软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(3):1292−1309 [doi: 10.13328/j.cnki.jos.006790]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于宽容训练和隐私保护的快速监控视频检索模型*

覃 浩[1,2], 王平辉[1,2], 张若非[1,2], 覃遵颖[3]

[1](西安交通大学 网络空间安全学院, 陕西 西安 710049)

[2](智能网络与网络安全教育部重点实验室(西安交通大学), 陕西 西安 710049)

[3](西安交通大学 软件学院, 陕西 西安 710049)

通信作者: 王平辉, E-mail: phwang@mail.xjtu.edu.cn

摘 要: 监控视频关键帧检索和属性查找在交通、安防、教育等领域具有众多应用场景, 应用深度学习模型处理海量视频数据在一定程度上缓解了人力消耗, 但是存在隐私泄露、计算资源消耗大、时间长等特点. 基于上述场景, 提出了一个面向大规模监控视频的安全、快速的视频检索模型. 具体地, 根据云端算力大、监控摄像头算力规模小的特点, 在云端部署重量级模型, 并使用所提出的宽容训练策略对其进行定制化知识蒸馏, 将蒸馏后的轻量级模型部署在监控摄像头内, 同时使用局部加密算法对图像敏感部分进行加密, 结合云端 TEE 技术和用户授权机制, 在极低资源消耗的情况下实现隐私保护. 通过合理控制蒸馏策略的"容忍度", 能够较好地平衡摄像头视频输入阶段和云端检索阶段的耗时, 在保证极高准确率的前提下, 保证极低的检索时延. 相比于传统检索方法, 该模型具有安全高效、可伸缩、低延时的特点. 实验结果显示, 在多个公开数据集上, 该模型相比于传统检索方法提供 9×−133×的加速.

关键词: 视频检索; 隐私保护; 知识蒸馏; 课程学习

中图法分类号: TP391

中文引用格式: 覃浩, 王平辉, 张若非, 覃遵颖. 基于宽容训练和隐私保护的快速监控视频检索模型. 软件学报, 2023, 34(3): 1292−1309. http://www.jos.org.cn/1000-9825/6790.htm

英文引用格式: Qin H, Wang PH, Zhang RF, Qin ZY. Fast Surveillance Video Retrieval Model Based on Tolerant Training and Privacy Protection. Ruan Jian Xue Bao/Journal of Software, 2023, 34(3): 1292−1309 (in Chinese). http://www.jos.org.cn/1000-9825/6790.htm

Fast Surveillance Video Retrieval Model Based on Tolerant Training and Privacy Protection

QIN Hao[1,2], WANG Ping-Hui[1,2], ZHANG Ruo-Fei[1,2], QIN Zun-Ying[3]

[1](School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

[2](Ministry of Education Key Laboratory for Intelligent Networks and Network Security (Xi'an Jiaotong University), Xi'an 710049, China)

[3](School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Surveillance video keyframe retrieval and attribute search have many application scenarios in traffic, security, education and other fields. The application of deep learning model to process massive video data to a certain extent alleviates manpower consumption, but it is characterized by privacy disclosure, large consumption of computing resources and long time. Based on the above scenarios, this study proposes a safe and fast video retrieval model for mass surveillance video. In particular, according to the characteristics of large computing power in the cloud and small scale of computing power in the surveillance camera, heavyweight model is deployed in the cloud, and the proposed tolerance training strategy is used for customized knowledge distillation, the distilled lightweight model is then deployed inside a surveillance camera, at the same time using local encryption algorithm to encrypt sensitive to image part, combined with cloud

* 基金项目: 国家自然科学基金(61902305, 61922067); 深圳基础研究资助项目(JCYJ20170816100819428); 教育部-中国移动"人工智能"项目(MCM20190701)

本文由"大数据治理的理论与技术"专题特约编辑杜小勇教授、杨晓春教授和童咏昕教授推荐.

收稿时间: 2022-05-15; 修改时间: 2022-07-29, 2022-09-07; 采用时间: 2022-09-23; jos 在线出版时间: 2022-10-27

TEE technology and user authorization mechanism, privacy protection can be achieved with very low resource consumption. By reasonably controlling the “tolerance” of distillation strategy, the time-consuming of camera video input stage and cloud retrieval stage can be balanced, and extremely low retrieval delay is ensured on the premise of extremely high accuracy. Compared with traditional retrieval methods, the proposed model has the characteristics of security, efficiency, scalability and low latency. Experimental results show that the proposed model provides 9×−133×acceleration compared with traditional retrieval methods on multiple open data sets.

Key words: video retrieval; privacy protection; knowledge distillation; curriculum learning

监控摄像头在我们日常生活中无处不在, 既有在交通、企业、校园等公共场所的公共摄像头, 也有一些住户安装的住宅私有摄像头. 这些摄像头通常记录大量的监控视频资源, 视频资源常常用于事故后溯源如事故车辆查找、场景信息收集和分析如人脸识别等, 这些应用场景对于视频检索系统的响应速度、安全性、用户隐私保护等方面都提出了极高的要求.

在深度学习发展之前, 检索需要依赖大量的人力投入, 通常耗时数小时甚至数天才能完全处理所有视频资源. 而随着深度学习的应用, 在视频检索上应用深度学习技术进行自动化处理已经被广泛采用, 现有方法大多直接使用高精度目标检测和属性分类网络, 或者训练大规模统一预训练模型, 然后在特定领域微调生成专用模型后对视频进行检索, 这在实际应用中存在检索时间和硬件成本高、用户隐私泄漏等问题.

- 首先是极高的时间成本. 越高的精度, 在一定程度上意味着参数量的剧烈上升. 例如在 CIFAR-100 数据集上, 具有 76.5%准确率的 ResNet50 模型[1]包含 25.6 M 参数, 而具有 81.2%准确率的 ResNet152[1], 其参数量增长到 60.1 M. 参数量的增加不仅需要更多的存储空间, 在推理阶段也需要更大的计算量, 导致推理速度变慢, 这在时延要求高的场景很不适用. 例如对一个长达一个月的监控视频进行实时查询, 使用 YOLO+VGG 模型[2,3]需要在 GPU 上连续运行 5 个小时. 然而事实上, 大多数的监控视频检索具有很高的时延要求, 通常需要分钟级甚至秒级的响应时间, 现有模型不能满足这种需求.
- 其次是极高的硬件成本. 在硬件支持上, 高精度模型势必会依赖高性能 GPU 资源. 例如, 使用精度较高的目标检测网络 YOLO[2]在 NVIDIA K80 GPU 上最高只能达到 50 f/s, 在 NVIDIA Tesla V100 GPU 上却能够达到 80 f/s[4]. 随着预训练模型[5,6]的发展, 模型参数规模在数量级上也急剧上升, 导致普通的 GPU 已经不能满足训练要求, 需要大量的 TPU 进行长时间的预训练, 造成预训练模型在实际应用部署中会产生很大的经济代价. 而与此同时, IoT 设备的发展也使得一些移动设备和摄像头拥有一定的计算能力, 边缘设备的算力增强, 使得在 IoT 设备上应用深度学习模型成为可能, 因此亟需一个边缘计算和云端算力结合的模型来压缩硬件成本.
- 最后是用户隐私泄漏问题. 在家庭、公司、学校的监控系统中, 通常包含大量用户敏感信息, 然而现有的监控视频检索模型通常直接对未加密用户的监控视频进行分析, 这导致在视频存储、传输、计算的整个过程图像以明文的方式暴露在第三方服务商或网络中, 一旦用户隐私泄漏会产生无法估算的后果. 同时, 由于基于同态加密和查分隐私的隐私保护方法计算复杂度过高, 在摄像头这样拥有极低计算资源的边缘设备中无法实现, 因此亟需一个高效的隐私保护策略保障用户隐私安全.

针对上述问题, 本文提出了一个全新的基于隐私保护的视频检索范式. 首先, 我们提出了宽容训练策略(tolerant training strategy, TTS)方法, 方法使用新颖的置信度容忍(confidence tolerance, CT)蒸馏损失和课程学习(curriculum learning, CL)对原始教师网络(vanilla teacher neural network, VTNN)进行知识蒸馏, 得到定制化小模型; 然后融入两阶段过滤模型(two-stage filtration model, TFM), 分别在视频流收集阶段和查询阶段使用小模型和大模型对目标帧和实体进行筛选. 我们可以根据实际应用中摄像头等边缘端算力和云端算力的差异, 通过控制 TTS 策略的“容忍度”, 合理控制定制化小模型的尺寸, 平衡收集阶段和查询阶段的负载和时延, 显著降低检索时延. 相比于传统的知识蒸馏训练方法, 我们的知识蒸馏方法在 TFM 框架下具有更好的收敛性和泛化能力, 对比传统视频检索, 我们的模型拥有更好的可伸缩性、更高的精度和更低的时延. 另一方面, 我们提出的基于 logistic 的局部混沌加密方法在几乎不消耗资源的情况下进行加密操作, 结合云端 TEE (trusted execution environment)技术和基于非对称加密的用户授权机制, 能够很好地保护用户隐私. 实验结果显示, 和

传统视频检索方法对比，我们的模型能够提供 9×–133×的加速. 此外，我们的模型还能提供图像检索功能,可支持更广泛的应用场景.

综上所述，本文的贡献有如下 4 个方面.

