中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

多端交通视频分析任务卸载决策

温震宇, 胡慧峰, 钱滨, 洪榛, 俞立

引用本文:
温震宇,胡慧峰,钱滨,洪榛,俞立. 多端交通视频分析任务卸载决策[J]. 控制与决策, 2024, 39(8): 2773−2782.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2023.0241

多无人机协同直播场景下自适应任务卸载决策

Adaptive task offloading decision of multi−UAVs cooperation in live broadcasting scenario

控制与决策. 2021, 36(4): 974−982 https://doi.org/10.13195/j.kzyjc.2019.1104

一种无人船动力定位跨平台实时控制模型

Real−time cross−platform control system for unmanned ship dynamic positioning

控制与决策. 2021, 36(4): 909−916 https://doi.org/10.13195/j.kzyjc.2019.0960

多端交通视频分析任务卸载决策

温震宇[1], 胡慧峰[2], 钱 滨[3], 洪 榛[1†], 俞 立[2]

(1. 浙江工业大学 网络空间安全研究院，杭州 310023；2. 浙江工业大学 信息工程学院，杭州 310023；
3. 纽卡斯尔大学 计算机学院，纽卡斯尔 NE45TG)

摘 要: 针对智慧交通中多点位监控视频分析时出现的计算任务繁重、设备之间资源利用不均衡的问题,提出一种基于云控制的视频分析卸载方案. 首先,针对客户端算力不足而无法完成视频分析任务的问题,使用一种视频卸载框架,将部分视频分析任务切块卸载至云服务器处理;其次,针对服务器与多客户端之间的算力资源竞争问题,提出一种阶段优化卸载算法,平衡设备之间负荷,提高资源利用率;最后,针对不同点位的客户端需求不同的问题,在算法中加入精度和能耗偏好,满足不同客户端的需求. 与其他卸载方案对比的实验表明,所提出方案能够更好地对视频分析任务进行合理分配,提高系统收益,并通过扩展实验验证所提出系统的扩展能力.

关键词: 多端；视频分析；任务卸载；资源竞争；卸载决策；智慧交通

中图分类号: TP919.8 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2023.0241

引用格式: 温震宇, 胡慧峰, 钱滨, 等. 多端交通视频分析任务卸载决策[J]. 控制与决策, 2024, 39(8): 2773-2782.

Multi-client traffic video analysis task offloading decision

WEN Zhen-yu[1], *HU Hui-feng*[2], *QIAN Bin*[3], *HONG Zhen*[1†], *YU Li*[2]

(1. Institute of Cyberspace Security，Zhejiang University of Technology，Hangzhou 310023，China；2. College of Information Engineering，Zhejiang University of Technology，Hangzhou 310023，China；3. School of Computing，Newcastle University，Newcastle NE45TG，UK)

Abstract: To address the problem of heavy computational tasks and uneven resource utilization among devices in multi-point monitoring video analysis in intelligent transportation, a cloud-controlled video analysis offloading scheme is proposed. Firstly, for the problem of insufficient client computing power to complete video analysis tasks, a video offloading framework is used to segment and offload some of the video analysis tasks to cloud servers. Then, a stage-optimized offloading algorithm is proposed to balance the load between servers and multiple clients, and improve resource utilization. Finally, to address the issue of different client requirements at different points, precision and energy consumption preferences are added to the algorithm to meet the needs of different clients. Experimental comparisons with other offloading schemes demonstrate that this scheme can better allocate video analysis tasks, improve system benefits, and the scalability of the system is demonstrated through extended experiments.

Keywords: multi-client；video analysis；task offloading；resource competition；offloading decision；smart traffic

0 引 言

在交通领域,伴随着智能型城市的快速发展,智慧交通的建设得到了推进[1-2],尤其在交通视频分析方面[3]. 交通视频分析是智慧交通的一个核心要点,国内外广泛采用CNN(convolutional neural networks)对视频进行推理分析[4-5],但交通视频的数据量较大,小型边缘设备难以完成繁重的视频分析任务. 此外,为了便于移动部署,临时的边缘设备经常会采用电池供电,但对视频数据进行分析非常耗能[6-7],这给边缘设备的持续性工作带来阻碍. 基于上述诸多难点,众多国内外学者对视频卸载进行了深入研究.

