系统着重于考虑数据与硬件环境等情况来优化算子的放置位置和执行顺序等; 而面向多查询的系统则善于根据查询不同的准确度或性能指标, 灵活地抽调与分配资源.
- 数据存储. 根据自身定位不同, 视觉数据库管理系统的数据存储可分为存储空间优化和读取效率优化. 具体来说: 在存储空间方面, 由于视觉数据可能存在重叠(如来自同一地点多个角度的视频), 系统通过合并或过滤这些重叠部分的冗余数据来减少所需存储的数据量, 从而降低存储空间; 在读取效率方面, 系统根据视觉数据的时间局部性(如短时间内的数据存在高度相似性, 均包含查询对象)进行采样处理, 从而有效地减少了需要读取的数据量. 此外, 系统利用高效的轻量级模型(如专用模型)的检测结果为查询对象建立索引, 以加速对象检索. 由于视觉数据通常压缩存储, 为避免在读取过程中产生过大的解码开销, 系统采用缓存技术来缓存不同格式的解码数据, 以避免反复解码. 同时, 系统还研究了数据细粒度分割技术, 使得系统只需解码包含查询对象的区域.

综上所述, 这4类相关技术分别从不同角度致力于解决视觉数据库管理系统所面临的问题. 接下来, 本文将详细介绍这4项技术分类的当前研究进展.

2 编程接口

现有视觉数据库管理系统的编程接口根据编程语言可分为基于通用编程语言以及基于 SQL 语言。在视觉数据库管理系统中,通过使用通用编程语言,用户可以使用丰富的第三方库和框架,如 OpenCV^[18]、TensorFlow^[19]、PyTorch^[20]等,来构建视觉分析应用的各种操作。同时,通用编程语言的灵活性也使得系统可对编程接口进行不同程度的抽象。其中, NoScope、Smol^[21]、FiGO^[22]的编程接口的抽象程度相对较低,接近于通用深度学习系统。相较于这些系统, Vista^[23]、Scanner^[24]提供了高度抽象的编程接口。特别地,这些系统能够向上层用户隐藏底层的并行执行逻辑。然而,通用编程语言的灵活性也导致其使用门槛较高。例如,相同的视觉分析算法可以通过不同的编程方式实现。对于缺乏足够编程经验的用户而言,很难进行快速地理解和应用。

此外,现有系统也提供了基于 SQL 语言的编程接口。利用 SQL 语言来编写视觉分析查询主要包括两方面的优点:一方面,SQL 语言大大降低了用户的使用门槛,这是因为 SQL 语言只需声明数据之间的转换关系就可以方便地供用户使用;另一方面,SQL 语言使得系统能够应用大量优化策略自动生成适当的执行计划,其中包括传统数据库中的各种标准查询优化(如跨查询的冗余消除操作等),也包括特定于视觉分析查询的逻辑、物理计划优化等。本文将在第 3 节中详细描述这些优化工作。然而,基于 SQL 语言的编程接口所面临的关键问题在于,如何赋予编程接口独立描述完整视觉分析应用的能力。这要求编程接口必须具备对分类、追踪等多样化视觉任务的支持,并且能够支持对准确度与误差等特定信息的指定。

2.1 自定义函数扩展

现有的视觉数据库管理系统首先通过扩展自定义函数的方式来实现编程接口,包括 Optasia^[25]、PP^[26]、CORE^[27]、Blazelt、LightDB^[28]、VIVA^[29]、EVA、MIRIS^[30]、Tahoma, 以支持多样化的视觉分析任务。自定义函数允许用户混合使用简单的 SQL 和通用编程语言来表达任务意图,而不必构建复杂的 SQL 查询,从而提高了编程接口的功能性和可读性。视觉数据库管理系统通常在系统中内置了较多的自定义函数,随后,用户在编写 SQL 时可直接调用这些函数。其中,每一个自定义函数通常包含独立的视觉分析任务,如按颜色和类型对车辆进行分类、在摄像头间重新识别车辆、跟踪车道变化、识别牌照等,用户可以根据需要自由选择 and 组合这些函数。同时,用户也可自行编写与使用自定义函数。除此之外,这些自定义函数也支持参数化,能够接收图像或视频等视觉数据、深度学习模型选择、阈值设置等输入参数,以便用户可以定制分析查询的具体要求。更重要的是,这些函数通常兼具良好的准确度和执行效率。举例来说,这些函数可以通过避免并行计算时的模型冗余存储来进一步提高查询并行度,从而加快查询处理^[31];这些函数还可以支持批式处理来有效地利用计算资源,减少处理过程中的模型加载等开销^[32]。

图 5 展示了基于通用编程语言和基于 SQL 语言的编程接口的交通监测示例应用。

<pre>input = loader(...) input = preprocess(input) bbox, label, score = ObjectDetector(input, ...) if(label == car): color_input = preprocess(bbox) color = VehicleColor(color_input, ...) model_input = preprocess(bbox) model = VehicleModel(model_input, ...) if (timestamp > 19 & color == 'red' & model == 'SUV'): license_input = preprocess(bbox) license = License(license_input, ...)</pre>	<pre>SELECT timestamp, LICENSE(bbox, frame) FROM VIDEO CROSS APPLY OBJECT_DETECTOR(frame) WHERE timestamp > 7pm AND label = 'car' AND VEHICLE_COLOR(bbox, frame) = 'red' AND VEHICLE_MODEL(bbox, frame) = 'SUV'</pre>
(a) 通用编程语言	(b) SQL 语言

图 5 交通监测应用

该应用的目的是,在交通视频中定位在晚 7 点后出现的红色 SUV 并识别其车牌。如图 5(a)所示,应用首先利用一个对象检测器检测视频帧中的汽车;然后,分别对检测到的汽车进行颜色和类型的识别;最后,符

符合条件的车辆将进行车牌识别, 并获取其时间戳信息. 如图 5(b)所示, 在利用 SQL 语言后, 一个复杂的应用可以在几行内声明, 这大大降低了用户的编程难度. 其中, 对象检测和车牌识别过程均被封装于自定义函数中(即 `ObjectDetector` 和 `License`). 同时, 时间戳、汽车颜色和类型等过滤条件则作为查询的谓词, 而颜色和类型识别同样以自定义函数的形式存在(即 `VehicleColor` 和 `VehicleModel`).

2.2 语法结构扩展.

除了自定义函数扩展外, `Blazelt`、`VIVA`、`Optasia` 等系统还通过扩展 SQL 关键词等语法结构, 以支持特定信息. 如图 6(a)所示, 为了表达准确度与误差等信息, `Blazelt` 首先引入了用于定义置信阈值(`AT CONFIDENCE`)和误差容忍(`ERROR WITHIN`)的关键词; 其次, `Blazelt` 还增加了面向限制(`limit`)查询的关键词(`LIMIT`), 例如, `LIMIT 10` 表示仅选择符合条件的前 10 辆 SUV. 此外, `Blazelt` 也提供了跨帧平均计数的查询语法, 使编程接口能够表达聚合(`aggregation`)查询, 例如计算每一帧中的平均汽车数量等. `VIVA` 则在 SQL 语法中添加了声明式接口, 用于表达模型间的关系等领域特定知识. 如图 6(b)所示, 在对象检测时, 低代价的 `ObjectDetectorFaster` 能够在昂贵的 `ObjectDetector` 之前进行数据过滤. 同时, 低代价但可能不够准确的 `LicenseFast` 则可以直接代替昂贵的 `License` 用于车牌识别. 这些领域特定知识为查询优化器提供了备选查询计划的模型信息. 查询优化器可以在这些计划中进行选择, 以提高查询性能或者在满足用户准确度要求的同时降低执行开销. 此外, 为了在分布式环境下更简便地利用自定义函数, `Optasia` 针对大数据处理系统进一步设计了分组、聚合和连接操作的语法. 通用数据分析系统, 如 `SQLFlow`^[33], 同样也选择直接在 SQL 上扩展语法以处理机器学习应用, 无须嵌入其他语言脚本. `SQLFlow` 也可以处理一些简单的视觉分析应用, 但由于其研究重点主要放在 SQL 编译方面, 其无法满足视觉分析多样的应用需求.

```

SELECT timestamp, License(bbox, frame)
FROM VIDEO CROSS APPLY
ObjectDetector(frame) WHERE ...
AT CONFIDENCE 95%
ERROR WITHIN 0.1
LIMIT 10
CREATE HINT ObjectDetectorFast CAN FILTER ObjectDetector
CREATE HINT LicenseFast CAN REPLACE License
SELECT timestamp, License(bbox, frame)
FROM VIDEO CROSS APPLY
ObjectDetector(frame) WHERE ...

```

(a) 指定准确度与误差等信息

(b) 指定模型间相互关系

图 6 语法结构扩展示例

2.3 小结与分析

表 1 从编程语言、语义完备性和易用性这 3 个方面展示了前述视觉数据库管理系统的编程接口, 其中, 各系统所使用的编程语言不再赘述.

表 1 编程接口总结

系统	编程语言	语义完备性	易用性
NoScope、Smol、FiGO	通用编程语言	***	*
Vista、Scanner	通用编程语言	***	**
PP、CORE、LightDB、MIRIS、Tahoma	SQL 语言	*	**
Optasia、Blazelt	SQL 语言	**	***
VIVA、EVA	SQL 语言	**	**

注: * 的数量体现了系统在某一方面的完善程度; * 的数量越多, 表示系统针对该方面的实现越完善

- 语义完备性

总体来说, 系统用户能够表达的任务描述越详细、指标等特定信息越丰富, 系统的语义完备性越高. 上述系统中, 采用基于通用编程语言的编程接口的系统具有最高的语义完备性, 包括 `NoScope`、`Smol`、`FiGO`、`Vista` 和 `Scanner`. 这是因为灵活的通用编程语言提供了较高的自主性, 允许用户实现各种复杂的任务逻辑和自定义处理. 语义完备性次之的一类系统包括 `Optasia`、`Blazelt`、`VIVA`、`EVA`, 它们提供了较为丰富的自定义函数, 使得用户能够组合这些函数来描述任务. 同时, 它们也扩展了语法结构供用户显式定义特定信息. 完备性最低