(1) 我们设计了新颖的置信度容忍蒸馏损失函数(CT loss)，并应用于学生模型知识蒸馏. 实验显示，相比于直接训练学生模型或传统知识蒸馏，该损失能使学生模型更好地学习教师模型的知识分布，起到一定程度正则化的效果，使模型拥有更好的泛化能力.

(2) 我们提出了新颖的宽容教师训练策略(TTS)知识蒸馏框架，该策略结合了置信度容忍蒸馏损失和难度自适应课程学习方法，其中，课程学习的引入，大大提高了模型收敛速度，结合自适应难度评估机制，实现对样本难度的动态评估. 实验结果显示，该方法在蒸馏过程中能够显著降低模型训练时间，提升模型 top-k 召回率.

(3) 我们提出了基于 logistic 矩阵局部加密、TEE 技术和用户授权的隐私保护策略，在消耗极低的计算资源的前提下，保证在传输过程、第三方云端环境不会发生隐私泄露，保证用户的隐私安全.

(4) 我们设计了具有可伸缩性的两阶段过滤模型(two-stage filtration model, TFM)，分别在视频流收集阶段和查询阶段对视频进行处理，并利用 CT Loss 置信度和宽容训练策略的“宽容度”指标来控制模型压缩比例和精确度的关系，从而分配视频检索在视频输入和查询阶段的时延. 在多个开放数据集上显示，该方法能够显著降低检索总时间消耗，提供最高 133×加速.

我们将本模型应用于监控视频检索任务中，包括多个公开数据集如 Stanford Cars、BornSpeed 监控数据集以及其他监控数据集中，进行了广泛的实验对比和消融实验. 通过理论和实验分析，我们提出的模型取得了卓越的性能，证明了本方法使用的蒸馏损失、训练策略和模型的有效性，以及隐私保护策略的安全性.

1 相关工作

1.1 监控视频隐私保护

监控视频检索通常需要很大的数据量，在摄像头等边缘端很难实现，需要将这些数据上传到云端进行分析. 但是监控数据通常包含了用户大量的隐私信息，因此用户不希望这些敏感信息被观测到. 为了让保障敏感信息的数据安全，必须考虑传输过程中的加密和计算过程的安全性.

监控视频隐私保护分为两个步骤. 首先是识别隐私区域，大多使用运动物体检测[7]、人脸检测[8]等方法；其次是隐私区域保护，有诸如隐私区域替换[9]、数据分割[10]等编码前保护方法，也有帧间模式加密[11]、运动矢量加密[12]、熵编码过程加密[13]等结合编码保护方法，也有基于数据隐藏[14]的编码后保护方法.

其次是隐私区域加密技术，如基于同态加密的方法将同态加密和深度学习相结合[15]. 基于同态加密的隐私保护分为训练、推理和结果这 3 个阶段：训练阶段，客户端使用同态加密对数据集进行加密，并发送给云端训练模型；推理阶段，客户端发送测试数据并直接作为模型输入得到结果；最后在结果阶段，将结果进行加密返回给客户端. 基于安全多方计算的隐私保护[16]则通过乱码电路来保护用户的隐私，客户端和服务器之间使用无关传输通信来进行安全保证. 另一种基于差分隐私的隐私保护方法则对是数据进行适当的加噪加扰，进而达到数据隐私与可用性之间的平衡.

另一方面，由于安全计算的计算开销和通信开销较高，基于 TEE 的“传输加密，计算明文”的思路被应用于可信硬件上，代表性的有 IntelSGX 技术，其通过加密内存和用户空间安全隔离的方式来保护关键代码和数据的安全性. Ryoan 等人[17]利用 InterSGX 和分布式沙箱技术保护数据所有者的数据免受不可信方的窃取，在计算平台不可信的情况下，依然能够提供对沙箱隐私数据的保护. Haven 等人[18]使用基于微软的 Drawbridge 沙箱机制提供了粗粒度的隔离应用程序的安全容器，将应用程序、标准库等放入其中，保护应用程序抵御特权软件或者外部物理攻击.

1.2 课程学习

课程学习(curriculum learning, CL)来源于人类从易到难的学习过程，通过设置不同难度的“课程”的方式，

让模型从简单的样本开始学习, 并逐渐提升样本的复杂度. 大量研究和实验表明, 课程学习可以显著加速模型的训练时间, 在达到相同模型性能的情况下, 模型收敛的速度较普通训练方法快, 模型通常也能获得更好的泛化能力.

课程学习的概念最初由 Bengio 等人[19]提出, 简而言之, 可概括为训练过程为“从简单到复杂”. 课程学习的思想是一种机器学习任务的训练策略, 而不是一种针对特定场景的模型, 因此在各个领域都有广泛应用, 包括计算机视觉[20,21]、自然语言处理中的监督学习任务[22,23]、医疗保健预测[24]、强化学习[25−27]、图学习[28]甚至神经架构搜索[29]等.

一个通用的课程学习框架通常需要考虑两个事情, 即如何定义数据的难易程度和如何处理更难的数据. 目前的工作对难易程度的定义主要集中在数据多样性、噪声估计值、内容复杂度等方面, 例如 CV 中的物体数量、图像锐度等, NLP 中的句子长度、语法解析树深度等. 对不同难度数据的调度主要有预设置的课程学习和自适应的课程学习两种方法: 预设置方法中, 通常预先设定好难度度量规则对数据进行排列; 自适应则较为灵活, 可以使用子网络自身的训练损失调整训练顺序, 即自学习, 或者使用额外的预训练数据集对训练集数据进行预训练, 得到难度排序, 甚至使用强化学习策略, 根据子网络训练反馈进行调节.

1.3 知识蒸馏

近年来, 深度神经网络在工业界和学术界都取得了成功, 尤其是在计算机视觉任务方面. 深度学习模型拥有更大的知识容量, 能够对大规模数据编码进行拟合, 然而, 将这些重量级的深度模型部署在存储和计算资源有限的设备, 如手机或 IoT 设备上是一个挑战. 为此, 需要对大模型的参数规模进行压缩, 作为模型压缩和加速的代表类型, 知识蒸馏能够有效地从大型教师模型中学习小型学生模型.

知识蒸馏的基本思想正是让卷积神经网络模仿人类的学习行为, 将大型网络学习到的知识提炼传授给小型网络, 并指导小型网络的训练. Hinton 等人[30]首次提出基于 softmax 输出层的知识蒸馏, 是知识蒸馏领域的开山之作. 他们提出在 softmax 层加入“软”标签来指导学生网络训练. 后来, Romero 等人[31]提出了 FiNet 对于网络中间隐藏层进行知识蒸馏, 对学生网络进行更加全面的指导. Zagoruyko 等人[32]在 FiNet 基础上使用注意力转移(attention transfer, AT)机制进行改进, 进一步提升了蒸馏性能. 除此之外, 对于网络容量相差较大, 或者不同结构的神经网络之间, Kim 等人[33]提出了特征图通道扩展的因子传输(factor transfer, FT)方法, 进一步解释教师的知识, 帮助学生网络学习.

1.4 视频检索

视频检索是指给定一组查询信息和一个候选视频数据库, 从数据库中选择与查询信息相关的视频(帧). 查询信息可以是文本, 对应文本-视频检索; 或者图像, 对应图像-视频检索. 针对监控视频这种应用场景, 视频检索的目标是找出视频流中所有包含特定类别物体的帧集合, 或者从视频流中找出和包含相似目标图像的帧集合. 整个检索过程通常分为两部分, 分别为视频特征提取阶段和检索阶段.

视频特征提取阶段包含两大子任务: 目标检测和属性分类. 目标检测任务最早可以追溯到 20 世纪 60 年代. 传统方法主要是一些经典的机器学习分类和聚类算法, 结合特征工程例如 SIFT[34]来进行目标检测. 后来经过发展, 产生了一些诸如 HOG[35]、DPM (deformable parts model)[36]和选择搜索(selective search)的复杂算法, 这些方法在一定程度上依赖人工特征筛选. 直到人工神经网络的快速发展, 使得基于卷积神经网络的深度学习算法达到甚至超过了人类的水平, 因此, 基于深度神经网络的模型成为目标检测和图像分类领域的首选方案.

现有的目标检测通常有两类.

- 一类是两阶段检测器: 在第 1 阶段, 使用区域建议网络(region proposal network, RPN), 提出候选目标边界框; 第 2 阶段, 通过 RoI (region of interest)操作, 从每个候选框中提取特征. 两阶段检测器最具代表性的是 R-CNN 系列, 包括 R-CNN[37]和 Fast R-CNN[38]. 实验结果表明, 在 PASCAL VOC 2007 数据集[39]上, Fast R-CNN 的 mAP 为 66.9%, 而 R-CNN 为 66.0%. 在提出 Fast R-CNN 后, Faster R-CNN[40]

进一步改善了基于区域的 CNN 基线. 此后, He 等人[41]对 Faster R-CNN 进行了拓展, 提出了更准确的目标检测器 Mask R-CNN.

- 除此之外, 另一大类是单阶段检测器, 即直接从输入图像中提出预测框, 不需要区域建议步长, 如 SSD[42]、YOLO[2]以及它的扩展版本 YOLOv3[43]、YOLOv4[44], 还有在计算资源有限情况下的优化版 YOLOv4-tiny、YOLOv5-tiny.

两阶段检测器具有较高的定位和目标识别精度, 而单阶段检测器具有较高的推理速度, 可用于实时设备.