就当前而言,在不同边缘节点部署客户端,并将客户端的部分视频卸载至服务器[8-9]进行分析是一较好的选择,如图1所示.

DeepDecision将低功耗的边缘客户端与性能强大的服务器结合,以降低边缘客户端的计算压力,但这仅限于单客户端对服务器,容易造成资源浪费[10];FastVA在边缘设备上使用了NPU(neural-network processing units)运行神经网络模型,使得边缘设备的视频分析速率得到提升,但分析精度较差[11];

收稿日期: 2023-03-03；**录用日期:** 2023-07-22.
责任编委: 郭戈.
†通讯作者. E-mail: zhong1983@zjut.edu.cn.

OsmoticGate采用分层队列模型(hierarchical queue model),改善了视频卸载系统中存在的数据拥挤问题,但容易导致卸载至服务器的视频过多,使服务器超负荷[12];采用动态DNN(deep neural network)模型选择进行视频分析可以增强系统处理能力,减少视频卸载,但容易导致客户端能耗增加[13];EdgeVison采用多客户端之间共同协作的方式,节约了系统资源,但需要每个客户端都有较强的边缘计算能力[14].以上解决方案虽然在一定程度上能较好地完成视频卸载任务,但大多数研究对象都是单客户端与服务器构成的简单卸载系统,对多客户端与服务器组成的多端系统考虑较少,并且这些方案在设计时虽然考虑了能耗与精度的平衡,或者考虑了视频卸载的延迟,但面对系统中成员数量变动时,往往不能快速自适应新状态,比如,客户端数量增加时,上述卸载方案无法满足变动的系统结构,扩展性较差.

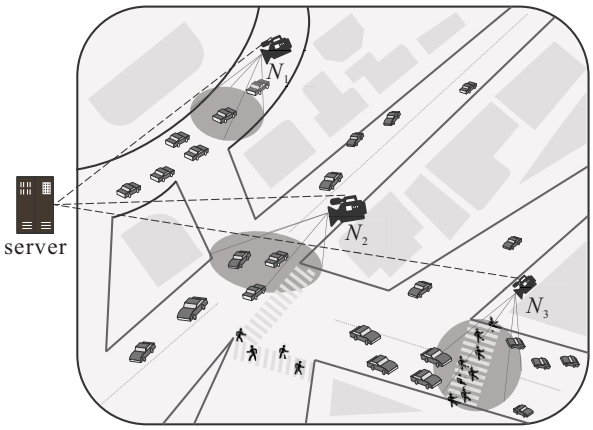


图 1 交通视频分析任务卸载

本文基于前人的研究,首先构建一个多客户端与服务器组成的云边视频卸载框架,并对框架中的输入和输出进行数学建模,量化系统收益;其次,设计基于云控制的阶段优化策略算法,用于调节卸载策略,平衡设备之间的负荷,最大化系统收益;最后,通过真实实验和仿真实验,验证本文算法的优越性.本文的主要贡献如下:

1) 针对多客户端之间的资源竞争问题,设计一种阶段优化算法,自适应地进行资源调配,提高多设备之间的资源利用;

2) 针对客户端之间需求不同的问题,在算法中加入精度和能耗的偏好选择,以满足实际场景中客户端偏好需求;

3) 针对变化的客户端数量,通过对不同数量客户端的实验,验证所提出算法对多态复杂系统的适应性及稳定性.

1 框架与建模

1.1 视频卸载框架

由多客户端与服务器组成的云边视频卸载框架如图2所示.