的一类系统(包括 PP、CORE、LightDB、MIRIS、Tahoma)则仅扩展了自定义函数,而未优化语法结构。

- 易用性

类似地,上述系统的易用性也可分为 3 个档次。通常来说,SQL 语言大大提高了编程接口的易用性。因此,结合丰富的自定义函数和语法结构优化,Optasia 与 Blazelt 的编程接口使得用户能够方便地编写多种多样的查询。相对于这类系统,PP、CORE、LightDB、MIRIS、Tahoma、VIVA、EVA、Vista 和 Scanner 则在易用性方面存在局限。具体来说,PP 等系统的编程接口通常仅适用特定的选择查询,但在处理跨帧平均计数等类型的查询时,它们的编程接口要求用户手动实现均值计算等底层逻辑。Vista、Scanner 的编程接口虽然基于通用编程语言,但得益于它们对编程接口进行了高度抽象,易用性也有所提升。易用性最低的一类系统包括 NoScope、Smolm FiGO,它们的编程接口基于通用编程语言且抽象程度较低,用户需要充分了解系统原理才能熟练地编写高效的查询代码。

3 查询优化

在传统数据库管理系统中,查询语句首先被解析成逻辑计划。相应地,在视觉数据库管理系统中,查询也会被解析成逻辑计划。该计划描述了算子的执行顺序,但不包含具体的物理实现。由于视觉分析算子通常包含代价昂贵深度学习模型计算,在计划中,不同位置执行这类算子所引起的开销有着显著不同。因此,如何做好逻辑计划优化,是提升查询性能的关键因素。此外,查询在经过逻辑计划优化后,需要确定算子的物理实现等细节,以生成物理计划。不同于传统数据库管理系统,视觉分析查询中的物理计划优化需要考虑查询准确度和性能之间的权衡。不同的物理计划具有明显的准确度和性能差异。因此,如何对物理计划进行优化同样至关重要。综上,本节将从逻辑计划优化和物理计划优化两个层面探究与视频分析相关的查询优化技术。

3.1 逻辑计划优化

逻辑计划优化的目标是,生成具有更好执行性能的等价查询计划。然而,在传统数据库管理系统中,诸如谓词下推、连接顺序优化等计划转换规则在视觉分析查询中的效果有限。这是因为视觉分析算子通常通过用户自定义函数在查询中实现,而传统的转换规则无法针对自定义函数进行优化^[26,27]。此外,由于其中昂贵的深度学习模型,这些算子通常计算代价较高。因此,视觉数据库管理系统的逻辑计划优化的关键在于减少算子处理的数据量或算子调用次数。

3.1.1 利用模型专用化技术生成等价计划

针对计划扩展的优化技术主要是模型专用化(model specialization)^[13,14]。模型专用化技术是一种针对特定场景进行模型优化的技术,其使用一个轻量级的专用模型来加速查询处理。通常情况下,视觉数据库管理系统需要处理各种不同类型的视觉数据,这些数据涵盖了不同的场景,系统需要针对特定的场景进行优化处理,以获得最佳的分析结果。然而,通用模型通常被训练来识别成千上万的物体,使用通用模型处理所有场景可能会导致不必要的计算开销,降低查询效率。例如,Mask R-CNN^[34](一种对象检测通用模型)能够从视频帧中提取对象类型和位置,其可用于回答查询如统计一帧中的公交车数量或找到同时有公交车和自行车的帧等。但是 Mask R-CNN 执行速度低至每秒仅能处理 3 帧图片。此时,系统通常利用模型专用化技术来提高系统的处理效率。专用模型无法识别通用模型能够检测的所有对象,但专注于检测查询的目标对象。举例来说,当查询用于统计视频帧中的公交车数量时,专用模型可用于初步回答该问题。系统只有在专用模型的检测结果不确定时(即结果的置信度较低)才会使用通用模型。通过减少包含重量级通用模型的算子所需处理的数据量,系统能够在可容忍精度下降的情况下加速查询。

具体地,专用模型可以分为过滤型专用模型和替代型专用模型。过滤型专用模型的输入/输出或检测类别通常与通用模型不同,但能够为通用模型过滤数据。例如:考虑一个图像分类模型,它为每张图像输出一个动物标签。再考虑一个对象检测通用模型,它可以产生相同的分类标签,但是却能够为图像中所有动物分类。图像分类模型可以被用于判断图像中是否有动物。系统只需将极有可能存在动物的图像传递给对象检测器。因此,图像分类模型起着过滤器的作用。如图 7 所示,过滤型专用模型过滤与查询对象不相关的数据,被认为

相关的数据随后使用通用模型进行处理; 相反地, 替代型专用模型具有和通用模型相同的输入/输出和相同或重叠的检测类别, 能够直接回答查询. 例如, 上文中统计公交车数量例子中使用的就是替代型专用模型. 虽然两个模型可以产生等效的输出, 但它们在性能和准确性方面存在差异. 如图8所示, 如果替代型专用模型不够精确, 系统才会使用通用模型.

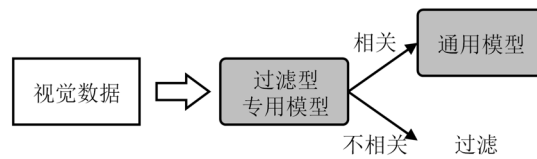


图7 过滤型专用模型工作流程

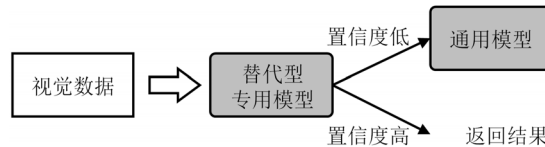


图8 替代型专用模型工作流程

系统在应用模型专用化技术后, 专用模型通常作为过滤算子添加至逻辑计划中. 图9展示了图3中应用示例查询的逻辑计划. 该计划中, 系统从数据集中读取数据并将它们传递至用于检测公交车的谓词, 通过谓词的数据会被作为结果返回. 应用专用模型的逻辑计划如图10所示, 系统在计划中插入专用模型来过滤不符合查询谓词的数据.

$$\text{VIDEO} \rightarrow \sigma_{bus} \rightarrow \sigma_{time} \rightarrow \text{Result}$$

图9 原始计划

$$\text{VIDEO} \rightarrow \sigma_{sp_bus} \rightarrow \sigma_{bus} \rightarrow \sigma_{time} \rightarrow \text{Result}$$

图10 专用模型优化后的等价计划

利用专用模型提升系统性能的技术已广泛应用于视觉数据库管理系统, 如 SVQ^[35]、PP、CORE、NoScope、SUPG^[36]、Blazeit、ABAE^[37]、ODIN^[38]等. SVQ、PP 与 CORE 应用了过滤型专用模型. SVQ 专注于处理与视频帧中对象的计数或空间约束相关的查询, 并相应地创建一系列确定查询对象计数和位置的专用模型. PP 未对查询类型进行限制, 其分别为查询中的每个谓词训练专用模型. PP 假设查询的各个谓词间相互独立并直接使用原始数据来训练专用模型. 然而在实际应用中, 查询中的谓词通常是相互关联的. 例如: 当查询中存在两个进行与操作的谓词且两者相关联时, 被第1个谓词丢弃的数据通常也会被第2个谓词丢弃. 此时, PP 基于独立性假设分别为两个谓词构建专用模型(两个专用模型分别标记为 $\hat{\sigma}_1$ 和 $\hat{\sigma}_2$)的做法会高估 $\hat{\sigma}_2$ 专用模型过滤的数据量. 这是由于当 $\hat{\sigma}_1$ 丢弃了不符合第1个谓词的数据时, 两个谓词的关联关系使得这些被丢弃的数据也可能不符合第2个谓词并被丢弃. 因此, PP 估计的 $\hat{\sigma}_2$ 的过滤数据量要大于其实际过滤的数据量, 从而产生次优的查询计划. 为了解决这一问题, CORE 利用谓词间的相关性分别为各个谓词的专用模型构建训练数据集, 从而优化了对专用模型过滤数据量的估计.

此外, NoScope、SUPG、Blazeit 等系统应用了替代型专用模型. 为了进一步减少通用模型处理的数据量, NoScope 不仅为给定的查询训练专用模型, 还实现了差异检测器来过滤近乎相同的视频帧. 这是由于当视频帧之间的时间间隔较小时, 视频帧内容相似度较高, 存在相似性. 例如: 在视频的同一秒内, 相邻几帧的内容几乎相同. 因此, NoScope 并不在每一帧上运行通用模型或专用模型, 而是训练一个低成本的差异检测器来确定视频内容是否在帧间发生了变化, 从而避免了在近乎相同的帧上进行冗余计算. 尽管 NoScope 利用专用模型减少了执行代价, 但是其使用的专用模型在面向罕见对象时, 会产生较多假阴性或假阳性的错误结果, 难

以保证查询的准确性. 为此, SUPG 着重针对专用模型的准确性进行优化. SUPG 首先利用重要性抽样(importance sampling)评估专用模型; 随后, 其通过对比专用模型结果与数据真实便签来估计一个专用模型置信阈值. 专用模型返回评分高于置信阈值的数据才会被认为是符合谓词的. 借助该置信阈值, SUPG 为查询提供了准确性保证. 然而, NoScope 和 SUPG 并不支持两类重要的查询, 即聚合和限制查询. 对于聚合查询, 常见做法是使用一种自适应抽样算法来保证用户的误差界限, 并根据样本运行在通用模型上的结果的方差提前终止采样. Blazeit 进一步利用专用模型来缩减样本间的方差, 从而减少样本数量. 然而, Blazeit 没有考虑聚合查询包含谓词的情况. ABAE 则在此基础上提出了面向多个谓词和按键分组的聚合查询的采样技术, 避免了 Blazeit 中对不满足谓词的记录进行抽样的缺陷. 限制查询常用于搜索罕见事件. Blazeit 可以对每一帧执行对象检测以搜索事件. 然而当事件不经常发生时, 这种连续扫描视频或随机采样的朴素方法通常性能低下. 为了解决该问题, Blazeit 使用专门模型进行有偏抽样. 通过采样优先级高的帧, Blazeit 在保证准确度的同时, 也因减少了样本数量而加快了查询执行. ODIN 利用专用模型来解决视频中漂移(drift)现象所带来的挑战. 通用模型在训练时通常假设所有的视频帧都来自静态分布, 因此, 当视频出现漂移现象时, 即视频的场景发生变化、视频数据内容与训练数据相差较大时, 通用模型将无法保证准确性. 例如一个无人驾驶汽车的通用模型, 如果训练自没有接受包含雪的图像, 在实际中遇到雪时检测准确度就较低. ODIN 通过一种无监督的算法来检测视频中的漂移现象. 当检测到漂移现象时, ODIN 调用漂移恢复算法来训练与部署针对新数据的专用模型. 这些专用模型在准确性和性能方面都将优于通用模型.

3.1.2 重用算子结果生成等价计划

计划重用优化问题类似于数据库中的多查询优化问题. 本文中, 算子重用优化旨在消除计划中可能存在的冗余计算. 针对算子中的模型计算, 除了数据库中消除多个计划间的公共算子的优化之外, 此处的算子重用还优化利用了视觉分析领域的 3 个特性.