属性分类任务是图像分类的一种, LeCun 等人提出的 LeNet[45]是最早卷积神经网络模型, 由于其结构相对简单, 因此无法处理复杂模型. 后来发展了包括 AlexNet[46]、ResNet[1]系列、VGG[3]等经典模型, 以及基于注意力机制的深度卷积神经网络模型和基于神经架构搜索(neural architecture search, NAS)[47]的深度卷积神经网络模型. 这些模型通常拥有可扩展的深度以及较大的网络容量, 已经能够充分拟合大规模数据特征, 达到较高的精度.

检索阶段一般基于特征提取, 对视频帧进行倒排索引或者构建一个向量检索数据库[48], 根据查询信息, 直接查找索引, 或者将查询信息转换为向量, 在向量检索数据库中进行基于相似度或者其他度量的搜索.

从整体上看, 虽然在数据特征提取阶段拥有大量备选算法, 但是基于大规模视频数据的快速检索几乎都是使用模型进行暴力推理分析, 即基于已有视频数据进行遍历特征抽取和检索, 无法实现实时或者亚实时的搜索要求.

另一方面, 随着边缘算力的发展, 出现了一些云边协同的模型训练和部署方法, 在边缘端部署深度学习模型负责模型的推理, 在云端负责深度学习模型的集中式训练, 实现分布式智能. Li 等人[49]设计了一个边缘协同 DNN 推理框架, 使用自适应的设备和边缘分区来进行实时 DNN 推理. Grulich 等人[50]设计了一个云边协作模型来进行实时视频处理, 通过压缩和通信差异传输算法, 大幅度降低了视频延迟. 而在视频检索领域, 目前缺乏一个云边协同的模型来压缩检索时间.

2 快速监控视频检索模型

在本节中, 首先在第 2.1 节介绍大规模视频检索任务定义和总体框架, 在第 2.2 节介绍基于 IoT 设备的边缘视频流输入端框架, 在第 2.3 节介绍查询阶段框架, 在第 2.4 节阐述局部加密和隐私保护机制, 第 2.5 节介绍使用宽容教师训练策略进行知识蒸馏的整体框架细节. 整体模型结构如图 1 所示.

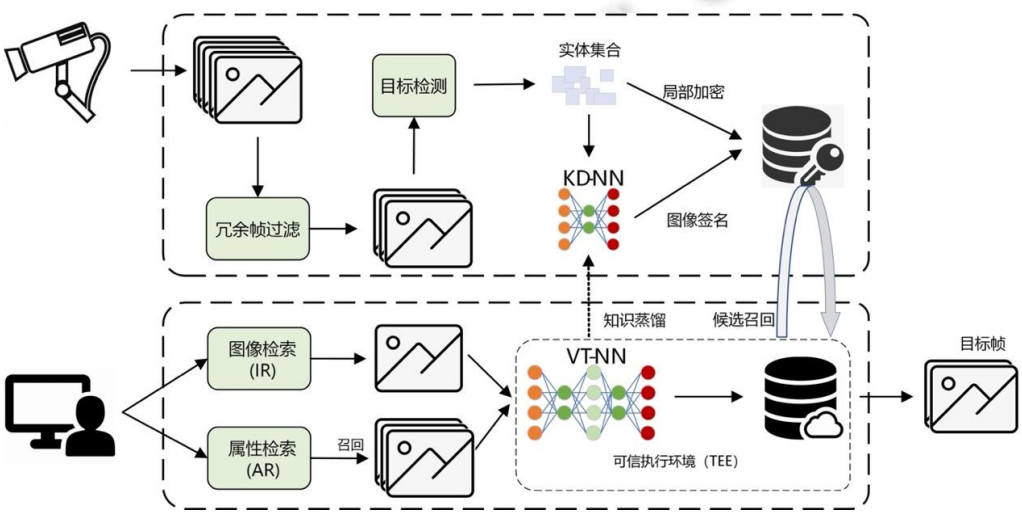

图 1 两阶段过滤模型(TFM)整体框架

2.1 任务定义

给定一个含有若干连续帧的视频图像序列 $S=f_1,f_2,\ldots,f_n$, 对于单帧视频图像, 检索系统将其转化为 m 维稠密图像向量 $I\in\mathbb{R}^{1\times m}$, 整个视频索引以向量矩阵 $M\in\mathbb{R}^{n\times m}$ 形式存储在检索引擎中.

我们定义单次查询输入为细粒度属性信息 Q_a, 检索模型将召回所有视频帧向量中最大概率类别为 Q_a, 即 $1[Index(\mathrm{Max}(I))=Q_a]$为 1 时, 召回此向量对应的帧, 否则不召回.

若单次查询输入为图像信息 Q_i, 检索模型首先通过编码器将原始像素信息转化为特征向量 V_i 后, 再计算特征向量和视频帧向量库中视频帧向量的相似度 $S_a(V_i,I)$, 找到 $S_a(V_i,I)>T$, 即相似度高于某个阈值 T 的向量集合, 通过向量索引召回所有相关帧集合 $U=\{\{f_{s1},\ldots,f_{e1}\},\{f_{s2},\ldots,f_{e2}\},\ldots,\{f_{sk},\ldots,f_{ek}\}\}$, 其中, $f_{s\cdot}$表示起始帧, $f_{e\cdot}$表示结束帧.

由于监控视频的时间和空间局部性特点, 通常相关帧集合是分段连续的, 因此, 召回的帧集合通常是若干个时间区间.

2.2 视频流输入阶段

在视频流输入阶段, 边缘端的完整处理流程包含几个步骤, 如图 1 上半部分所示. 对于一个视频流的帧序列输入 $S=\{f_1,f_2,\ldots,f_n\}$, 首先进行冗余帧过滤操作, 其方法是对此帧进行二分类. 我们训练了一个轻量级二分类网络 Φ, 将视频流逐帧输入到此二分类网络中, 判断当前帧是否包含特定目标属性物体, 得到结果 $y_i=\Phi(f_i)\in\{0,1\}$. 若 $y_i=0$, 即判断结果为不包含, 则跳过冗余帧, 避免后续的目标检测和属性分类网络的额外消耗. 此操作基于一个基本假设: 在实际监控视频中, 视频流输入数据量往往很大, 但是含有有用信息的视频帧占比非常小. 例如, 在 Focus[51]使用的数据集中, 每类物体平均在视频帧中出现占比只有不到 0.16%, 出现最频繁的几类占比也在 26%–78%之间; 在 Noscope[52]使用的交通监控数据集中, 夜晚情况下, 存在车辆的帧数不到总数的 10%. 特别地, 如果针对更细粒度的属性筛选, 目标帧占比则更小. 实验结果显示, 在夜晚、郊区、车流量小的监控场景下, 最多能过滤 94%的帧.

得到了过滤后的帧, 可用于后续目标检测. 目标检测阶段使用的轻型嵌入端模型 yolov4-tiny, 可以在缺乏 GPU 算力的边缘端运行, 对第 i 帧视频, 识别得到 n 个物体 $O_i=\{o_{i1},o_{i2},\ldots,o_{in}\}$. 随后使用快速局部加密将关键物体信息加密, 具体在第 2.4 节阐述. 同时, 将这 n 个物体将全部被送入细粒度属性编码器 $g(x;\theta_s)$中, 其中, x 表示输入, θ_s 代表此学生网络参数. 注意, 此网络为轻量级学生蒸馏模型.

输入第 j 个物体, 得到特征向量:

$$v_i=g(o_{ij};\theta_s) \tag{1}$$

在特征向量之后, 连接多分类器对物体进行细粒度分类:

$$y_{ij}=softmax(w_s\cdot g(o_{ij};\theta_s)+b) \tag{2}$$

其中, w 和 b 为分类器参数, y_{ij} 表示第 i 帧视频第 j 个物体的分类结果. 在大规模数据检索过程中, 对数据建立索引是一个常用的技术, 在此任务中, 我们对分类结果置信度最高的 k 个分类结果$\{y_{ij1},y_{ij2},\ldots,y_{ijk}\}$来对视频帧做倒排索引. 其索引项为:

$$object\ class\rightarrow\langle object\ ID,frames\ ID\rangle \tag{3}$$

其中, 主键设置为类别, 方便查询阶段进行检索. 另外, 再对特征向量做索引, 类似地, 其索引项为:

$$v_i\rightarrow\langle object,frames\ ID\rangle \tag{4}$$

向量将被存入向量引擎中, 使用聚类分桶(inverted file system, IVF)和乘积量化(product quantizer, PQ)优化向量索引, 以便于更快召回候选集.

2.3 查询阶段

如图 1 下半部分所示, 在查询阶段, 主要处理图像索引(image retrieval, IR)和属性检索(attribute retrieval, AR)两种查询方式. 图像检索将目标图像 o 输入到教师编码模型 $f(x;\theta_t)$中, 得到其向量表示:

$$v_{input}=f(o;\theta_t) \tag{5}$$

在向量数据库中进行相似度检索, 通过向量倒排索引召回高于指定阈值的所有帧.

属性检索方式直接使用属性值在数据库中进行候选召回得到候选集, 将每个候选集中的 *object* 输入到教师编码模型中进行编码, 并进行高精度分类:

$$\hat{y} = softmax(w_t \cdot f(o;\theta_t)) \tag{6}$$

将分类结果 $\hat{y}$ 作为此 *object* 的基准(ground truth, GT)标签, 并通过此标签类别的索引找到所有含有基准标签类别的帧.