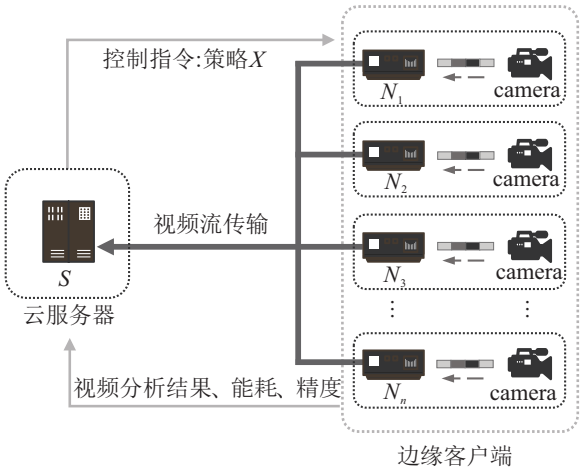


图 2 视频卸载框架

带有摄像头监控的边缘客户端N根据一定的策略将视频卸载至云服务器S或留在本地分析,本地分析完成后将结果、精度、能耗等信息一起发送给云服务器进行信息融合.云服务器会基于能耗、精度、负荷等信息对客户端的卸载策略进行调整,卸载策略包含两种决策变量r和m.其中:r决定了每个客户端卸载多少视频至服务器,称为卸载率;m决定了本地客户端采用什么CNN模型进行视频分析,称为模型选择.上述卸载框架可抽象表达为以下数学模型:

在由云服务器S和n台边缘客户端N组成的多端视频卸载系统$\xi=\{N_1,N_2,\dots,N_n,S\}$中,具有决策变量$r$和$m$,分别为

$$\begin{cases} r=[r_1,r_2,\dots,r_n]; \\ m=[m_1,m_2,\dots,m_n]. \end{cases} \tag{1}$$

其中:$r_1\sim r_n$分别表示客户端$N_1\sim$客户端N_n的卸载率;$m_1\sim m_n$分别表示客户端$N_1\sim$客户端N_n的CNN模型选择.例如,客户端N_n的策略为(r_n,m_n),则整个系统的策略X可表示为

$$X=\{(r_1,m_1),(r_2,m_2),\dots,(r_n,m_n)\}. \tag{2}$$

在本框架中,关键是要确定每个客户端的视频卸载数量.为便于定量分析,本文将连续的视频流分割为均匀的多段视频,根据视频时长,每隔D_n秒进行一次划分,D_n便称为一个边缘周期.在边缘周期D_n内,继续将视频均匀划分为若干个视频块,每个视频块的时长为t秒,则每个边缘周期D_n内共有D_n/t个视频块,若卸载率为r_n,则每个边缘周期内卸载至云服务

器的视频块数为

$$a_n(r_n)=\frac{r_nD_n}{t}, \tag{3}$$

留在边缘客户端本地的视频块数为

$$b_n(r_n)=\frac{D_n}{t}-a_n(r_n). \tag{4}$$

其中: t 的取值[12]通常为1, 即每个视频块时长为1 s.

值得注意的是, 本文采用CNN进行视频分析, CNN模型的输入是图片, 而一段连续的视频实际上是由一帧帧的离散图片组成的, 通过FFmpeg视频处理工具即可实现对视频帧的细致划分. 在对原视频进行分割时, 原视频可以被切割成包含若干帧的一小段视频块, 分割前后的图片帧总数相同, 不会对原视频的分析结果造成影响. 此外, 部分在客户端切割后的视频块可能需要传输至云服务器进行推理分析, 当下常用的传输方式有基于视频帧[15]传输或者比特率[16-17]传输. 本文采用基于比特率的传输方式, 在传输时将视频帧编码压缩, 传输完成后将其解码恢复成视频帧再输入至CNN模型进行推理分析, 通过视频编解码的方式可以大大提高数据的传输效率且对原视频的分析结果几乎不会造成影响[12].