- (1) 数据重叠: 同一地点不同角度拍摄的数据之间存在重叠的部分. 显然, 系统能够消除重叠部分的冗余计算来提高计算效率.
- (2) 模型公共前缀: 两个模型可能拥有公共的前缀部分. 这些公共前缀通常由一些卷积层和池化层组成, 用于提取输入数据的基本特征. 例如: 在图像分类任务中, 两个模型可能会共享一些较浅的卷积层, 因为这些层用于提取出图像中的基本特征(如边缘和纹理). 随后, 两个模型都有各自不同的后缀层, 以根据任务的要求学习更高级别的特征表示. 通过对算子中间结果的重用, 两个模型的公共前缀仅需计算一次, 从而避免了冗余计算.
- (3) 高准确度模型的结果覆盖低准确度模型的结果: 对于输入/输出和检测类别相同的两个模型, 高准确度模型的检测结果将会涵盖低准确度模型的检测结果. 因此, 将高准确度模型的检测结果物化并重复使用可避免低准确度模型的计算.

根据对算子结果重用的支持, 现有系统可分为两类.

- 第 1 类系统在单个查询内进行重用优化, 如 VisualWorldDB^[39]和 VIVA. 在需要确定对象的位置和方向时, 系统通常查询不同角度的视频. VisualWorldDB 则专注于重叠视频之间结果的重用. VIVA 针对具有多个谓词和模型的查询, 在查询中存在具有公共前缀的两个模型时, VIVA 会自动确定是否执行公共前缀部分的计算.
- 第 2 类系统则在多个查询间寻找重用优化的机会, 如 Optasia、Vista 和 EVA. 其中: Optasia 能够识别并消除多个查询计划间的公共算子; Vista 面向特征迁移负载进行优化, 其中多个查询涉及在同一模型的多个层上进行特征迁移, 这些查询存在着冗余计算, 因此, Vista 协调查询执行并在查询执行过程中重用模型的中间结果; 而 EVA 中的查询包含了目标相同但准确度要求不同的推理任务, 如两个查询分别需要利用高准确度和低准确度模型进行对象检测, 并利用检测结果进行后续处理, EVA 则自动确定如何在查询之间重用高准确度模型的检测结果, 以权衡结果的物化代价与重用优化的收益.

3.2 物理计划优化

确定逻辑计划后,系统需要进行物理计划层面的进一步优化。不同于传统数据库中的算子,视觉分析算子通常包含计算代价昂贵的深度学习模型,且这些算子往往是查询的性能瓶颈,因此,如何高效地实现这些算子是物理计划优化的关键问题之一。算子实现上的多样性使得同一个逻辑计划将会派生出多个物理计划,系统自然需要根据实际情况对这些物理计划进行选择。对物理计划做出选择依赖于算子代价的估计,然而传统数据库的代价估计仅面向选择、投影等通用关系算子,难以应对视觉分析算子,尤其是这些算子通常以自定义函数的方式实现。因而,对这些算子进行代价估计是物理计划优化的关键问题之二。特别地,视觉分析查询通常伴随着结果准确度等要求,而不同的物理计划之间的准确度存在着显著不同。一般情况下,物理计划的准确度越高,其执行性能越差。系统需要在两者之间进行权衡,以选择满足准确度要求且性能最好的计划。因此,如何进行计划选择是物理计划优化的关键问题之三。接下来,本文将从算子实现、代价估计、计划选择这3个方面介绍相关工作。

3.2.1 基于模型优化的算子实现

现有系统已提出了两种成熟的技术来降低算子内深度学习模型的内存需求和计算开销,即模型压缩和模型级联(model cascade)。

其中,模型压缩是一组旨在通过削减模型的参数数量从而减少深度学习模型的大小和计算复杂度的技术。这些技术通常会在一定程度上降低模型的准确度,但可以通过权衡准确度和计算效率来达到最佳性能。这些技术包括去除部分昂贵的卷积层^[40]、降低输入图像分辨率^[41]、参数压缩^[42]等。其中,去除部分昂贵的卷积层是一种典型的压缩技术。这是由于部分卷积层可能对模型的准确性贡献较小,但会大大增加模型的计算复杂度。因此,去除这些不必要的卷积层可以有效地减少模型的大小和计算复杂度。举例来说,ResNet18是ResNet152去除部分卷积层后的变体。相较于ResNet152,ResNet18获得了8倍以上的性能提升,同时保持了较高的准确性。输入图像的分辨率对于模型的计算量和内存需求也具有重要的影响。在图像分辨率较高的情况下,模型输入的特征图大小会相应增加,这会导致其中的卷积等操作需要处理更多的参数,从而增加模型的计算量和内存需求。因此,降低输入图像分辨率可以减少特征图的大小,降低模型需要处理的参数数量,从而减少模型的计算量和内存需求。最后一类重要的压缩技术则是参数压缩,其主要利用模型参数的冗余性进行模型压缩和加速。参数压缩方法十分多样,包括基于重要性准则判断参数的重要程度进而删除冗余参数的参数剪枝^[43]、将参数从32位全精度浮点数量化到更低位数的参数量化^[44]、将高维参数向量降维分解为稀疏低维向量的低秩分解^[45]以及利用结构化矩阵或聚类等方法映射网络内部参数的参数共享^[46]等。

除了模型压缩技术外,模型级联是另一种更为流行的优化技术^[47,48]。该技术从另一个维度对模型实现进行优化,即,通过级联多个分类或对象检测模型来提高模型的计算效率和准确度。图11展示了模型级联的一般过程:将图像输入分类或对象检测模型,如果模型结果具有高置信度,则接受其输出;否则,接着使用第2个模型进行检测。类似地,低置信度的结果将发送给第3个模型。依此类推,直到达到级联的最后一个模型并接受其输出。通常,级联中的第1个模型是快速且准确的,用于快速将大多数图像分类到正确的类别。例如:在人脸检测中,可以使用第1个模型来判断图像中是否存在人脸。如果大多数图像根本没有人脸,第1个模型能够会快速过滤大多数图像。级联中后续的模型则在第1个模型的基础上进一步检测,如确定人脸的表情、年龄或性别等信息,从而提高准确度。模型级联技术已广泛应用于视觉数据库管理系统。Tahoma通过组合多个不同专用模型来构建一系列模型级联,并根据查询的输入特征(如图像分辨率、存储位置等)选择合适的模型级联,以在满足用户准确度要求的前提下最大化系统吞吐量。面对压缩后的输入视频,CoVA^[49]通过级联压缩领域和未压缩的像素处理领域的技术,缓解了查询执行中的解码瓶颈。通常来说,现有模型只能识别有限的对象(如人、车等)且假设数据中不会出现新的对象。然而现实中的查询通常更为多样且细致,例如“这个人是谁?”,而我们不可能事先获得所有人脸的标签,因为新人会不断出现。为解决该问题,Panorama^[50]通过连接通用模型和一个嵌入提取模块,在不重新训练模型的情况下提升模型的泛化能力。

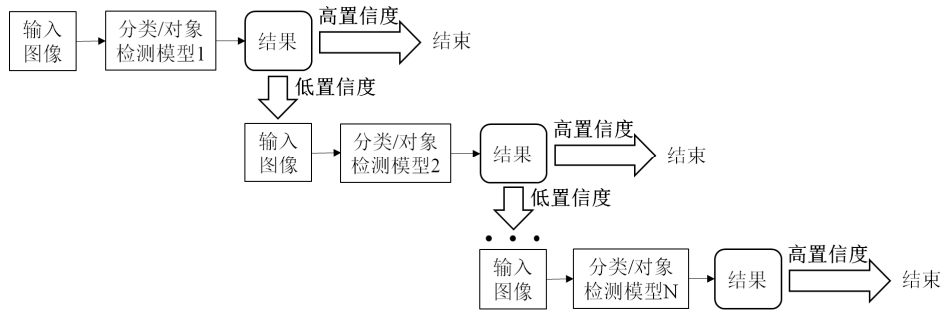


图 11 模型级联技术

3.2.2 面向视觉分析流程的代价估计

在视觉数据库管理系统中,如何估计包含深度学习模型的算子代价至关重要.由于这些算子可能包括数据获取、预处理和模型推理中的多个步骤^[14],因此,对算子的代价估计也就转变成了对这些步骤的代价估计.数据获取代价指视频或图像等视觉数据的加载时间,该代价与系统的部署场景紧密相关.预处理代价则包含图像解码、大小调整、裁剪、归一化等特征工程操作的时间以及将图像特征传输到 GPU 等加速器的时间.模型推理代价指在特定硬件上执行 CNN 对图像特征进行预测的时间.现有视觉数据库管理系统通常专注于对模型推理代价的估计.根据对批处理优化的支持,现有系统可分为两类.

- 第 1 类系统在模型执行过程中固定批次大小,如 NoScope、Vista、EVA、FiGO,它们仅需简单地模型在特定批次大小上的时间进行测量.
- 第 2 类系统则面向可变批次大小的模型执行过程.为了快速获取模型推理代价,Nexus^[51]、Clipper^[32]、VIVA 采用了静态测量或分位数回归来建模不同批次大小下的推理代价.然而,这些系统没有考虑预处理代价.Smol 则在此基础上估计了预处理代价,建模了解码、归一化等操作的代价.此外,Smol 还进一步考虑了预处理与模型推理步骤流水线执行的情况,并对该情况下的执行代价进行了估计.

更多地,上述系统均没有考虑系统的部署场景可能有很大的不同.例如:有些系统可能已将视频帧作为单独的图像文件存储在磁盘上,而有些系统可能需要在处理之前通过网络传输视频等.为避免该问题,Tahoma 进一步将数据获取时间纳入算子的代价估计中,从而保障了代价估计的准确性.

3.2.3 权衡准确度和性能的计划选择

在完成对算子的代价的估计后,系统需要进行计划选择来权衡查询准确度和性能.在选择计划时,系统会考虑不同的数据集、模型集合、硬件条件等因素,这些因素会影响不同计划的性能和准确度表现.系统需要综合考虑这些因素,选择适合当前环境和需求的计划,以达到准确度和性能的权衡.根据计划粒度的不同,计划选择的思路分为两种.