2.4 局部加密和隐私保护

在一些非公众场合下, 如家庭监控或者包含其他敏感信息如人脸等监控视频中, 我们希望对视频中敏感部分进行加密, 避免隐私泄露. 传统的同态加密等方法对计算量要求极大, 因此我们采用局部加密、TEE 技术和用户授权结合的方式, 分别保证存储安全、计算安全和传输安全.

具体到我们的 TFM 模型中, 我们希望在摄像头内部直接对图像进行局部加密. 我们在视频流输入阶段已经使用目标检测获得敏感区域的像素位置和大小, 因此我们只需对这部分进行加密; 我们使用 logistic 混沌序列函数生成随机序列:

$$x_{n+1}=\mu x_n(1-x_n),\ 0<x<1 \tag{7}$$

当μ>3.570 时, 系统进入混沌状态, 此时我们对其进行迭代, 生成一个混沌矩阵 *M*, 如果我们对输入 *x* 施加一个极小的扰动后, 会产生完全不同的矩阵 *M*. 我们利用混沌矩阵 *M* 与像素异或的方式加密图像:

$$I'=I\oplus M \tag{8}$$

解密图像时, 再次对图像进行异或即可. 如算法 1 所描述, 由于在摄像头环境下 CPU 资源十分宝贵, 我们不希望每次加密都生成一个全新的混沌矩阵, 因此我们预先生成一个全局混沌矩阵, 并且在每次加密时使用不同的出发点, 出发点即混沌矩阵的行、列坐标. 设定不同的出发点, 能够避免云端请求一次混沌初始值就能解密所有图像的行为. 当生成的混沌矩阵足够大时, 图像加密能够保证足够安全. 除此之外, 我们定义混沌游走方式 *Next*(·)以行、列或对角线共 8 个方向达到下一个矩阵坐标, 一般定义为按行增加,列增加方向即可.

算法 1. 快速混沌加密算法.

输入: *I*: 输入图像; *M*: 混沌矩阵; (c_0,r_0): 出发点; *Next*(·): 混沌游走方式.

输出: *I′*: 加密后的图像.

```
1.  //步骤 1: 预定义加密混沌矩阵, 只在初始化执行一次.
2.  Initialize(x,μ);
3.  for i=1 to M do:
4.      for j=1 to N do:
5.          M[i][j]=x·factor mod 256;   //放缩到像素值范围内, factor>256, 为放缩因子
6.          x=μ·x·(1−x);
7.      end for
8.  end for
9.  //步骤 2: 局部加密
10. I=[r,g,b]   //展开图像三通道
11. Prev=x
12. for i=0 to length(I) do:
13.     for j=0 to width(I) do:
14.     //混沌游走到下一点
15.         c_{i·length(M)+j},r_{i·length(M)+j}=Next(c_{i·length(M)+j−1},r_{i·length(M)+j−1});
16.         v = M[c_{i·length(M)+j}][r_{i·length(M)+j}];
```

17. $r'_{ij} = r_{ij} \oplus v$;
18. $g'_{ij} = g_{ij} \oplus v$;
19. $b'_{ij} = b_{ij} \oplus v$;
20. $I'_{ij} = [r'_{ij}, g'_{ij}, b'_{ij}]$;
21. **end for**
22. **end for**
23. **return** I'

由于我们的数据仅在查询阶段才会在云端被访问, 因此我们能够很好地避免将数据直接传输到云端进行存储. 在查询阶段, 云端会召回所有候选图像, 在图像进行传输的过程中, 为了保证传输目标是一个可信实体,我们会进行基于非对称加密的授权验证. 我们假设摄像头和云端分别有一个公钥和私钥, 如图 2 所示, 其中, *Public Key of A*[·]意为使用 *A* 的公钥进行加密, *Private Key of A*[·]意为使用 *A* 的私钥进行解密.

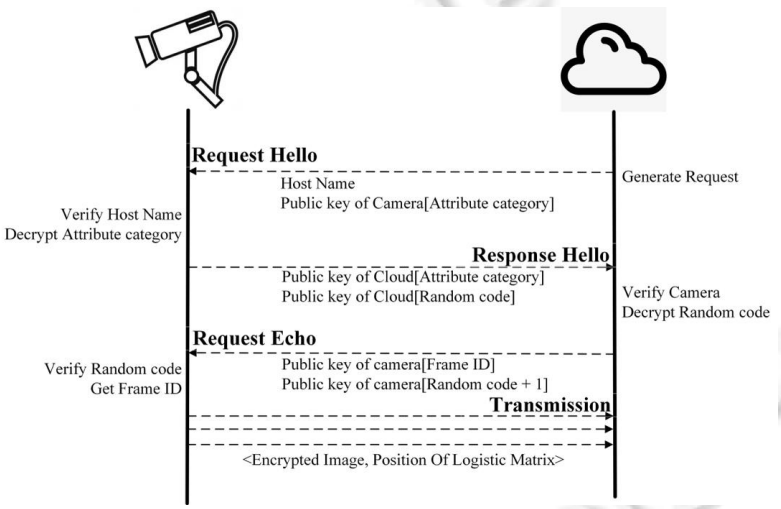

图 2 基于非对称加密的授权认证流程

整个授权机制分为 3 个部分, 分别是请求阶段、响应阶段、传输阶段.

(1) 在请求阶段, 云端生成请求上下文 *C*, 其携带主机名称 *H* 和使用摄像头端公钥加密的物体类别信息, 记为 *Public Key of Camera*[*Attribute category*], 在摄像头等边缘设备中, 有预定义的一组可信云端注册列表, 摄像头接收到上下文信息后, 首先查找注册列表, 使用本身的私钥对公钥加密过的类别信息进行解密: *Private Key of Camera*[*Public Key of Camera*[*Attribute category*]], 然后发送给云端公钥加密过的一段随机序列码 *Public Key of Cloud*[*Random Code*+1].

(2) 在响应阶段, 云端使用私钥对随机序列码进行解密之后, 将 *Random Code*+1, 并再次使用云端私钥对其进行加密: *Private Key of Cloud*[*Random Code*+1], 同时, 将帧序列索引集合使用摄像头端公钥加密后返回. 响应结果传递到摄像头端后, 摄像头分别使用云端公钥和自身私钥解密随机序列和帧索引集合信息来验证随机值的正确性.

(3) 当上述两个阶段无任何异常之后, 进入传输阶段, 此时通信双方都将对方作为可信通信方, 摄像头会将局部加密之后的图像和每个图像携带的 logistic 矩阵初始位置坐标传输给云端, 完成整个授权认证过程.

在用户授权认证和传输的整个过程中, 云端的执行代码均运行于 TEE 的安全飞地(secure enclaves)中. 安全飞地基于 IntelSGX 处理器结构, 通过扩展原有指令集和内存访问模式, 保护运行在飞地内部的代码和内存数据. 云端的处理流程分为两个阶段, 分别为传输阶段和计算阶段, 整体架构如图 3 所示.

(1) 在传输阶段，授权机制的请求响应等相关代码全部保存在安全飞地中，在调用过程中，保持和外部用户级的代码和数据的安全隔离. 相比于普通的执行环境，安全飞地拥有更高的执行权限，在安全飞地中的指令能够访问所有资源；反之，非安全飞地的指令则不能访问此区域的资源，因此传输过程的密钥不会发生泄漏. 当局部加密图像和混沌矩阵传输到云端后，局部加密图像可存储在磁盘中，而混沌矩阵和混沌矩阵初始位置信息存储在飞地中，外部代码无法获得解密序列，因此能够很好地保证用户图像隐私安全性.

(2) 在计算阶段，首先将模型载入 GPU 中；随后，将图像载入到飞地内存区域中，使用快速混沌解密方法进行解密；最后输入到模型中，得到计算结果. 整个计算过程无法通过内存或磁盘窃取用户隐私数据.

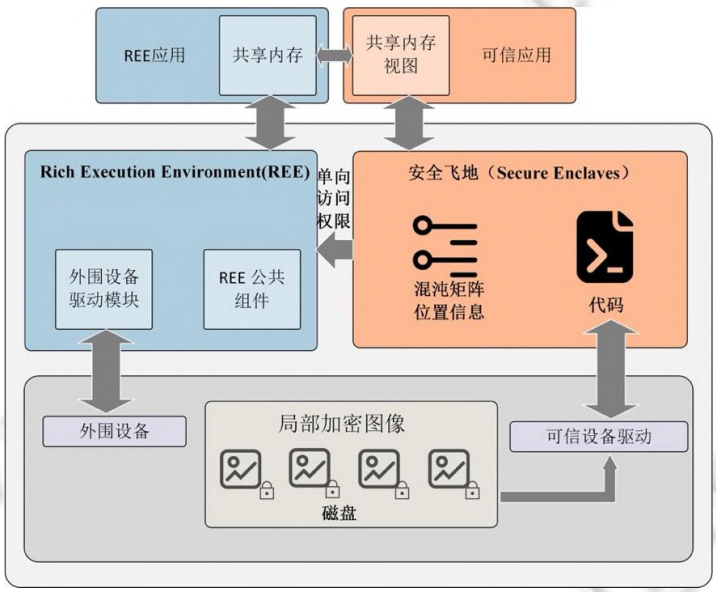

图 3 云端 TEE 架构图

利用 TEE 技术，将授权认证过程、解密流程和模型加载流程作为受信任代码放入安全飞地内部，保证云端模型运行在可信环境下. 由于 TEE 所在区域和操作系统分离，并从硬件指令上保证了计算的私密性，因此可以兼顾计算效率和安全性.