1.2　计算模型

基于系统控制角度, 需要对系统的输入和输出进行数学建模, 并将输出结果量化成系统收益. 系统输入即每个客户端的卸载率和CNN模型选择, 可定义为 $X=(r_1,m_1),(r_2,m_2),\dots,(r_n,m_n)$; 系统输出(即设备的负荷率、精度、能耗)会被量化为系统收益, 系统收益作为反馈信息, 促使算法对卸载率和CNN模型选择做出调整, 进行卸载策略迭代.

1.2.1　负荷模型

设备的负荷率分为客户端负荷率和服务器负荷率, 负荷率被定义为设备进行视频分析工作时长占总运行时长的比例.

若客户端 n 在规定的边缘周期 D_n 内需要分析的视频块数为 b_n, 完成单个视频块分析所需的时长为 $T_{b_n}(m_n)$, 时长取决于采用的CNN模型, 则该客户端进行视频分析的工作时长为 $b_n(r_n)T_{b_n}(m_n)$, 其负荷率可表示为

$$L_n^{\mathrm{e}}(r_n,m_n)=\frac{b_n(r_n)T_{b_n}(m_n)}{D_n}. \tag{5}$$

值得注意的是, 真实环境中设备的负荷率无法超过100 %, 但本文为了便于后续计算收益, 将负荷率设定为可突破100 %限制. 当负荷率超过100 %时, 意味着有视频块由于无法及时处理而被丢弃, 交通监控信息损失较大, 因此要尽量避免设备超负荷.

云服务器需要处理来自所有客户端的视频块, 若系统中的客户端数量为 n, 在边缘周期 D_n 内, 每个客户端卸载至服务器的视频块数量为 $a_n(r_n)$, 服务器完成每个视频块分析所需要的时间为 T_{a_n}, 则服务器的负荷率为

$$L_n^{\mathrm{c}}(r_n)=\frac{\sum_{n=1}^{n}a_n(r_n)T_{a_n}}{D_n}. \tag{6}$$

1.2.2　能耗模型

边缘客户端进行视频分析的功率为 $P_n^p(m_n)$, 其大小取决于当前使用的CNN模型; 待机时的功率为 P_n^w, 这取决于设备的物理构造. 留在客户端分析的视频块数量为 $b_n(r_n)$, 每块耗时为 $T_{b_n}(m_n)$, 则该客户端在执行视频分析任务过程中消耗的能量为

$$E_n^p(r_n,m_n)=b_n(r_n)T_{b_n}(m_n)P_n^p(m_n), \tag{7}$$

待机空闲时间消耗的能量为

$$E_n^w(r_n,m_n)=[D_n-b_n(r_n)T_{b_n}(m_n)]P_n^w. \tag{8}$$

需要注意的是, 上述定义有一个限制条件, 即视频分析时长必须在边缘周期时长之内, 写为

$$b_n(r_n)T_{b_n}(m_n)\leqslant D_n. \tag{9}$$

若视频需要的分析时长超出周期长度, 则意味着客户端在整个周期 D_n 内不存在待机空闲状态, 且超出周期时长的视频块将被放弃处理. 最终, 客户端在周期 D_n 内的总能耗可表示为

$$E_n^{\mathrm{sum}}(r_n,m_n)=\begin{cases}E_n^p(r_n,m_n)+E_n^w(r_n,m_n),\ b_n(r_n)T_{b_n}(m_n)\leqslant D_n;\\ D_nP_n^p(m_n),\ b_n(r_n)T_{b_n}(m_n)>D_n.\end{cases} \tag{10}$$