- 第 1 种思路是粗粒度计划选择,即,为整个数据集选择一个固定的计划.对应的系统有 PP、CORE、NoScope、Blazelt、Focus^[52]、Tahoma、Panorama、EVA、VIVA、Smol.它们通常依赖于各个计划在保留数据(held-out data)或视频中的采样数据上的评估结果,并根据该结果为整个视频选择一个计划.然而在给定准确度约束下,粗粒度计划选择可能会导致更高的查询处理时间.这是由于:一方面,查询的对象往往不会出现在视频的每个片段中,特别是在对象出现频率较低的视频中;另一方面,查询对象易于检测的视频片段也将使用为那些查询对象难以检测的视频片段定制的高准确度计划.即使该计划满足了准确度要求,但就执行时间而言,该计划也可能不是最优的.
- 为此,第 2 种思路使用细粒度计划选择,对应的系统有 FiGO、MIRIS、Chameleon^[53].如图 12 所示,细粒度计划选择将为数据集中的不同片段选择合适的计划.然而,无论是粗粒度还是细粒度计划选择,都需要在部分数据上评估候选计划.如果系统分析所有可能的计划,那么计划选择开销可能会超过减少的查询执行时间,从而导致负收益.因此,FiGO、CORE、VIVA 提出了计划裁剪技术.该技术的核心思路是,只评估可能比迄今为止发现的最佳计划的成本更低的计划.例如:如果一个低准确度计

划已经能够满足准确度要求, 而当前计划使用更高准确度的模型, 那么系统将会丢弃该计划. 裁剪技术使得系统只需分析具有更高效用的计划子集, 从而降低查询优化时间.

为了进一步提升系统的性能, DeepCache^[54]、Krypton^[55]和 Deluceva^[56]采用了增量计算来优化执行计划. 增量计算已被广泛应用于大多数数据库与并行计算系统中, 其核心思想是: 如果仅有部分数据被更新(即增量), 那么系统可以仅对该增量部分执行相关操作, 从而避免了对未更新数据的重复处理. 类似地, 增量计算也可应用于视频分析应用. 这是由于视频数据的时间局部性, 即相邻视频帧上的大多数像素几乎相同, 在完整视频帧上执行推理将会造成大量的计算浪费. 如图 13 所示, 为了应用增量计算加速技术, 系统需要物化参考帧在模型计算过程的所有张量. 对于后续帧的推理, 增量计算不对完整帧重新运行推理, 而是尽可能地重用物化张量, 只对更新区域执行计算, 从而避免计算冗余. 然而, 增量计算在模型计算过程中存在增量消减效应^[54,55]. 具体来说, 随着模型的执行, 增量计算在模型每一层的特征图上可重用的张量会不断减少. 因此, Krypton 利用近似思想扩展了增量计算, 即近似计算. 如图 13 所示, 我们已卷积层为例解释近似计算. 近似计算的主要思想是: 裁剪更新区域的计算输出, 使其不超过完整计算中输出大小的给定阈值. 值得说明的是, 近似计算只会裁剪处在“边缘”位置的一些元素, 因为它们对后续层的计算结果贡献较小. 在图 14 中, 由于我们设置了阈值为 3/5, 而此时更新区域的输出大小是完整计算中输出大小的 4/5, 故其需要被裁剪. 为了裁剪后的输出, 近似计算也需要相应地缩小卷积层的输入区域, 并因此增大了可重用的区域, 从而避免了增量消减效应.

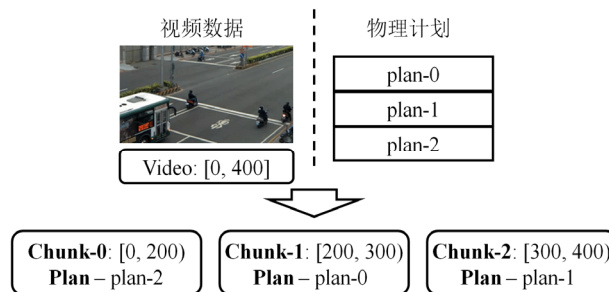


图 12 细粒度计划选择

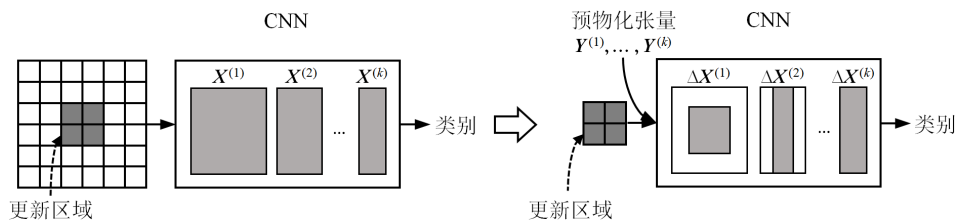


图 13 增量推理

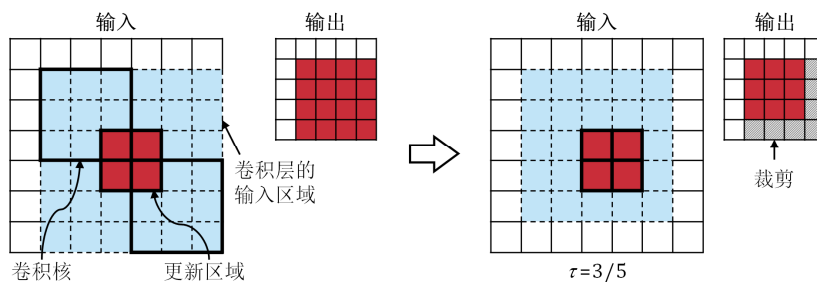


图 14 近似推理

3.3 小结与分析

表 2 列举了前述视觉数据库管理系统在查询优化层面的共性和差异, 主要分为逻辑计划或物理计划优化以及查询类型通用性两个方面.

- 逻辑计划或物理计划优化. 上述视觉数据库管理系统实现了对算子与计划的多种优化方式. 其中, VIVA、EVA 的查询优化程度较高, 这些系统从逻辑计划与物理计划两个层面进行了综合考虑. SVQ、SUPG、NoScope、ABAE、ODIN、Vista、PP、CORE、Blazeit、Optasia 单独对逻辑计划进行了优化. 其余的系统(包括 Tahoma、Panorama、FiGO、DeepCache、Krypton、Deluceva)则专门优化了物理计划. 由于这些系统仅面向单一层面进行优化, 其查询优化技术相应地存在局限性. 以 NoScope 这一典型系统为例, 为了充分发挥专用模型的性能优势, 系统仍需在物理计划层面根据数据和查询特性来挑选合适的专用模型.
- 查询类型通用性. 总体来说, 上述系统中, Blazeit 采用的查询优化技术较为全面, 其能够覆盖多种多样的查询类型(包括选择(selection)查询、聚合(aggregation)查询以及限制(limit)查询), 而并不仅仅针对某种特定查询. PP、CORE、Optasia、CoVA 所采用的优化技术可优化选择和聚合查询, 但无法优化限制查询. 剩余的系统则专门针对某一类查询进行查询优化, 查询通用性较低. 虽然这些系统的适用范围有限, 但在特定查询中往往可以取得较好的查询优化效果. 其中, SVQ、SUPG、NoScope、ODIN、Vista、Tahoma、Panorama、FiGO、DeepCache、Krypton、Deluceva 提出的优化技术仅针对选择查询. 类似地, ABAE 局限于优化带有谓词的聚合查询的执行效率.

表 2 查询优化总结

系统	逻辑计划优化	物理计划优化	查询类型通用性
SVQ、SUPG、NoScope、ABAE、ODIN、Vista	√	—	※
PP、CORE、Optasia	√	—	※※
Blazeit	√	—	※※※
Tahoma、Panorama、FiGO	—	√	※
CoVA	—	√	※※
VIVA、EVA	√	√	※
DeepCache、Krypton、Deluceva	—	√	※

注: ※的数量体现了系统在某一方面的完善程度; ※的数量越多, 表示系统针对该方面的实现越完善

4 执行调度

在得到经过优化的物理计划后, 为了高效执行该计划, 系统需要确定执行方式以及分配所需资源. 分布式并行执行是数据库和大数据处理系统中处理大规模数据的常见方式. 鉴于视觉数据通常也呈现出大规模的特点, 视觉数据库管理系统同样可以采取并行执行方式. 然而在并行执行过程中, 系统仍需考虑如何在各个并行实例间划分数据, 以避免数据倾斜. 在确定具体的执行方式后, 系统需要为计划分配适当的资源以确保其运行. 然而, 系统在资源分配时仍面临着一些挑战. 这是由于系统不仅需要应对不同的数据规模和硬件环境, 还必须考虑到视觉分析应用中参数的多样性(如分辨率、模型种类等), 不同的参数组合所带来的计算与资源需求也有所不同. 因此, 本节将从并行执行与资源分配两个方面综述系统在执行调度层面的优化.

4.1 并行执行

单机视觉数据库管理系统的性能面临着瓶颈, 越来越多的系统支持分布式计算, 以解决临界资源带来的瓶颈问题. 通过水平扩展, 系统可以将视觉分析任务下发到多个物理节点上, 利用物理并行方式为视觉分析提供大规模的并行数据处理功能. 尽管通用深度学习系统支持分布式计算与多种复杂的模型和硬件加速器, 但在大规模数据下, 仍需要手动划分数据和内存. 此外, 部分视觉数据可能以结构化或半结构化的形式表达, 存储在数据库或数据仓库中^[57], 而通用深度学习系统通常不支持对结构化数据的查询或自定义转换. 因此, 单独使用通用深度学习系统作为视觉数据库管理系统的底层执行引擎存在较大的局限性. 为了解决该问题,

视觉数据库管理系统中常见的做法是,将通用深度学习系统和大数据处理系统结合使用(包括 Optiasia、Vista、VIVA)。大数据处理系统(如 Scope^[58]、MapReduce^[59]、Spark^[60]、Flink^[61]等)面向通用数据分析负载,能够灵活地支持对结构化数据操作。更重要的是,这类系统支持自动化的分布式数据和内存管理。此外,这类系统的高可扩展性也使得视觉数据库管理系统能够轻松扩展至大规模集群或云平台^[62]。通过两者结合使用,视觉数据库管理系统能够充分发挥各自的优势,提供高效的分布式计算能力和对数据的灵活处理,从而克服了单一系统的局限性。