2.5 宽容教师训练策略

为了充分运用边缘端算力，需要在摄像头等 IoT 设备上部署轻量级模型. 但是轻量级模型的运行也需要保证较高的召回率，以确保在边缘端不遗漏任何候选帧. 传统的蒸馏模型通常需要花费大量的时间以及大量的数据来训练学生模型，以确保学生模型获取到足够的经验知识. 针对本模型边缘端过滤不需要极高的准确率的特点，我们提出了更加灵活的宽容教师训练策略，能够大大提高学生模型蒸馏训练时间及其 top-k 召回率，如图 4 所示.

假设 $\{x_i,y_i\}_{i=1}^N$ 表示 N 个训练样本，其中，x_i 表示第 i 个样本实例，y_i 表示其标签. 首先需要训练得到教师模型，定义教师模型的训练目标为

$$\theta_t=\arg\min_{\theta}\sum_{i\in[N]}L_{CE}^T(x_i,y_i;[\hat{\theta}_t,W]) \tag{9}$$

其中，L_{CE}^T 表示教师模型的交叉熵损失，θ_t 和 W 表示模型参数，$[N]=\{1,2,\ldots,N\}$ 表示样本全集. 给定一个样本数据图像 x_i，教师模型对其进行编码得到特征向量 $v_i=f(x_i;\theta_t)\in\mathbb{R}^d$，然后使用 softmax 层对其进行分类:

$$\hat{y}_i = P^T(y_i \mid x_i) = softmax\left(\frac{Wv_i}{T}\right) \tag{10}$$

其中, P^T表示教师模型预测的概率分布. T为温度系数, 当T=1时, 表示模型原始概率分布; 当T>1时, 概率分布被“软化”, 学生能够通过软化的标签信息学习模仿教师模型的行为.

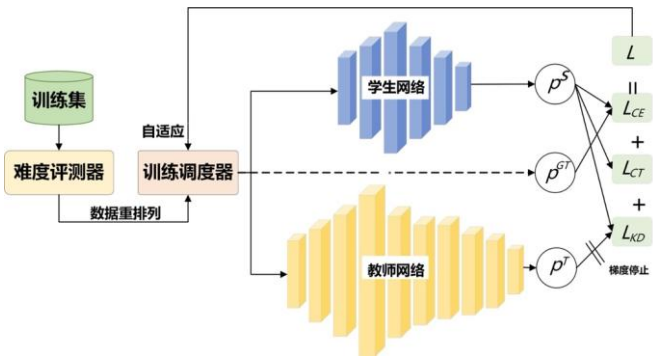

图 4 宽容教师训练策略

我们希望学生模型能通过软化的标签信息学习到教师模型的概率分布, 因此定义学生模型的蒸馏损失:

$$L_{KD} = -\sum_{i\in[N]}\sum_{c\in C}[P^T(y_i = c \mid x_i;\theta_t)\cdot \log P^S(y_i = c \mid x_i;\theta_s)] \tag{11}$$

其中, $P^S(\cdot|\cdot)$表示学生模型预测的概率分布, θ_s为学生模型参数, c为分类标签, C为分类标签集合. 通过蒸馏损失学生模型, 能够拟合教师模型的概率分布. 除了使学生模型模仿教师模型的行为, 我们必须在具体的任务上微调学生模型, 即定义交叉熵损失来规范学生模型的任务导向:

$$L_{CE} = -\sum_{i\in[N]}\sum_{c\in C}\mathbb{1}[y_i = c]\cdot \log P^S(y_i = c \mid x_i;\theta_s) \tag{12}$$

除了需要在任务导向和教师知识导向上给予学生模型指导外, 还需要考虑噪声问题. 蒸馏过程的噪声问题主要有两个: 一方面是数据集标签本身具有少量误差; 另一方面是教师模型的错误预测带来的噪声, 这样的错误往往表现在教师模型预测的top-1概率置信度较低的情况下. 对于这样的模糊样本, 其真实标签往往包含在top-k预测中. 因此, 考虑到模型容量差异、边缘端任务要求, 在学生模型中, 我们并不需要其对每个样本的正确类别都有极高的置信度, 因此我们设计了CT损失函数:

$$L_{CT} = \left(g_1 - \frac{1}{K-1}\sum_{k=2}^{K} g_k\right)^2 \tag{13}$$

其中, g_i表示学生模型预测概率输出的第i大元素, K为超参数. 损失函数约束学生模型概率分布集中在top-K附近. 可以观察到: 当i>K时, g_i≈0. 这说明当K值大时, 模型容忍度越高, 抗噪能力越高. CT损失函数在一定程度上拉近了分类结果中top-1和其余top-k概率之和的距离, 结合本节的课程学习方法的难度评估机制, 可以对模型做出正则化贡献, 是专门为课程学习设置的辅助正则损失函数, 其他损失函数无法很好地和宽容教师训练策略的难度评估函数联合起来, 从概率分布的平滑程度上对模型收敛速度提供帮助, 甚至会降低模型收敛速度. 总的损失函数为

$$L=\alpha L_{CE}+(1-\alpha)L_{KD}+\beta L_{CT} \tag{14}$$

其中, α和β为超参数, 分别用来平衡模型预测层任务导向程度和模型容忍度.

从总体上看, 蒸馏采用的是软标签损失和交叉熵损失函数联合CT损失函数的方式, 其中, 软标签作为教师模型知识的“放大器”, 能够传递给学生模型更多知识. CT损失函数用于配合宽容教师训练策略提供收敛加速和正则化, 而交叉熵损失用于衡量教师模型和学生模型概率分布的差异. 除交叉熵损失来衡量分布之间的差异成都外, 也可以使用KL散度损失函数或其他衡量分布差异的损失, 其在实际使用效果和交叉熵相似, 可以作为交叉熵损失的替代.

对于学生模型，我们认为，在训练的初始阶段，学生模型没有对知识的抽象能力，这个时候对学习资料难度的刻画主要来源于教师模型，因此，我们首先利用教师模型对知识的归纳能力对数据集难易程度进行划分，给学生模型一个启发式的数据排列信息. 具体地，使用教师模型的分类结果，按照其分类结果的概率分布对数据进行重排列. 一般来说，对某个样本的分类结果置信度越高，说明模型对此样本有更高的辨识度，此样本就会被归结到简单样本，反映在分类向量中是此向量元素峰值更高，且峰值和次峰值之差更大. 我们在难度评测器中定义难度系数(coefficient of difficulty, CoD):

$$CoD=\frac{1}{\sum_{i=2}^{n}(p_{i-1}-p_i)^2} \tag{15}$$

其中, p_i 代表分类结果第 i 大的元素; n 为超参数，出于计算量考虑，在实验中设置 n=3.

难度系数越高，说明模型更容易出错，学习顺序会更靠后. 将样本通过难度系数从小到大划分后，我们定义训练调度器. 传统的课程学习方法通常采用线性输入的方式，即使用由易到难的顺序将样本输入到蒸馏模型中，这样会导致简单样本被学习很多次. 不同于传统方法，我们提出滑动抽样方法和自适应难度调节机制，如算法 2 所描述，使得课程学习速度更快，准确度更高.

算法 2. 宽容教师训练策略.

输入: x: 输入样本; y: 样本标签; g: 学生模型; f: 教师模型; K: 容忍度下限.

输出: g': 蒸馏后的学生模型.

1. //步骤 1: 根据教师模型先验重排数据
2. $p^t=f(x;\theta_t)$; //计算教师模型对样本的概率分布
3. $CoD\_t=\left(\sum_{i=2}^{n}(p_{i-1}^t-p_i^t)^2\right)^{-1}$ //根据公式(15), 计算每个样本难度系数
4. $x\rightarrow[x_1,x_2,\ldots,x_n]$; //根据 CoD_t 将样本 x 按照从易到难分为 n 份
5. //步骤 2: 滑动采样和自适应调节
6. **While** g' not converge **do**
7. **for** ($epoch$=1 to M) **do**
8. $upperbound=\lceil n/epoch\rceil$ //n 为难度区间个数, 计算样本采样上界
9. $batch_x\sim U(x_1,x_{upperbound})$ //在难度范围内均匀采样
10. $\theta_s\leftarrow\nabla_\theta L(g(batch_x;\theta_s),f(batch_x;\theta_t),y)$; //梯度下降
11. **end for**
12. k=max(k−1,K); //容忍度收缩
13. $L_{CT}\leftarrow k$; //将置信容忍度函数 k 值更新
14. $g'\leftarrow g$; //更新模型
15. $p^s=g'(x;\theta_s)$; //计算学生模型对样本概率分布
16. $CoD\_s=\left(\sum_{i=2}^{n}(p_{i-1}^s-p_i^s)^2\right)^{-1}$ //更新学生自适应难度
17. $x\rightarrow[x_1,x_2,\ldots,x_n]$; //根据 CoD_s, 将样本 x 按照从易到难分为 n 份
18. **end while**
19. **return** g'

滑动抽样方法是为了解决课程学习的“遗忘”特点，如果我们直接将样本根据难度排序输入，那么训练到后期模型会倾向于对困难样本的特征建模，“遗忘”简单样本的特征建模方式，导致模型陷入局部最优解. 因此，根据训练阶段不同，我们将简单样本和困难样本设置不同比例的采样概率，保证在一个 batch 中含有不同难易程度的样本. 在实验中，我们将样本均匀划分到 5 个区间，分别为易、较易、中等、较难、难. 在训练初始阶段，从简单样本开始采样，逐渐扩大难度上界进行采样. 随着训练时间增长，逐渐变为均匀采样.