1.2.3　精度模型

视频分析的精度取决于客户端的卸载率和模型选择. 本文采用3种规模不同的CNN模型, 模型1为Yolov5n, 模型2为Yolov5s, 模型3为Yolov5m, 模型的规模和精度依次增大. 模型1的精度为最小精度 A_n^1, 模型2的精度为 A_n^2, 模型3的精度为最大精度 A_n^3, 由于本文服务器固定采用模型2, 客户端选择任意模型, 根据不同的卸载率 r_n 和模型选择 m_n, 客户端 n 在周期 D_n 内可获得的平均精度为

$$A_n^{\mathrm{avg}}(r_n,m_n)=r_nA_n^2+(1-r_n)A_n^{m_n}. \tag{11}$$

1.3　收益函数

综合考虑以下系统输出: 服务器负荷率, 每个边缘客户端的负荷率、精度、能耗, 将上述输出量化为

收益后,系统总体的收益可表示为

$$G_{\text{sum}} = \frac{1}{n}\sum_{n=1}^{n}(G_c^L + G_n^L + \gamma_n G_n^E + \eta_n G_n^M). \quad (12)$$

其中:G_c^L为云服务器的负荷收益;G_n^L为边缘客户端n的负荷收益;G_n^E为边缘客户端n的能耗收益;G_n^M为边缘客户端n的模型精度收益;γ_n和η_n分别为能耗和模型精度的权重,满足$\gamma_n + \eta_n = 2$. 不同的权重表示该客户端对能耗或精度的需求偏好,不同的边缘设备可以根据实际场合需求进行权衡,例如:闹市区的客户端可采用较好的CNN模型进行视频分析,以获得更高的分析精度;而偏远地区临时部署的客户端则更关注续航问题,希望设备能耗尽可能低. 若无特殊需求,则默认$\gamma_n = \eta_n = 1$.

值得注意的是,式(12)中,总量最后除以客户端数量n,是将收益平均至每个客户端,这么做是为了让不同数量客户端组成的系统之间也能进行收益对比,否则容易导致客户端数量越多,系统总收益便越高的情况.

1.3.1 负荷收益

负荷收益分为客户端负荷收益G_n^L和服务器负荷收益G_c^L,两者获取收益的规则相同.经过考虑及文献查阅,设备长时间处于超负荷状态会降低系统稳定性[18-19],但负荷率太低又会导致系统资源浪费,因此,将设备的负荷率保持在80%左右是较为合理的选择. 负荷率获得的收益规定如下:当设备的负荷率在80%时可获取最高的负荷收益,当负荷率降低时,收益随之降低,最低为0;当负荷率高于80%时,收益也会有所降低;当负荷率高于100%时,意味着当前设备无法满足算力需求,必定有视频块被放弃处理,作为惩戒,此时收益将设为负,负荷率超出越高,惩戒越重,即负收益越高.

客户端n的负荷率为L_n^e,则其负荷收益为

$$G_n^L = \begin{cases} 0,\ 0 \leqslant L_n^e < 0.4; \\ 1 - 2.5|L_n^e - 0.8|,\ 0.4 \leqslant L_n^e \leqslant 1; \\ 0.5(1 - L_n^e),\ 1 < L_n^e. \end{cases} \quad (13)$$

服务器的负荷率为L_n^c,则其负荷收益为

$$G_c^L = \begin{cases} 0,\ 0 \leqslant L_n^c < 0.4; \\ 1 - 2.5|L_n^c - 0.8|,\ 0.4 \leqslant L_n^c \leqslant 1; \\ 0.5(1 - L_n^c),\ 1 < L_n^c. \end{cases} \quad (14)$$

1.3.2 能耗收益

对客户端而言,在运行期间能耗越低,获得的能耗收益越高. 客户端的最低能耗是其待机能耗,即$D_n P_n^w$. 其中:P_n^w为待机功率,D_n为整个周期时长. 当客户端使用最大模型3进行视频分析时,功率为$P_n^p(3)$,客户端能达到的最高能耗为$E_n^{\max}$,表示为

$$E_n^{\max} = D_n P_n^p(3). \quad (15)$$

在最低与最高能耗区间之内,定义能耗的收益为

$$G_n^E = \frac{E_n^{\max} - E_n^{\text{sum}}}{E_n^{\max} - D_n P_n^w}. \quad (16)$$

1.3.3 精度收益

根据客户端n在周期内获得的实际平均精度A_n^{avg},其对应的精度收益为

$$G_n^M = \frac{A_n^{\text{avg}} - A_n^1}{A_n^3 - A_n^1}. \quad (17)$$

在真实世界中,输入的交通视频不存在标签,无法实时验证视频精度,因此本文只能基于当前已知CNN模型在特定数据集上的精度效果代替真实世界中视频分析的精度. 也有将累积置信度作为精度评判的方法[20],可以避免使用标签,但本文经过实验测试后发现其效果不佳,原因是评判本文CNN模型精度的指标是P-R曲线(precision-recall curves)[21-23],而仅靠置信度无法评估P-R曲线面积.