借助分布式计算,视觉数据库管理系统可以通过视频级并行执行优化和片段级并行^[25]执行优化两个方面来提高执行效率。视频级并行执行是最常见的优化方式^[24,29,63],其将一个完整的视频作为任务单元。系统将不同的视频分配给查询的不同并行实例进行处理,每个并行实例负责处理一个视频的所有视频帧,从而实现并行计算。这种并行执行方式的主要特点是任务单元具有完整性,能够保证处理结果的一致性和准确性,因此其适合于所有查询类型。然而,视频级并行可能会导致倾斜现象^[25]。这是因为不同视频中的查询对象数量可能相差较大,查询对象较多的视频对应的并行实例将会产生更多的计算量。与此同时,视频级并行的查询并行度取决于视频的数量。这也可能导致系统无法充分利用集群资源来提升性能。为解决这些问题,该领域也出现了以视频片段为单位的并行执行方式。片段级并行执行将视频分割成多个片段,并分配到不同的并行实例上进行处理。片段级并行执行通过调整片段的大小来协调并行实例的计算量,从而缓解倾斜现象。此外,这种方式还可以根据资源的可用性调整片段的数量,从而优化查询的并行度,避免资源浪费。具体来说,片段级并行执行可以分为两个阶段:分段阶段和分发阶段。分段将整个视频分成多个连续的片段。通常,分段是基于时间戳进行的,每个片段都对应于一个连续的时间段。在分发阶段,系统将各个片段分发到不同的并行实例中,每个并行实例都独立地进行处理。当所并行实例完成处理后,系统再将处理结果合并成一个完整的视频结果。值得注意的是:片段级并行执行在面向聚合查询时,如从公路视频中统计车辆总数,可能会导致错误的结果^[25]。这是由于聚合查询需要跨帧的状态信息以避免重复计数。为此,Optiasia提出了分段优化技术,即在分段时将视频分成包含部分重叠帧的片段,以避免片段间的依赖性。Optiasia发现跨帧的状态保持在一个有限的时间范围内,例如每辆车只在视频的画面中停留有限的一段时间。因此,系统可以将视频分成多个部分重叠的片段。如果车辆在 t 帧内通过视频画面且每个视频片段的长度为 s 帧,那么第 n 个片段则包含视频的第 $ns-t$ 帧-第 $ns+s$ 帧。系统在处理第 n 个片段时,先使用前 t 帧来预热其状态,用于检测与第 $n-1$ 个片段内重叠的车辆信息。随后,系统利用该状态处理其余 s 帧,从而避免了片段级并行在聚合查询中的错误结果。

4.2 资源分配

视觉分析中的资源分配问题是指如何将分析过程映射到硬件资源或分布式节点,其与查询的配置紧密相关。视觉分析查询包含各种参数,包括帧率、帧分辨率、批次大小和模型种类(如 Mask R-CNN、ResNet 或 VGG)等。本文将这些参数的组合称为查询配置。随着参数的数量以及参数的可选值数量的增加,查询配置的数量呈指数增长。查询配置既会影响资源需求,也会影响查询的准确性。例如:使用高分辨率(如 1080p)或具有多层的负载模型可以精确地检测对象,但也需要更多的 CPU 或 GPU 资源。基于查询配置,系统可以从两个方面进一步优化视觉分析的资源分配:首先,面对单个查询,系统需要根据查询配置将查询中的各个算子(包括多个预处理与模型推理算子),调度到可用的硬件资源,以便高效利用资源;其次,面对多个相互竞争的查询,系统需要为各个查询分配合适的资源。多查询情况通常会引入额外的挑战,包括更大的调度算法复杂性以及额外的要求或约束(如考虑查询间公平性等)。下面我们分别从单查询任务调度和多查询任务调度两个方面综述这类技术。

4.2.1 单查询资源分配

在单查询资源分配中,我们通常有一组固定的或动态的资源(计算节点、CPU、GPU),可用于运行一组源自单个查询的算子。因而,单查询资源分配通常面临以下 3 个问题:(1) 确定查询中的哪个算子位于哪个资源上(如计算节点、CPU、GPU);(2) 确定各个算子何时执行或以何种方式执行,这包括并发执行的算子数量、算子的调度顺序、并行度等;(3) 确定何时以及如何调整查询或算子的资源分配。在执行查询的过程中,可能

会出现资源利用不均衡或性能瓶颈的情况。为了应对这些问题,系统需要动态地调整资源分配。解决这些问题需要综合考虑算子的计算需求、算子之间的依赖关系以及硬件资源的可用性和计算特性等。

Smol 和 Scanner 在固定资源上进行资源分配。Smol 在单个计算节点内根据查询配置与硬件情况决定将预处理算子调度至 CPU 或 GPU 上,以平衡模型推理和预处理的吞吐量。具体来说,Smol 预估预处理和模型推理的吞吐量:若预处理吞吐量较高,则将其中的操作(解码、大小调整等)尽可能地调度至 CPU 运行;反之,则尽可能地调度至 GPU 运行。随后,Smol 将预处理与模型推理以流水线方式高效执行。Scanner 则将查询的并行实例分配至异构集群的计算节点以及节点内的并行处理资源上。Scanner 通过查询的资源需求来确定每个节点可以创建的最大实例数量,然后在不过度分配节点的情况下最大化并行度,从而充分利用硬件资源。然而,这类面向固定资源的分配策略可能会限制查询的性能提升^[64]或导致过度分配和更高的成本。

Chameleon、Sprocket^[65]、Llama^[66]提出了动态资源分配策略。Chameleon 发现,满足准确度阈值的配置在资源需求上通常会有许多数量级的差异。在 Chameleon 中,具有最低资源需求且其查询准确度超过所需阈值的配置被称为最佳配置。Chameleon 中的动态调度策略考虑到查询的最佳配置将随着时间的推移而变化,因而其在线周期性地选择最佳配置、调整资源分配,以在达到准确度阈值的同时最小化资源使用。然而,频繁地搜索大量配置将引入大量资源开销,从而掩盖动态调度带来的收益。Chameleon 进一步利用影响最佳配置的对象特征(如对象的速度和大小)具有足够的时间和空间相关性,来随着时间和在多个视频间平摊搜索开销。但是,Chameleon 仍专注于在查询层面进行动态资源分配而非算子层面。Sprocket 和 Llama 则细粒度地利用无服务器(serverless)平台为每个算子动态分配资源。此外,为了避免在无服务器平台手动选择硬件类型或查询配置以满足延迟目标,Llama 还进一步在执行过程中自动为每个算子确定配置。

4.2.2 多查询资源分配

在多查询资源分配中,多个分析查询共享一组公共资源。因此,系统需要负责将这些查询按需调度至不同的资源。当下流行的通用资源调度器包括 Mesos^[67]、YARN^[68]以及 Borg^[69],这类系统面向通用数据分析负载,但并不是为视觉分析负载的特定属性而量身定制的。例如:除了前述的大量查询配置外,不同的查询可能有不同的准确度或延迟要求。有些查询需要实时生成结果,而其他查询甚至可能容忍几分钟的延迟。这意味着在资源暂时短缺的情况下,可以临时从那些容忍延迟的查询中抽调部分资源来满足实时查询的需求。考虑到这些特定的属性,系统需要制定特定于视觉分析的资源调度策略。

VideoStorm^[63]是一个专注于处理大量面向实时视频流的查询的分布式系统,其通过为每个查询生成其在不同查询配置下的资源需求以及准确度信息来权衡查询的准确度和延迟。具体来说,当存在相互竞争的两个查询时,VideoStorm 将分配更多的资源给准确度将得到更多提高的查询。同时,对于处于处理滞后的查询,VideoStorm 降低其准确度来提高其处理速度,使其不超过允许的延迟范围。这种资源分配策略可以有效地平衡查询性能和延迟要求,提供高效的实时视频流处理。Nexus 是一个面向 GPU 集群的资源调度器,其专注于对查询的模型推理算子进行调度,并能够在延迟约束下提高系统的吞吐量。对于输入数据较多的查询,Nexus 将为其分配多个 GPU 节点进行并行处理以降低处理延迟。为了避免资源浪费并提高吞吐量,Nexus 会将多个查询调度至同一 GPU。Nexus 会计算每个查询的最大批次大小和对应的执行时间,并根据每个查询的延迟约束调整它们在 GPU 上的执行顺序。这样可以最大程度地满足查询的性能要求,并提高整个系统的效率和处理能力。

4.3 小结与分析

表 3 从针对并行执行或资源分配优化、保证性能指标以及支持通用操作这 3 个方面分析了前述视觉数据库管理系统的执行调度,展示了这些系统之间的共性和差异。

- 针对并行执行或资源分配的优化。根据应用场景的不同,视觉数据库管理系统侧重的执行调度优化也有所不同。总体来说,Scanner、Sprocket、Llama、VideoStorm、Nexus 综合考虑了并行执行与资源分配两方面的优化,其优化技术具有较高的扩展性。相比之下,Optiasia 专门针对查询并行执行进行优化,而 Smol 和 Chameleon 仅优化了查询资源分配。这些系统针对执行调度的优化局限在某一方面。

当然,在不同的硬件环境上,系统也体现了不同侧重的资源分配优化. 其中: Smol、Chameleon 着重于优化单机内的 CPU、GPU 资源分配; VideoStorm、Nexus 着重于同构集群的资源分配,以提高整个集群的性能和效率; Scanner 则面向异构集群,着重于考虑如何发挥不同类型计算节点的特定优势和能力; 而 Sprocket、Llama 考虑云计算场景,利用云服务来实现算子层面的动态资源分配.

- 保证性能指标. 能够支持实时分析和满足性能目标是系统的一项重要能力. 用户通常要求系统在处理查询时需要满足延迟目标,更进一步地,同时最小化资源使用或提高查询准确度. 上述系统中, Chameleon、Llama 通常能够保证延迟约束并降低资源代价; VideoStorm 在满足延迟约束的同时提高了查询的准确度; 而 Nexus 则通过提高 GPU 的利用率,在不违反延迟约束的前提下提升了系统的整体吞吐量. 另一类系统包括 Optiasia、Smol、Scanner、Sprocket, 则面向事后分析. 这类系统通常不支持设置性能目标,而专注于充分利用可用资源,以在整体上提升系统性能.
- 支持通用操作. 为了构建完整的视觉分析应用,用户在查询中需要将多个分析操作结合起来,包括视频解码、图像过滤以及模型推理等. 例如,对车辆进行计数的交通应用通常包括解码操作、重新调整帧大小和采样操作以及紧接着的对象检测操作. 在上述系统中, Optiasia、Smol、Scanner、Chameleon、Llama 和 VideoStorm 支持对通用的分析操作分配资源; 而 Sprocket 和 Nexus 仅关注模型推理操作,依赖外部服务进行其他操作.