此外, 另一个重要的部分是容忍度收缩策略. 在训练过程中, 在训练开始阶段定义一个较高的“容忍度”, 这样使得模型能够快速收敛; 随后, 我们逐渐降低“容忍度”, 提升模型准确率, 直到到达我们设定“容忍度”下限. 和人类学习过程类似, 我们在学习初期应该有更多的容错率, 掌握了知识之后, 就需要降低容忍度来提高准确率, 直到容忍度降低到阈值, 即设置的容忍度下限. 此时, 置信度容忍损失起到正则化作用, 防止对困难样本的学习频率增大后陷入局部过拟合, 从而提高测试集泛化能力.

另一方面, 我们认为: 学生模型和教师模型在容量上的差异, 特别是对教师模型和学生模型骨干网络(backbone)不同的情况下, 两者对输入样本的建模方式和注意力会不同, 在一定程度上导致两者对待同一个数据样本有不同的难度判断. 因此, 需要在训练过程中对样本难度进行自适应调节, 执行到指定 *epoch* 后, 学生模型会根据当前状态对样本重新进行难度系数评估并重排列, 保证合理的难度输入.

在教师模型的训练过程中, 我们使用了常见的正则化如 Dropout 等, 一般的正则化方法的目的是使得模型在测试集取得更高的精确度, 而在知识蒸馏的过程中, 使用 CT 损失函数和容忍度收缩策略代替正则化的主要目的是保证在边缘端的小模型具有极高的 top-k 召回率, 即保证真实类别在最高的 k 个概率输出中即可, 仅仅依靠普通的正则化无法达到这种效果.

3 实验结果与分析

3.1 实验数据集

对于大规模视频监控检索任务, 我们使用 BrnoCompSpeed 数据集和 CoraReefLong 数据集, 其中, BrnoCompSpeed 数据集包含 21 个全高清视频, 每个视频长约 1 h, 在 6 个不同位置捕获, 车流密度分布比较均匀, 见表 1; CoraReefLong 数据集属于夜间交通监控视频, 存在大量的冗余帧.

表 1 BrnoCompSpeed 车流统计

	left	center	right
Session 1	854	848	849
Session 2	1 163	1 258	1 583
Session 3	193	193	193
Session 4	1 188	1 192	1 177
Session 5	2 021	2 027	2 030
Session 6	1 358	1 353	1 358
Total		20 865	

为了验证知识蒸馏策略的有效性, 我们使用 Stanford Cars 数据集和 Comprehensive Cars 数据集. Stanford Cars 数据集由 196 类汽车组成, 共有 16 185 张图像, 包含 8 144 个训练图像和 8 041 个测试图像. CompCars 数据集包含来自两个场景的数据, 包括来自网络自然和监视捕捉的图像, 监控数据为从车辆前方拍摄的 50 000 张图像. 每个车型都标有 5 个属性, 包括最大速度、排量、门数、座位数和汽车类型.

3.2 实验设置

两阶段过滤模型分别在边缘端和服务端部署, 其中, 服务端使用带有 Tesla V100 的 GPU 服务器, 部署重量级的教师模型和向量检索库, 边缘端使用带有 CPU 算力的摄像头部署蒸馏模型.

在检索过程中, 通常有两种负载耗时: 一种是输入时负载(ingest cost, IC), 也就是在输入过程中使用模型进行分析; 一种是查询时负载(query cost, QC), 在查询阶段才用模型进行分析. 因此, 我们定义两种基线模型(baseline).

(1) 当只在输入使用重量级教师网络过滤时, 我们称为 IC-all. 此时, 我们仅在摄像头部署重量级模型, 并在视频流输入的同时进行精确分类, 查询需要等到所有视频帧被处理完成才能进行检索.

(2) 当只在查询时使用重量级网络进行过滤时, 我们称为 QC-all. 此时, 我们仅在云端部署重量级模型, 在每次查询时, 都需要完整地对所有视频帧进行精确分类和检索.

在边缘端, 我们训练了一个轻量级的二分类网络, 用来过滤冗余帧. 结果显示, 在多个数据集上, 平均冗

余帧过滤率在 20%左右，在 CoraReefLong 数据集中超过 80%. 对于冗余帧过滤，我们在实验过程中采用两种策略来应对不同的监控场景.

(1) 在视频流中对每帧做二分类检测，判断是否包含待检测目标. 例如夜间的交通监控应用场景，在大部分时间通常不包含我们感兴趣的人或者车辆，此时可以使用此策略来过滤大量帧.

(2) 在视频流中设置特定的时间间隔，对视频流进行采样. 例如在交通拥堵路段，根据视频的时间局部性，在很短时间内视频流中的物体基本不发生变化，因此对一个 50–80 f/s 的视频流，可以根据最高车速限制设置采样间隔 0.2 s–1 s 左右.

我们在 BrnoCompSpeed 数据集中采用策略 2, CoraReefLong 数据集中采用策略 1.

边缘端目标检测模型，我们使用轻量级的 YOLOV4-tiny 模型，蒸馏的属性分类网络我们分别使用 ResNet 和 VGG 作为模型骨干网络. 对于教师网络，我们使用在 Stanford Cars 和 Comprehensive Cars 数据集上训练完成的 ResNet-151 网络模型，将其分类结果作为 ground truth 标签，向量检索使用 faiss 检索引擎. 云端使用支持 SGX 功能的 Xeon 系列 CPU, CentOS 操作系统，同时配置 16 GB 普通内存, 128 MB 安全内存, 2 GB 网络带宽.

3.3 实验结果

本文实验分为 4 个部分：首先论证我们的隐私保护策略安全性，比较加密算法的性能；其次，我们将监控数据集上提出的 TFM 方法和传统方法进行对比；再者，使用消融实验验证提出的蒸馏损失和策略的有效性；最后，对知识蒸馏中涉及的超参数进行分析讨论.

我们应用基于 logistic 的加密方法对图像进行局部加密，这种加密方法能够在计算能力受限的情况下快速加密图像. 我们测试了几种图像加密算法在 Inteli7-4720HQ 硬件环境下的速度对比，见表 2.

表 2 几个加密算法速度对比(单位：秒/个)

加密方法/数量	10	100	1 000
HE	>>1s	>>1s	>>1s
Arnold@2	0.062 9	0.051 8	0.044 9
Arnold@1	**0.029 9**	0.028 7	0.022 9
OriginLogistic	0.081 9	0.060 4	0.058 9
Ours	0.081 3	**0.014 1**	**0.009 1**

其中, HE 为同态加密算法, Arnold@K 意为进行 K 次置换的 Arnold 算法, OriginLogistic 指在每次加密都生成 Logistic 混沌序列. 实验结果显示，在大数据量、长时间运行的环境下，我们的加密算法速度显著高于其他算法.这是由于我们的算法只需生成一次全局混沌矩阵，减少了大量计算力.

使用改进后的 logistic 混沌序列函数进行加密，是基于效率的考虑. 从边缘算力规模来看，更加复杂的加密算法势必会进一步增加计算负担，导致边缘端视频流输入处理变慢，出现视频流输入速度大于边缘端处理速度，长时间积累后会导致模型整体效率不足. 例如，若采用 Arnold 加密算法，在边缘端的图像无法被实时处理，长时间累积，图像处理时延会越来越大. 当模型运行相当一段长时间之后，大量的滞后加密会让检索时间达到无法忍受的程度. 摄像头内部的轻量级加密算法为敏感图像提供了存储安全，结合用户授权验证机制为图像提供传输安全，在云端使用 TEE 技术可以保证计算安全，这样一来，实现了全方位的用户隐私保护.

据我们了解，现有研究没有将边缘端和服务端结合的工作，并且对于视频检索往往只局限于检索包含特定类别的帧，对于其细粒度检索要求完全无法满足. 如图 5 所示，实验中，我们使用 3 种 TFM 模型，其中，TFM-q 表示以检索阶段时延为高优先级，即使用较为严格的容忍度蒸馏一个准确率较高的学生模型，使之在输入时对每个物体尽可能精确分类，与此同时，会增加一些输入负载消耗；相反地，TFM-i 表示以输入阶段损耗为高优先级，即使用较为宽松的容忍度蒸馏一个准确率较低的学生模型,使之在输入阶段有更小的计算量，与此同时，会产生比 TFM-q 更多的候选实体. 在查询阶段，需要教师模型筛选，因此查询负载较大. 为了平衡两种方案，我们折中得到 TFM-b 方案，为上述两种方案的折中. 实验结果显示，相比于一阶段检索的 baseline, TFM 模型在保证准确率达到 90%以上的前提下，大幅度地降低了两种负载均衡.

在 BrnoCompSpeed 数据集上，我们对比了基线模型和 TFM 的检索效率，如图 6 所示. 结果显示，在 6 个

道路监控中，在全部召回率保证在92%以上的情况下，相对于IC-all，平均加速9倍；相对于QC-all，平均加速133倍. 这显示我们的TFM模型具有低延时、高准确度的特点.