1.4 优化目标

基于数学模型和收益函数,对于追求负荷均衡并使系统收益最大化,可以写成以下优化问题:

$$\arg\max G_{\text{sum}}(X). \quad (18)$$

$$\text{s.t. } X = \{(r_1, m_1), (r_2, m_2), \ldots, (r_n, m_n)\};$$

$$L_i^e(r_i, m_i) \leqslant 1,\ i = 1, 2, \ldots, n; \quad (19)$$

$$L_n^c(r_n) \leqslant 1. \quad (20)$$

其中:$G_{\text{sum}}(X)$为系统收益;X为系统策略;式(19)表示所有客户端负荷率必须小于1;式(20)表示服务器负荷率必须小于1.

本文算法设计的最终目标是通过调整策略X,使系统收益$G_{\text{sum}}(X)$最大化. 在满足限制条件下,求X的最优解是一个NP-hard难题,其主要挑战如下:

1) 策略数量众多:任意客户端可选3种模型和11种卸载率,n个客户端组成的系统便有33^n种卸载策略,需要从中选出最优策略较为困难.

2) 变量关系复杂:云端与客户端存在资源竞争,客户端能耗与精度之间存在冲突,负荷率对能耗与精度有影响,系统中各变量的关系密切且复杂.

2 决策算法

2.1 算法框架

本系统的算法框架基于数据驱动,将决策变量$X = (r_1, m_1), (r_2, m_2), \ldots, (r_n, m_n)$作为框架的输

入,随后框架根据因变量(负荷和收益等)的反馈对策略X进行逐步调整,输出更新后的策略X. 根据1.1节场景描述中所述的决策依据,算法的最终目的是权衡每个决策变量,通过数次迭代使系统达到整体稳态并有效增加收益,其定义如下.

定义1　对于具有n个边缘客户端组成的云边系统$\xi=\{N_1,N_2,\dots,N_n,S\}$,每次决策前系统的初始收益为$G_{\text{sum}}^0(X^0)$,具有决策变量集

$$X^*=\{(r_1^*,m_1^*),(r_2^*,m_2^*),\dots,(r_n^*,m_n^*)\}.$$

其中:r_n^1为决策后卸载率,m_n^1为决策后CNN模型大小. 经过1次算法策略调整后,系统收益为

$$G_{\text{sum}}=G_{\text{sum}}^1(X^1),\tag{21}$$

其中$X^1=\{(r_1^1,m_1^1),(r_2^1,m_2^1),\dots,(r_n^1,m_n^1)\}$. 若经过$x$次迭代后,满足

$$G_{\text{sum}}^x(X^x)\geqslant G_{\text{sum}}^{x-1}(X^{x-1}),\tag{22}$$

则称系统增益有效,X^x为有效策略. 若$\forall y>x$,使得

$$\frac{2|G_{\text{sum}}^y-G_{\text{sum}}^x|}{G_{\text{sum}}^y+G_{\text{sum}}^x}<\mu,\tag{23}$$

则称系统达到整体稳定,X^x为稳定且最优策略. 其中:μ为可接受误差,通常取$\mu=0.05$,但在实际系统中,由于设备受限于温度等影响,本身具有不稳定性,能耗等也存在一定的反馈延迟或误差,因此μ的取值可适当放宽.