表 3 执行调度总结

系统	针对并行执行的优化	针对资源分配的优化	保证性能指标	支持通用操作
Optiasia	√	—	—	√
Smol	—	√	—	√
Scanner	√	√	—	√
Chameleon	—	√	√	√
Sprocket	√	√	—	—
Llama	√	√	√	√
VideoStorm	√	√	√	√
Nexus	√	√	√	—

5 数据存储

视觉数据通常具有大规模性,尤其是在视频监控、医疗成像和遥感等领域. 对于大规模数据,存储空间优化成为视觉数据库管理系统设计中不可或缺的一环. 与此同时,图像和视频等视觉数据通常具有较高的维度,高维度的数据包含了大量的像素和特征信息,这也导致数据的存储需求剧增. 此外,这些数据还常常需要在实时或尽可能短的时间内进行读取,以确保系统的性能和响应速度. 因此,如何高效地读取数据也是视觉数据库管理系统亟须解决的挑战之一. 接下来,本文将从存储空间优化和读取效率优化两方面进行介绍.

5.1 存储空间优化

针对存储空间优化,视频分析系统采取了多种策略,包括图像和视频数据压缩、冗余消除等. 其中,冗余消除是目前应用广泛的典型策略. 冗余消除主要用于过滤相似的视频帧或消除视频帧间重叠的区域.

VSS^[70]和 VisualWorldDB 观察到:同一地点多个角度的视频间会存在重叠的区域,而这些重叠的区域会导致冗余存储. 为了缓解这一问题,这两个系统首先利用特征检测算法或对摄像机位置的先验知识来识别视频帧中的重叠区域; 随后,系统可以选择将重叠区域进行合并,只保留一个包含完整信息的帧. 系统也可以对帧中的像素行进行细粒度存储. 通过消除重叠区域的冗余存储,系统减少了所需存储的数据量,从而降低了存储成本. 为了减少 VR(虚拟现实)视频数据所需存储空间, LightDB 首先将 VR 数据投影为传统的二维视频数据. 类似地, LightDB 消除每帧内的冗余信息来压缩存储空间,它能够识别具有高相似性的区域,并仅存储每个区域一次.

5.2 读取效率优化

视觉分析应用通常需要从数据存储中快速检索和读取大规模的视觉数据。为了实现低延迟和高吞吐量的目标,系统需要对数据存储和检索的方法进行优化。这种优化涵盖了多个方面。

- 首先,降低采样率来减少读取的数据量是一种合理的策略。然而,随机采样等做法可能无法提供足够的样本来支持查询,尤其是在查询对象较为罕见的情况下^[14]。
- 其次,使用索引结构来加速数据检索是传统数据库中常用的技术。视觉数据的索引构建通常依赖于高代价深度学习模型,因此,如何快速、高效地构建索引至关重要。
- 此外,高维度视觉数据通常压缩存储,如何降低读取过程中的解码开销也同样关键。

下面我们将分别介绍这 3 个方面的优化技术,包括采样优化、索引优化以及缓存和细粒度读取优化。

5.2.1 采样优化

采样优化旨在减少昂贵的检测模型在大型视频中进行对象搜索查询时需要处理的帧数。尽管模型专用化技术能够高效地过滤视频帧,但其仍然受到一些限制。

- 1) 昂贵的训练开销。专用模型通常需要特定于查询进行训练,并且需要来自人为或通用模型的大量注释。一方面,从人为或通用模型中获取大量训练数据来训练专用模型通常代价昂贵。例如:NoScope 和 BlazeIt 需要数小时的 GPU 计算来执行通用模型,以产生标签来训练专用模型;PP 等其他系统则需要昂贵的人工注释。另一方面,这些系统需要为每个查询提供适当的训练过程,并且特定于查询的专用模型难以在不同的查询类型之间共享计算。
- 2) 不适用于高选择率查询,即满足查询的对象在视频中经常出现。模型专用化技术所获得的加速依赖于数据减少量。假设我们有一个 N 帧的视频与一个过滤的帧数百分比为 r 的专用模型。系统在 1 帧上执行专用模型和通用模型的成本分别是 C_f 和 C_o 。为了加速查询, r 必须满足以下约束:

$$N \times (C_f + (1-r) \times C_o) < N C_o, r > C_f / C_o.$$

然而,具有高选择率的查询通常不满足该约束。

- 3) 不适用于难以检测或罕见的对象。一方面,专用模型通常设计得层次较浅以便更快地执行,因此,对于难以检测的对象,如帧背景中的小物体,专用模型无法提供较高的准确性;另一方面,专用模型依赖于视频的一个子集进行训练,所选子集中缺乏正例会极大地影响模型的质量。鉴于这些限制,与其通过专用模型来加速查询执行,部分现有工作选择在查询执行期间视频的采样率上优化执行速度。同时,理想的采样率也依赖于数据和查询特点。

ExSample^[71]面向不同对象的限制查询,如在数据集中找到 20 个红绿灯,这些查询通常涉及由多个摄像机产生的大量视频。不同对象查询要求每个结果都应该是不同的对象,对同一对象的多次检测只会产生一个结果。执行此类查询,一个简单优化方法是减少采样帧的数量。举例来说,我们可以合理地假设所有的交通灯在 1 s 以上都是可见的,因此我们可以只对视频中每一秒的一帧图像进行检测。采样做法相比于检测每一帧的做法而言大大提高了计算效率,然而这种做法在不同对象查询中也存在不足。例如:一个红绿灯可能在视频存在超过 30 s,那么对该对象的重复检测会造成冗余计算。鉴于此,ExSample 将采样问题建模为一个基于过去检测内容来决定从视频中何处提取下一帧来检测的问题。ExSample 首先将视频数据集分割为时间块,随后,其优先采样找到新对象的概率较高的块。对于每一块中找到新对象的概率,ExSample 以处理多臂老虎机问题(每臂对应视频的一块)的方式,利用汤普森采样算法进行估计。ExSample 将会迭代地选择具有最佳估计的块,从该块中随机抽取一个帧,并通过检测模型处理该帧。

MIRIS 以及 Vaas^[72]则用于处理对象跟踪查询,例如搜索一辆汽车近距离超过了一个骑自行车的人。专用模型在这类查询中还会存在额外的缺陷。这是因为专用模型通常用于优化以单帧为处理粒度的查询,而对象跟踪查询则涉及一个物体在几个视频帧(即视频片段)中的运动轨迹。虽然我们可以训练一个输入视频片段的专用模型,但这种方法产生的专用模型精度与执行速度都较低。因此,这两个系统也选择优化查询执行期间视频的采样帧率。它们通过在处理视频时尽可能地采用较低的帧率,但在需要时提高帧率,以确保查询的高

精度,从而提高了查询的执行速度.

5.2.2 索引优化

索引优化则与数据库中的索引相似,在视觉分析领域,索引优化的目的是加速对包含特定对象的数据的检索,因而其主要挑战在于如何根据查询对快速地对数据进行分类以创建索引.

为了对数据进行分类,TASTI^[73]发现,许多数据的推理结果在语义上是相似的.例如:查询可能计算汽车的数量或选择左下角的汽车,此时在左下角有一辆汽车的不同帧对于这类查询会有相同的结果.利用该发现,TASTI首先利用专用模型为一组参考的视频帧建立索引.在查询处理过程中,TASTI仅在这一组参考帧上使用通用模型进行标签预测.对于未检测的帧,TASTI使用专用模型来计算其与参考帧之间的距离,并通过聚类方法将其归类到相应的集群中.集群中,参考帧的标签将被视为该帧的标签.TASTI的关键假设是,专用模型的性能比重量级通用模型快一个数量级.然而,最近提出的通用模型(如Yolov5s^[74])的性能与专用模型相当,甚至可能更低.鉴于此,Seiden^[75]直接在视频帧的子集上应用通用模型来构建与查询无关的索引.对于未被采样到的帧,Seiden利用视频的时间连续性原则来为其分配标签.Seiden将根据已采样帧的标签,推断未被采样帧的标签.

Focus则发现,视频帧在通用模型上的top检测结果通常涵盖该帧在通用模型上的top检测结果,可利用该top-K结果来为每一帧建立索引.Focus认为,历史视频数据将会被多次查询,故将其查询处理分为数据摄入和查询两个阶段,数据摄入后可被多次查询.为了降低查询阶段的延迟,一种简单方法是在摄取阶段完成大部分工作.例如:在摄取阶段使用通用模型对每一帧执行对象检测,那么查询阶段对特定类(如汽车)的搜索将只需要一个简单的索引查找.然而,摄取阶段在这种做法中的代价较高.尽管采样或模型专用化技术能够过滤数据来减少摄入代价,但这些技术通常是特定于查询的.当我们需要在相同数据上查询多个不同对象时,为许多对象进行采样或运行多个专用模型的代价可能非常高.为了降低摄入代价,Focus在数据摄入时使用拥有更少卷积层的廉价模型.但廉价模型在准确度上不如通用模型.为了提高准确度,Focus在摄取阶段利用廉价模型的top-K结果来为每一帧建立索引,并在查询阶段利用该索引来过滤数据.但Focus在利用廉价模型降低摄取阶段代价的同时,却增大了后续查询阶段的延迟.因此,Focus进一步提供了参数(包括廉价模型和k值)自动优化策略,以满足用户指定的延迟和准确度目标.

5.2.3 缓存与细粒度分割优化

缓存优化旨在减少数据的解码开销,进而加速数据读取.通常情况下,系统往往采用一种默认组织结构对数据进行存储.然而,相同的数据通常被用于构建不同的视频分析查询,并且这些查询对数据的组织结构(如分辨率、编码方式等)要求有所不同.因此,系统需要反复对数据进行繁琐的解码操作.为了减少这一转换开销,系统可以多种组织结构对数据进行缓存.VSS和VStore^[76]都采用了缓存优化技术来提升系统性能.具体来说,这两个系统会生成多种不同组织结构的数据,并将其缓存起来.然而需要注意的是,缓存将会占用额外的存储空间.如果不加限制地缓存数据,会迅速消耗存储资源.因此,系统采取了一些策略来限制缓存空间:首先,系统为不同组织结构的缓存数据设置了存活时间,超过该时间的数据将被自动清除;此外,系统还设定了一个缓存空间阈值,一旦缓存空间超过该阈值,系统会按照一定的策略删除多余的缓存数据,以保持合理的存储空间使用.这样的缓存管理机制能够确保系统在提高性能的同时,有效地控制存储空间的消耗.

细粒度分割的关键假设是,查询对象经常只出现在数据的部分区域中.因此,系统不必读取数据所有内容.基于这一假设,TASM^[77]、LightDB等系统根据查询情况将每个视频帧分割成若干个不重叠的矩形图块.当需要查询数据中的部分对象时,这些系统不再读取完整的视频帧.取而代之的是,这些系统将过滤掉不包含目标对象的图块,仅读取包含目标对象的图块.这种细粒度的数据存储与读取方式极大地降低了系统的解码开销.