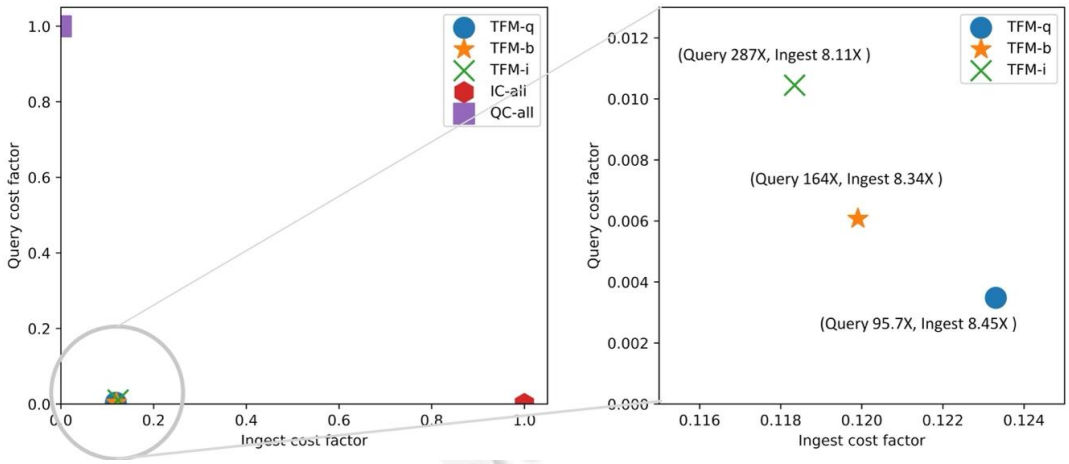

图 5 CoraReefLong 数据集下, TFM 系列模型与 IC-all 和 QC-all 对比

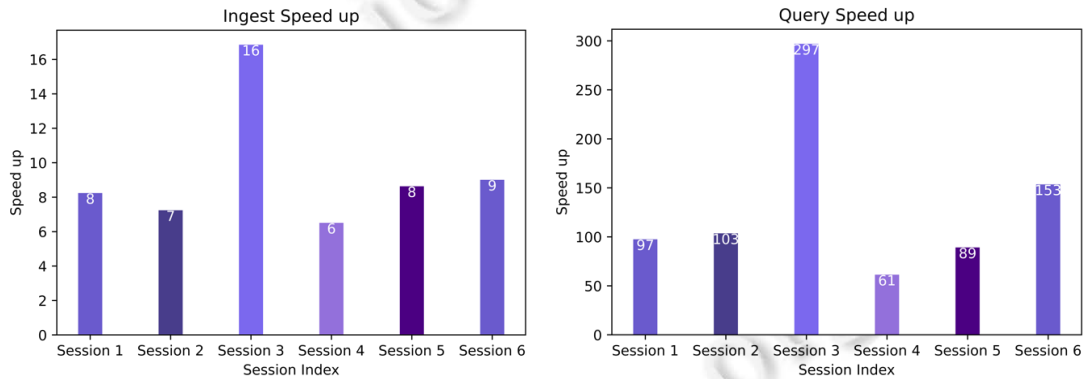

图 6 在 BrnoCompSpeed 数据集下, TFM 相对 IC-all 和 QC-all 的加速

相对于输入负载和查询负载的单向优化，在实际应用中，我们倾向于寻找一个平衡点，能够综合两方面的优势. 在TFM模型中，主要体现在对学生模型细粒度分类结果取前 K 个结果的决策上，对 K 的参数设置将在后面讨论.

除了在输入负载和查询负载上的加速以外，蒸馏后的模型在边缘端占据了更小的存储空间，见表3，蒸馏后的模型只占原教师模型的 12%左右，我们在保证压缩比例高达 88%的情况下，依然保证较高的准确率，相比于直接训练小模型，其准确率提升了4%.

表 3 Stanford Cars 数据集下，模型压缩规模对比

Model	Speed up	Model size	*P*@1
ResNet151 (teacher)	1×	232 M	100 (ground truth)
ResNet101	1.35×	173 M (74%)	98
VGG-middle	5.6×	41 M (18%)	95
ResNet-small	8.6×	27 M (12%)	88
KD-NN	**8.6**×	**27 M (12%)**	**92**

为了探索在定制化的学生属性分类网络训练过程中宽容教师训练策略的有效性，我们进行了一系列消融实验. 如图 7 所示，我们比较了蒸馏模型(KD-NN)和直接进行训练的学生模型(None-KD-NN)的模型在Stanford Cars数据集下训练的收敛度. 从左图可以看出，使用TTS训练策略在40个epoch之后已经趋于稳定，

而原始训练方法的训练集需要训练到 70 个 epoch 才逐渐趋于稳定. 从右图观察到, 在一般的训练过程中, 到了后期损失函数会减小到非常低, 模型会拟合, 但是 TTS 训练策略的损失函数具有一定的正则化效果, 使得相对于一般的训练过程更不容易陷入局部最优解.

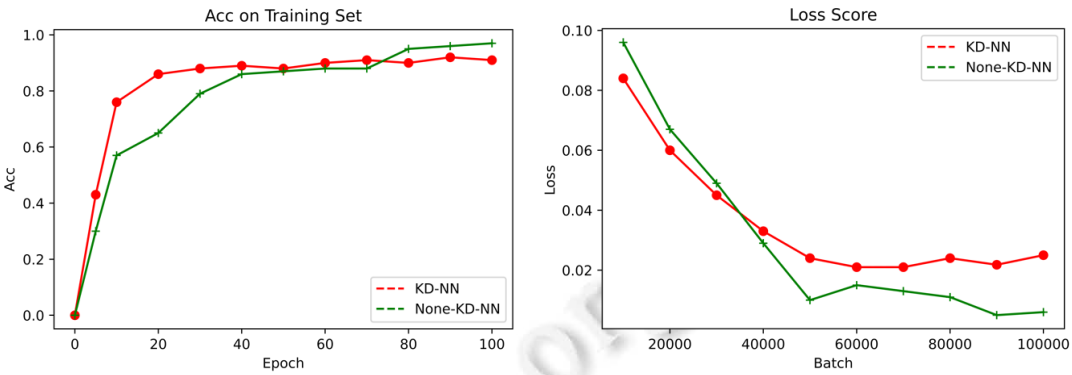

图 7 Stanford Cars 数据集下, TTS 和普通训练策略对比

表 4 进一步证实了我们的想法, 我们可以发现, TTS 蒸馏模型的损失函数具有正则化效果, 即训练集和测试集的准确率和召回率成反相关关系. 这表明 TTS 有较强的泛化能力, 完全符合我们的预期假设.

表 4 Stanford Cars 数据集下, TTS 和普通训练策略训练集测试集准确率/召回率对比(%)

Method	*P* (training set)	*R* (training set)	*P* (test set)	*R* (test set)
None-KD-NN	**97.5**	**95.2**	90.1	89.9
KD-NN	95.7	93.3	**93.6**	**93.0**

为了更好地验证 CT 损失函数和 TTS 训练策略的有效性, 我们基于 Comprehensive Cars 数据集分别对 KD Loss、CT Loss 和 TTS 进行了消融实验, 见表 5.

表 5 Comprehensive Cars 数据集下, KD-NN 消融实验(%)

Method	*P*@1	*P*@5	*P*@10	MRR
KD-NN	**87.9**	**92.3**	**93.7**	**94.4**
w/o KD Loss	86.8 (↓1.1)	91.3 (↓1.0)	93.4 (↓0.3)	94.1 (↓0.3)
w/o CT Loss	87.3 (↓0.6)	91.2 (↓1.1)	93.4 (↓0.3)	94.2 (↓0.2)
w/o TTS	87.7 (↓0.2)	92.0 (↓0.3)	93.6 (↓0.1)	94.4 (↓0.0)
None-KD-NN	85.5 (↓2.4)	90.3 (↓2.3)	93.1 (↓0.6)	93.9 (↓0.5)

其中, *P*@1 表示 Top-1 准确率, 其余同理, MRR 表示应用此方式在最后的视频检索上的平均倒数指标 (mean reciprocal rank). 结果显示, CT 损失函数在训练过程中起到了不可或缺的作用, 最高会有 1.1%的性能提升, 而 TTS 对准确率也起到了一定的提升作用. 总的来说, 前者对模型准确率、正则化有重大影响, 后者对知识蒸馏过程的加速、收敛起到巨大作用.

在 CT 损失函数中, 设置不同的超参数 *K* 值会对模型产生影响, 见表 6.

从实验结果来看, 在 Comprehensive Cars 数据集上, 不同的 top-*N* 准确率的最优 *K* 值是不同的. 当我们需要使模型的 top-*N* 准确率提高时, 实验证明, 设置 *K* 略大于 *N* 值最佳.