2.2　决策流程

如图3所示,算法决策更新发生在每个云结算周期之间,云结算周期由若干个边缘周期组成. 例如:客户端边缘周期是10 s,即$D_n=10$ s;而服务器的云结算周期为60 s,即$C=60$ s,表示每6个边缘周期进行一次云结算,更新卸载策略. 因此,在结算周期60 s内,D_n^1(第1个边缘周期)结束后会立刻进入D_n^2(第2个边缘周期),持续进行,直到60 s(云结算周期C)结束. 在一个云结算周期结束后,云服务器通过算法更新卸载策略,并将更新后的策略调整指令分别发送给每个客户端,客户端根据指令对卸载策略进行更改,随后系统立即进入下一个云结算周期.

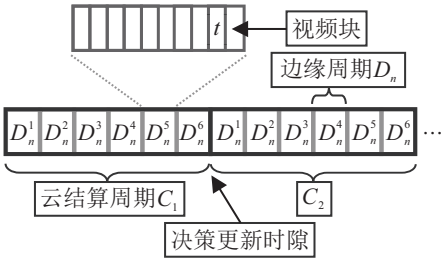


图 3　算法决策周期

2.3　进位调整描述

针对1.4节挑战1)所述的策略复杂问题,提出进位调整思想,增强自变量耦合,降低决策复杂程度.

在十进制中,数字总是按照满十进一的进位规则,同理,二进制数则按照满二进一的进位规则,两者本质上的区别是位的权重不同. 照此思路,对于客户端n的策略(r_n,m_n),卸载率r_n和模型选择m_n共同决定了客户端负荷$L_n^e(r_n,m_n)$的大小,但r_n和m_n对L_n^e的影响程度不同,在卸载率r_n调整幅度过大而导致负荷L_n^e剧烈波动的情况下,可以通过先改变模型m_n再改变卸载率r_n使负荷率L_n^e波动降低,类似于数字的进位操作. 此外,为进一步增强系统稳定性并减少搜索时间,在客户端调整卸载率时,限制其调整步长在0.3之内. 具体定义如下.

定义2　客户端n的原策略为(r_n^0,m_n^0),对应负荷为L_n^0;被算法调整卸载率后的拟定目标策略为(r_n^*,m_n^0),对应负荷为L_n^1.若存在可变模型参数m_n^*替代m_n^0,使得负荷L_n^*处于L_n^0与L_n^1之间,则称(r_n^*,m_n^*)是(r_n^*,m_n^0)的进位调整策略,其中m_n^*为进位模型.

2.4　贪心阶段优化

针对1.4节挑战2)所述的变量关系复杂问题,提出贪心阶段优化思想,分阶段多次优化不同目标.

第1阶段目标是优化客户端的偏好需求,提升客户端能耗和精度收益.根据客户端的不同需求,让客户端采用不同的CNN模型.

第2阶段目标是优化客户端的负荷,提升客户端负荷收益. 该阶段针对每个客户端进行单独调整,根据客户端当前卸载率,进行梯度式范围搜索,选取高收益策略.

第3阶段目标是优化服务器的负荷,提升服务器负荷收益. 该阶段针对所有客户端进行调整,根据服务器负荷大小,升高或降低所有客户端的卸载率.

2.5　决策算法描述

卸载策略X调整的具体步骤如下,伪代码如表1所示,相关变量的描述在表2中.

step 1: 云结算周期结束,云端立刻计算并保存运行结果的输出和收益,以及所有决策参数r_i^0和m_i^0.

step 2: 若客户端i的偏好权重$\eta_i<1$,则模型调整为$m_i=1$;若$\eta_i>1$,则$m_i=3$.

step 3: 根据客户端卸载率r_i^0,分别搜索计算$r_i^0\pm0.3$步长内所有模型的预估负荷收益并取其最大值G_i^L,保存取得最大收益时的策略参数r_i^*和m_i^*;若偏好权重不为1,则保留原模型参数m_i^0.

step 4: 重复执行step 2, 直到所有边缘设备的调整完毕后进入step 5.

step 5: 根据云负荷L_i^c, 拟调整所有客户端的卸载率, 该调整会与step 3中的卸载率调整融合, 融合后的卸载率为最终确定的卸载率, 并据此最终拟卸载率对模型是否进位进行判定.

step 6: 记录并保存所有策略参数X^*.

step 7: 程序执行完毕, 进入下一个云结算周期.