5.3 小结与分析

表4展示了前述视觉数据库管理系统的数据存储部分在针对存储空间优化以及读取效率优化、存储介质两个方面的差异.

- 针对存储空间或读取效率的优化. 上述系统均支持数据存储层面的优化, 但是优化程度却不尽相同. 优化程度最高的一类系统包括 VSS 和 LightDB, 它们从针对存储空间以及读取效率两方面进行了综合考虑. 相比之下, VisualWorldDB 主要专注于存储空间的优化. 然而, 它在读取效率方面可能存在一定的局限性. 另外, ExSample、MIRIS、VaaS、TASTI、Seiden、Focus、VStore 以及 TASM 系统也在不同程度上对读取效率进行了优化. 它们采用了采样、缓存等一系列针对读取过程的优化技术和策略, 以提高数据的检索速度和读取效率.
- 存储介质. 根据系统定位的不同, 系统侧重的硬件环境也有所不同. 具体来说, VSS、LightDB、VisualWorldDB VStore 和 TASM 通常设计为视觉数据库管理系统的存储子系统. 这些系统致力于透明且自动地将数据以高效的组织结构排列在磁盘上, 以满足视觉数据库管理系统对大规模数据存储的需求. 其余系统(包括 ExSample、MIRIS、VaaS、TASTI、Seiden、Focus)则偏向于计算子系统, 它们主要关注在内存中调整数据布局来加速后续的处理工作, 以加快数据的访问速度和处理效率, 进而实现更快速的查询响应.

表 4 数据存储总结

系统	针对存储空间的优化	针对读取效率的优化	存储介质
VSS、LightDB	√	√	磁盘
VisualWorldDB	√	-	磁盘
ExSample、MIRIS、VaaS、TASTI、Seiden、Focus	-	√	内存
VStore、TASM	-	√	磁盘

6 研究展望与未来的趋势

视觉数据库管理系统在众多领域得到了广泛应用, 本节将结合视觉分析领域的前沿工作, 从本文所列举的 4 个研究层面出发, 简要探讨视觉数据库管理系统未来的潜在发展方向.

- 趋于自动化的编程接口. 尽管目前的编程接口具备良好的语义完备性和易用性, 但对不具备专业经验的用户来说, SQL 语言的学习和使用仍需一定的时间投入. 这要求编程接口拥有直观的可视界面并能够智能地为用户推荐查询. 例如: 基于用户的历史查询和目标, 编程接口可以提供查询模板、建议的参数配置, 并识别用户可能感兴趣的数据源和深度学习模型. 因此, 如何针对视觉分析设计自动化编程接口, 将成为未来发展中值得研究的重要话题.
- 面向新硬件的查询优化. 新型硬件加速器层出不穷, 例如: VPU、NPU、TPU 等专用处理器具有强大的计算能力, 能够快速执行复杂的视觉算法和模型. 然而, 与典型的 CPU/GPU 相比, 这类新硬件在计算模式、内存访问、数据传输等方面都存在着不小的差异. 查询优化层面需要充分关注不同的硬件特性, 以优化算子实现、代价估计等方面的策略, 从而最大程度地发挥硬件的潜力. 因此, 如何面向新硬件设计高效的查询优化, 是未来值得关注的研究方向.
- 针对异构平台的执行调度. 为了应对不同的场景, 视觉分析应用需要能够在多种平台上运行, 包括移动设备、边缘设备和云服务器等. 例如: 一个查询可能需要从传感器等边缘设备读取数据, 然后将数据传输至云服务器进行分析计算. 然而, 现有系统中的执行调度层面通常局限于针对单一平台, 如何有效地整合各个平台的计算资源有待进一步探究. 例如: 将部分清洗等预处理操作调度至边缘设备, 以避免将成像模糊的数据传输到云端所产生的网络传输开销. 因此, 如何适应异构平台并设计执行调度也值得深入研究.
- 融合多种技术的数据存储. 现有系统中的存储优化技术(包括冗余消除、采样、索引等技术)分别侧重于优化数据存储层的某一方面, 而这些方面通常相互正交. 因此, 存储层面的技术存在相互融合的可能. 融合这些技术也带来了新的挑战. 一个典型的例子是: 冗余消除技术虽然通过合并存在重叠区域的数据降低了存储空间, 但当采样到该数据时, 合并后的数据却增加了计算开销. 因此, 如何根据应用需求, 深度融合多种技术来进一步挖掘提升空间, 仍是数据存储层面值得探究的前沿方向.

7 总 结

本文首先探究了视觉分析应用的发展历程, 着重强调了深度学习技术在视觉数据库管理系统中起到的关键作用; 然后, 本文总结了视觉数据库管理系统在不同层面上所面临的挑战和相关技术分类, 并且详细介绍了上述分类下的现有技术如何应对这些挑战; 最后, 根据前述分析, 本文展望了视觉数据库管理系统的研究趋势.

References:

- [1] Wang PJ, Yan ZY, Rong XE, *et al.* Review of multimodal data processing techniques with limited data. *Journal of Image and Graphics*, 2022, 27(10): 2803–2834 (in Chinese with English abstract). [doi: 10.11834/jig.220049]
- [2] Lew MS, Sebe N, Djeraba C, *et al.* Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM)*, 2006, 2(1): 1–19.
- [3] Ogle VE, Stonebraker M. Chabot: Retrieval from a relational database of images. *Computer*, 1995, 28(9): 40–48.
- [4] Bach JR, Fuller C, Gupta A, *et al.* The Virage image search engine: An open framework for image management. In: *Proc. of the Storage and Retrieval for Still Image and Video Databases*. 1996. 76–87.
- [5] Dönderler ME, Şaykol E, Arslan U, *et al.* BilVideo: Design and implementation of a video database management system. *Multimedia Tools and Applications*, 2005, 27(1): 79–104.
- [6] Xu Z, Kakkar GT, Arulraj J, *et al.* EVA: A symbolic approach to accelerating exploratory video analytics with materialized views. In: *Proc. of the 2022 Int'l Conf. on Management of Data*. 2022. 602–616.
- [7] Hu W, Xie N, Li L, *et al.* A survey on visual content-based video indexing and retrieval. *IEEE Trans. on Systems*, 2011, 41(6): 797–819.
- [8] Fan J, Gao Y, Luo H. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In: *Proc. of the 12th Annual ACM Int'l Conf. on Multimedia*. 2004. 540–547.
- [9] Flickner M, Sawhney H, Niblack W, *et al.* Query by image and video content: The QBIC system. *Computer*, 1995, 28(9): 23–32.
- [10] Hampapur A. Semantic video indexing: Approach and issues. *ACM SIGMOD Record*, 1999, 28(1): 32–39.
- [11] Cheng SF, Chen W, Sundaram H. Semantic visual templates: Linking visual features to semantics. In: *Proc. of the 1998 Int'l Conf. on Image Processing*. 1998. 531–535.
- [12] Voulodimos A, Doulamis N, Doulamis A, *et al.* Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018, 2018: Article ID 7068349.
- [13] Kang D, Emmons J, Abuzaid F, *et al.* NoScope: Optimizing neural network queries over video at scale. *Proc. of the VLDB Endowment*, 2017, 10(11): 1586–1597.
- [14] Anderson MR, Cafarella M, Ros G, *et al.* Physical representation-based predicate optimization for a visual analytics database. In: *Proc. of the 35th Int'l Conf. on Data Engineering*. 2019. 1466–1477.
- [15] Kang D, Bailis P, Zaharia M. BlazeIt: Optimizing declarative aggregation and limit queries for neural network-based video analytics. *Proc. of the VLDB Endowment*, 2019, 13(4): 533–546.
- [16] Meng LR, Ding GY, Xu C, Qian WN, Zhou AY. Survey of novel video analysis systems based on deep learning. *Ruan Jian Xue Bao/Journal of Software*, 2022, 33(10): 3635–3655 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6631.htm> [doi: 10.13328/j.cnki.jos.006631]
- [17] Ibrahim N, Maurya P, Jafari O, *et al.* A survey of performance optimization in neural network-based video analytics systems. *arXiv:2105.14195*, 2021.
- [18] OpenCV. 2023. <https://opencv.org/>
- [19] Abadi M, Barham P, Chen J, *et al.* TensorFlow: A system for large-scale machine learning. In: *Proc. of the 12th USENIX Symp. on Operating Systems Design and Implementation*. 2016. 265–283.
- [20] PyTorch. 2023. <https://github.com/pytorch/pytorch>
- [21] Kang D, Mathur A, Veeramacheneni T, *et al.* Jointly optimizing preprocessing and inference for DNN-based visual analytics. *Proc. of the VLDB Endowment*, 2020, 14(2): 87–100.