表 6 Comprehensive Cars 数据集上, CT 损失函数的超参数实验(%)

Value	*P*@1	*P*@5	*P*@10
K=5	**92.4**	93.1	95.4
K=10	91.8	**94.0**	**95.8**
K=15	91.6	93.5	94.8
K=20	90.1	91.2	94.5

4 结 论

大规模监控视频检索任务拥有广泛的应用场景, 现有模型大部分基于服务端输入构建视频分析和检索系统, 不能满足保护用户隐私、检索准确率高、延时低的要求. 在边缘计算蓬勃发展的背景下, 我们提出了结合边缘算力和服务端算力的两阶段过滤模型 TFM. 模型使用结合了 CT 损失函数、课程学习的宽容教师训练策略对大模型进行知识蒸馏, 并将生成的小模型部署在边缘端. 一方面, 训练策略大幅度加快模型收敛速度, 减少了训练时间, 使知识蒸馏后的模型在保证高可用性的前提下大大压缩其参数量, 能够在低资源环境下高效运行; 另一方面, 利用边缘端实时粗粒度过滤, 检索阶段细粒度属性识别的方法, 大大降低了输入负载和查询负载. 同时, 在用户隐私保护上, 针对边缘端计算资源有限的情况, 我们提出了快速局部加密结合用户授权、云端 TEE 技术的隐私保护方法, 在极低的资源下, 保证了隐私安全. 在实际应用中, 提供了一种安全、高效的视频检索模型.

References:

[1] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2016. 770−778.

[2] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2016. 779−788.

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc. of the 3rd Int'l Conf. on Learning Representations (ICLR 2015). 2015. 1−14.

[4] Jia Z, Maggioni M, Staiger B, Scarpazza DP. Dissecting the nvidia Volta GPU architecture via microbenchmarking. arXiv:1804.06826, 2018.

[5] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), Vol.1. 2019. 4171−4186.

[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the Advances in Neural Information Processing Systems. 2017. 5998−6008.

[7] Zhou D, Frémont V, Quost B, Dai Y, Li H. Moving object detection and segmentation in urban environments fro m a moving platform. Image and Vision Computing, 2017, 68: 76−87.

[8] Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 2016, 23(10): 1499−1503.

[9] Dufaux F, Ebrahimi T. A framework for the validation of privacy protection solutions in video surveillance. In: Proc. of the IEEE Int'l Conf. on Multimedia and Expo (ICME 2010). 2010. 66−71.

[10] Upmanyu M, Namboodiri AM, Srinathan K, Jawahar CV. Efficient privacy preserving video surveillance. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2009. 1639−1646.

[11] Ahn J, Shim HJ, Jeon B, *et al*. Digital video scrambling method using intra prediction mode. In: Advances in Multimedia Information Processing, 2005. 386−393.

[12] Liu Z, Li X. Motion vector encryption in multimedia streaming. In: Proc. of the 10th Int'l Multimedia Modelling Conf. (MMM 2004). 2004. 64−71.

[13] Zhou J, Liang Z, Chen Y, Au OC. Security analysis of multimedia encryption schemes based on multiple Huffman table. IEEE Signal Processing Letters, 2007, 14(3): 201−204.

[14] Zhang W, Cheung SCS, Chen M. Hiding privacy information in video surveillance system. In: Proc. of the Int'l Conf. on Image Processing (ICIP), Vol.3. 2005. II−868.

[15] Park J, Kim DS, Lim H. Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures. IEEE Access, 2020, 8: 203564−203579.

[16] Liu J, Tian Y, Zhou Y, Xiao Y, Ansari N. Privacy preserving distributed data mining based on secure multi-party computation. Computer Communications, 2020, 153: 208−216.

[17] Hunt T, Zhu Z, Xu Y, Peter S, Witchel E. Ryoan: A distributed sandbox for untrusted computation on secret data. ACM Trans. on Computer Systems, 2018, 35(4): 1−32.

[18] Baumann A, Peinado M, Hunt G. Shielding applications from an untrusted cloud with haven. ACM Trans. on Computer Systems, 2015, 33(3): 1−26.

[19] Bengio Y, Louradour J, Collobert R, Weston J. Curriculum learning. In: Proc. of the 26th Annual Int'l Conf. on Machine Learning. 2009. 41−48.

[20] Guo S, Huang W, Zhang H, *et al*. CurriculumNet: Weakly supervised learning from large-scale Web images. In: Proc. of the European Conf. on Computer Vision (ECCV). 2018. 135−150.

[21] Jiang L, Meng D, Mitamura T, Hauptmann AG. Easy samples first: Self-paced reranking for zero-example multimedia search. In: Proc. of the ACM Conf. on Multimedia (MM 2014). 2014. 547−556.

[22] Platanios EA, Stretcu O, Neubig G, Poczos B, Mitchell TM. Competence-based curriculum learning for neural machine translation. In: Proc. of the Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), Vol.1. 2019. 1162−1172.

[23] Tay Y, Wang S, Tuan LA, Fu J, Phan MC, Yuan X, Rao J, Hui SC, Zhang A. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019). 2020. 4922−4931.

[24] El-Bouri R, Eyre D, Watkinson P, Zhu T, Clifton DA. Student-teacher curriculum learning via reinforcement learning: Predicting hospital inpatient admission location. In: Proc. of the 37th Int'l Conf. on Machine Learning (ICML 2020). 2020. 2848−2857.

[25] Florensa C, Held D, Wulfmeier M, Zhang M, Abbeel P. Reverse curriculum generation for reinforcement learning. In: Proc. of the Conf. on Robot Learning. 2017. 482−495.

[26] Narvekar S, Sinapov J, Stone P. Autonomous task sequencing for customized curriculum design in reinforcement learning. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2017. 2536−2542.

[27] Qu M, Tang J, Han J. Curriculum learning for heterogeneous star network embedding via deep reinforcement learning. In: Proc. of the 11th ACM Int'l Conf. on Web Search and Data Mining (WSDM 2018). 2018. 468−476.

[28] Gong C, Yang J, Tao D. Multi-modal curriculum learning over graphs. ACM Trans. on Intelligent Systems and Technology, 2019, 10(4): 1−25.

[29] Guo Y, Chen Y, Zheng Y, Zhao P, Chen J, Huang J, Tan M. Breaking the curse of space explosion: Towards efficient NAS with curriculum search. In: Proc. of the Int'l Conf. on Machine Learning. 2020. 3822−3831.

[30] Hinton G, Vinyals O, Dean J, *et al*. Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.

[31] Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y. FitNets: Hints for thin deep nets. In: Proc. of the 3rd Int'l Conf. on Learning Representations (ICLR 2015), Vol.2. 2015. 1−13.

[32] Zagoruyko S, Komodakis N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: Proc. of the 5th Int'l Conf. on Learning Representations (ICLR 2017). 2017. 1−13.

[33] Kim J, Park Y, Kim G, Hwang SJ. SplitNet: Learning to semantically split deep networks for parameter reduction and model parallelization. In: Proc. of the 34th Int'l Conf. on Machine Learning (ICML 2017), Vol.4. 2017. 1866−1874.

[34] Lowe DG. Object recognition from local scale-invariant features. In: Proc. of the IEEE Int'l Conf. on Computer Vision, Vol.2. 1999. 1150−1157.

[35] Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR 2005), Vol.1. 2005. 886−893.

[36] Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1627−1645.

[37] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. 2014. 580−587.

[38] Girshick R. Fast R-CNN. In: Proc. of the IEEE Int'l Conf. on Computer Vision. 2015. 1440−1448.

[39] Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes challenge: A retrospective. Int'l Journal of Computer Vision, 2015, 111(1): 98−136.

[40] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137−1149.

[41] He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2020, 42(2): 386−397.

[42] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. SSD: Single shot multibox detector. In: Proc. of the European Conf. on Computer Vision (ECCV 2016). 2016. 21−37.

[43] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv:1804.02767, 2018.

[44] Bochkovskiy A, Wang CY, Liao HYM. Yolov4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020.

[45] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc. of the IEEE, 1998, 86(11): 2278−2324.

[46] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 2017, 60(6): 84−90.

[47] Elsken T, Metzen JH, Hutter F. Neural architecture search: A survey. Journal of Machine Learning Research, 2019, 20(1): 1997−2017.

[48] Johnson J, Douze M, Jegou H. Billion-scale similarity search with GPUs. IEEE Trans. on Big Data, 2021, 7(3): 535−547.

[49] Li E, Zhou Z, Chen X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In: Proc. of the Workshop on Mobile Edge Communications (MECOMM 2018). 2018. 31−36.

[50] Grulich PM, Nawab F. Collaborative edge and cloud neural networks for real-time video processing. Proc. of the VLDB Endowment, 2018, 11(12): 2046−2049.

[51] Hsieh K, Ananthanarayanan G, Bodik P, Venkataraman S, Bahl P, Philipose M, Gibbons PB, Mutlu O. Focus: Querying large video datasets with low latency and low cost. In: Proc. of the 13th USENIX Symp. on Operating Systems Design and Implementation (OSDI 2018). 2018. 269−286.

[52] Kang D, Emmons J, Abuzaid F, Bailis P, Zaharia M. NoScope: Optimizing deep CNN-based queries over video streams at scale. Proc. of the VLDB Endowment, 2017, 10(11): 1586−1597.

覃浩(1998－), 男, 硕士生, 主要研究领域为自然语言处理, 视觉语言预训练模型,模型压缩, 视频检索.

张若非(1974－), 男, 博士, 教授, 博士生导师, 主要研究领域为机器学习, 数据挖掘, 自然语言处理, 多模态内容表示和理解.

王平辉(1984－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为机器学习与数据挖掘, 自然语言处理, 移动互联网安全.

覃遵颖(1985－), 女, 高级工程师,主要研究领域为机器学习, 数据挖掘.