- [22] Cao J, Sarkar K, Hadidi R, *et al.* FiGO: Fine-grained query optimization in video analytics. In: Proc. of the 2022 Int'l Conf. on Management of Data. 2022. 559–572.
- [23] Nakandala S, Kumar A. VISTA: Optimized system for declarative feature transfer from deep CNNs at scale. In: Proc. of the 2020 Int'l Conf. on Management of Data. 2020. 1685–1700.
- [24] Poms A, Crichton W, Hanrahan P, *et al.* Scanner: Efficient video analysis at scale. ACM Trans. on Graphics, 2018, 37(4): 1–13.
- [25] Lu Y, Chowdhery A, Kandula S. Optasia: A relational platform for efficient large-scale video analytics. In: Proc. of the 7th ACM Symp. on Cloud Computing. 2016. 57–70.
- [26] Lu Y, Chowdhery A, Kandula S, *et al.* Accelerating machine learning inference with probabilistic predicates. In: Proc. of the 2018 Int'l Conf. on Management of Data. 2018. 1493–1508.
- [27] Yang Z, Wang Z, Huang Y, *et al.* Optimizing machine learning inference queries with correlative proxy models. Proc. of the VLDB Endowment, 2022, 15(10): 2032–2044.
- [28] Haynes B, Mazumdar A, Balazinska M, *et al.* LightDB: A DBMS for virtual reality video. Proc. of the VLDB Endowment, 2018, 11(10): 1192–1205.
- [29] Romero F, Hauswald J, Partap A, *et al.* Optimizing video analytics with declarative model relationships. Proc. of the VLDB Endowment, 2022, 16(3): 447–460.
- [30] Bastani F, He S, Balasingam A, *et al.* MIRIS: Fast object track queries in video. In: Proc. of the 2020 Int'l Conf. on Management of Data. 2020. 1907–1921.
- [31] Chen Q, Ding G, Xu C, *et al.* Euge: Effective utilization of GPU resources for serving DNN-based video analysis. In: Proc. of the 4th Int'l Joint Conf. on Web and Big Data. 2020. 523–528.
- [32] Crankshaw D, Wang X, Zhou G, *et al.* Clipper: A low-latency online prediction serving system. In: Proc. of the 14th USENIX Symp. on Networked Systems Design and Implementation. 2017. 613–627.
- [33] Wang Y, Yang Y, Zhu W, *et al.* SQLFlow: A bridge between SQL and machine learning. arXiv:2001.06846, 2020.
- [34] He K, Gkioxari G, Dollár P, *et al.* Mask r-CNN. In: Proc. of the Int'l Conf. on Computer Vision. 2017. 2961–2969.
- [35] Koudas N, Li R, Xarchakos I. Video monitoring queries. IEEE Trans. on Knowledge and Data Engineering, 2020, 34(10): 5023–5036.
- [36] Kang D, Gan E, Bailis P, *et al.* Approximate selection with guarantees using proxies. Proc. of the VLDB Endowment, 2020, 13(11): 1990–2003.
- [37] Kang D, Guibas J, Bailis P, *et al.* Accelerating approximate aggregation queries with expensive predicates. Proc. of the VLDB Endowment, 2021, 14(11): 2341–2354.
- [38] Suprem A, Arulraj J, Pu C, *et al.* ODIN: Automated drift detection and recovery in video analytics. Proc. of the VLDB Endowment, 2020, 13(12): 2453–2465.
- [39] Haynes B, Daum M, Mazumdar A, *et al.* VisualWorldDB: A DBMS for the visual world. In: Proc. of the 10th Conf. on Innovative Data Systems Research. 2020. Article No.12.
- [40] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.
- [41] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2017. 7263–7271.
- [42] Gao H, Tian YL, Xu FY, Zhong S. Survey of deep learning model compression and acceleration. Ruan Jian Xue Bao/Journal of Software, 2021, 32(1): 68–92 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6096.htm> [doi: 10.13328/j.cnki.jos.006096]
- [43] Zhuang Z, Tan M, Zhuang B, *et al.* Discrimination-aware channel pruning for deep neural networks. In: Advances in Neural Information Processing Systems. 2018. 875–886.
- [44] Zhang D, Yang J, Ye D, *et al.* LQ-Nets: Learned quantization for highly accurate and compact deep neural networks. In: Proc. of the European Conf. on Computer Vision. 2018. 365–382.
- [45] Novikov A, Podoprikin D, Osokin A, *et al.* Tensorizing neural networks. In: Proc. of the Advances in Neural Information Processing Systems. 2015. 442–450.

- [46] Son S, Nah S, Lee KM. Clustering convolutional kernels to compress deep neural networks. In: Proc. of the European Conf. on Computer Vision. 2018. 216–232.
- [47] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Proc. of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2001. 511–518.
- [48] Sun Y, Wang X, Tang X. Deep convolutional network cascade for facial point detection. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. 2013. 3476–3483.
- [49] Hwang J, Kim M, Kim D, *et al.* CoVA: Exploiting compressed-domain analysis to accelerate video analytics. In: Proc. of the 2022 USENIX Annual Technical Conf. 2022. 707–722.
- [50] Zhang Y, Kumar A. Panorama: A data system for unbounded vocabulary querying over video. Proc. of the VLDB Endowment, 2019, 13(4): 477–491.
- [51] Shen H, Chen L, Jin Y, *et al.* Nexus: A GPU cluster engine for accelerating DNN-based video analysis. In: Proc. of the 27th ACM Symp. on Operating Systems Principles. 2019. 322–337.
- [52] Hsieh K, Ananthanarayanan G, Bodik P, *et al.* Focus: Querying large video datasets with low latency and low cost. In: Proc. of the 13th USENIX Symp. on Operating Systems Design and Implementation. 2018. 269–286.
- [53] Jiang J, Ananthanarayanan G, Bodik P, *et al.* Chameleon: Scalable adaptation of video analytics. In: Proc. of the 2018 Conf. of the ACM Special Interest Group on Data Communication. 2018. 253–266.
- [54] Xu M, Zhu M, Liu Y, *et al.* DeepCache: Principled cache for mobile deep vision. In: Proc. of the 24th Annual Int'l Conf. on Mobile Computing and Networking. 2018. 129–144.
- [55] Nakandala S, Kumar A, Papakonstantinou Y. Incremental and approximate inference for faster occlusion-based deep CNN explanations. In: Proc. of the 2019 Int'l Conf. on Management of Data. 2019. 1589–1606.
- [56] Wang J, Balazinska M, Deluceva: Delta-based neural network inference for fast video analytics. In: Proc. of the 32nd Int'l Conf. on Scientific and Statistical Database Management. 2020. 1–12.
- [57] Park K, Saur K, Banda D, *et al.* End-to-end optimization of machine learning prediction queries. In: Proc. of the Int'l Conf. on the Management of Data. 2022. 587–601.
- [58] Chaiken R, Jenkins B, Larson PÅ, *et al.* Scope: Easy and efficient parallel processing of massive data sets. Proc. of the VLDB Endowment, 2008, 1(2): 1265–1276.
- [59] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM, 2008, 51(1): 107–113.
- [60] Zaharia M, Chowdhury M, Das T, *et al.* Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: Proc. of the 9th USENIX Symp. on Networked Systems Design and Implementation. 2012. 15–28.
- [61] Carbone P, Katsifodimos A, Ewen S, *et al.* Apache Flink™: Stream and batch processing in a single engine. The Bulletin of the Technical Committee on Data Engineering, 2015, 38(4):28–38.
- [62] Cheng XQ, Jin XL, Wang YZ, GUO JF, Zhang TY, Li GJ. Survey on big data system and analytic technology. Ruan Jian Xue Bao/Journal of Software, 2014, 25(9): 1889–1908 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4674.htm> [doi: 10.13328/j.cnki.jos.004674]
- [63] Zhang H, Ananthanarayanan G, Bodik P, *et al.* Live video analytics at scale with approximation and delay-tolerance. In: Proc. of the 14th USENIX Symp. on Networked Systems Design and Implementation. 2017. 377–392.
- [64] Kannan RS, Subramanian L, Raju A, *et al.* GrandsLAM: Guaranteeing SLAs for jobs in microservices execution frameworks. In: Proc. of the 14th EuroSys Conf. 2019. 1–16.
- [65] Ao L, Izhikevich L, Voelker GM, *et al.* Sprocket: A serverless video processing framework. In: Proc. of the ACM Symp. on Cloud Computing. 2018. 263–274.
- [66] Romero F, Zhao M, Yadwadkar NJ, *et al.* LLAMA: A heterogeneous & serverless framework for auto-tuning video analytics pipelines. In: Proc. of the ACM Symp. on Cloud Computing. 2021. 1–17.
- [67] Hindman B, Konwinski A, Zaharia M, *et al.* Mesos: A platform for fine-grained resource sharing in the data center. In: Proc. of the 8th USENIX Symp. on Networked Systems Design and Implementation. 2011. 295–308.

- [68] Vavilapalli VK, Murthy AC, Douglas C, *et al.* Apache hadoop YARN: Yet another resource negotiator. In: Proc. of the 4th Annual Symp. on Cloud Computing. 2013. 1–16.
- [69] Verma A, Pedrosa L, Korupolu M, *et al.* Large-scale cluster management at Google with Borg. In: Proc. of the 10th European Conf. on Computer Systems. 2015. 1–17.
- [70] Haynes B, Daum M, He D, *et al.* VSS: A storage system for video analytics. In: Proc. of the Int'l Conf. on Management of Data. 2021. 685–696.
- [71] Moll O, Bastani F, Madden S, *et al.* ExSample: Efficient searches on video repositories through adaptive sampling. In: Proc. of the IEEE 38th Int'l Conf. on Data Engineering. 2022. 2956–2968.
- [72] Bastani F, Moll O, Madden S. Vaas: Video analytics at scale. Proc. of the VLDB Endowment, 2020, 13(12): 2877–2880.
- [73] Kang D, Guibas J, Bailis PD, *et al.* TASTI: Semantic indexes for machine learning-based queries over unstructured data. In: Proc. of the Int'l Conf. on Management of Data. 2022. 1934–1947.
- [74] YOLOv5. 2023. <https://github.com/ultralytics/yolov5>
- [75] Bang J, Kakkar GT, Chunduri P, *et al.* SEIDEN: Revisiting query processing in video database systems. Proc. of the VLDB Endowment, 2023, 16(9): 2289–2301.
- [76] Xu T, Botelho LM, Lin FX. VStore: A data store for analytics on large videos. In: Proc. of the 14th EuroSys Conf. 2019. 1–17.
- [77] Daum M, Haynes B, He D, *et al.* TASM: A tile-based storage manager for video analytics. In: Proc. of the 37th Int'l Conf. on Data Engineering. 2021. 1775–1786.

附中文参考文献:

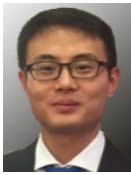
- [1] 王佩瑾, 闫志远, 容雪娥, 等. 数据受限条件下的多模态处理技术综述. 中国图像图形学报, 2020, 27(10): 2803–2834. [doi: 10.11834/jig.220049]
- [16] 孟令睿, 丁光耀, 徐辰, 钱卫宁, 周傲英. 基于深度学习的新型视频分析系统综述. 软件学报, 2022, 33(10): 3635–3655. <http://www.jos.org.cn/1000-9825/6631.htm> [doi: 10.13328/j.cnki.jos.006631]
- [42] 高晗, 田育龙, 许封元, 仲盛. 深度学习模型压缩与加速综述. 软件学报, 2021, 32(1): 68–92. <http://www.jos.org.cn/1000-9825/6096.htm> [doi: 10.13328/j.cnki.jos.006096]
- [62] 程学旗, 靳小龙, 王元卓, 郭嘉丰, 张铁赢, 李国杰. 大数据系统和分析技术综述. 软件学报, 2014, 25(9): 1889–1908. <http://www.jos.org.cn/1000-9825/4674.htm> [doi: 10.13328/j.cnki.jos.004674]



丁光耀(1996—), 男, 博士生, CCF 学生会员, 主要研究领域为大规模数据处理系统, 智能计算系统.



钱卫宁(1976—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为可扩展事务处理, 大数据管理系统基准评测, 海量数据分析处理及其应用, 数据驱动的计算教育学.



徐辰(1988—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为大规模数据处理系统, 数据库系统, 智能计算系统.



周傲英(1965—), 男, 博士, 教授, 博士生导师, CCF 会士, 主要研究领域为数据库, 数据管理, 数据驱动的教授教育学, 教育科技、物流科技等基于数据的应用.