表1 算法伪代码

decision algorithm

input: $X^0=\{(r_1^0,m_1^0),(r_2^0,m_2^0),\dots,(r_n^0,m_n^0)\}$

result: $L_i^e(r_i^0,m_i^0),L_i^c(r_i^0),i=\{1,2,\dots,n\};G_{\text{sum}}(X^0)$

$X^1\leftarrow\{\}$

for each client i do

  if $\eta_i=1$ then

    for m_i^* in $[1,2,3]$ do

      for r_i^* in $(r_i^0\pm0.3)$ do

        compute $G_i^L(r_i^*,m_i^*)$

    if max $G_i^L(r_i^*,m_i^*)>G_i^L(r_i^0,m_i^0)$ then

      add (r_i^*,m_i^*) to X^1

  if $\eta_i\neq1$ then

    if $\eta_i<1$ then $m_i^*=1$ else $m_i^*=3$

    for r_i^* in $(r_i^0\pm0.3)$ do

      compute $G_i^L(r_i^*,m_i^*)$

    if max $G_i^L(r_i^*,m_i^*)>G_i^L(r_i^0,m_i^0)$ then

      add (r_i^*,m_i^*) to X^1

if $L_i^c(r_i^0)\neq0.8$ then

  for each client i do

    $r_i^{**}=r_i^0-(L_i^c(r_i^0)-0.8)$

    $X^1\leftarrow(r_i^*+r_i^{**})\&\Delta m$

return X^1

表2 变量表

序号	变量名	序号	变量名
x^0	输入策略	x^1	输出策略
L_i^e	边缘客户端负荷	L_i^c	云服务器负荷
G_i^L	客户端负荷收益	Δm	模型进位

3 系统实现

3.1 设备构建

根据使用场景, 本框架构建使用如下软硬件设备: 一台服务器, 其CPU为英特尔11 700K, GPU为NVIDIA 3090, 内存容量为64 G; 一台NVIDIA Jetson Xavier NX作为边缘设备; 云服务器和边缘设备的操作系统均为Ubuntu20.04, 使用python3.7开发; 使用FFmpeg对视频进行编解码, 得到时间长度固定的视频块比特流数据; 使用基于TCP的RTP协议进行数据传输, 通信带宽为100 MHz; 使用RabbitMQ作为云服务器控制的数据收发窗口.

3.2 精度实验

本文使用3种CNN模型进行视频分析, 分别为Yolov5n、Yolov5s、Yolov5m, 所有模型的训练基于BDD100K交通数据集[24]的图片子集. 为了准确地评估CNN模型能力, 本文采用mAP (mean average precision)[25]评估模型精度, 3种CNN模型的精度结果如图4所示.

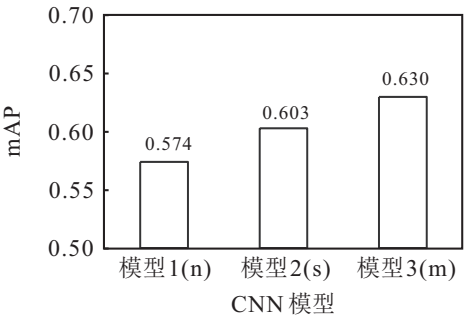


图4 不同CNN模型精度

3.3 性能实验

云服务器和边缘设备的性能共同决定了边缘设备最大可连接数量. 本实验中云服务器固定使用模型2, 边缘设备选择使用任意3种模型, 在有限带宽下进行视频块传输和分析, 得到服务器和客户端的性能结果如表3所示, 其中云服务器数据处理时间包括数据从边缘设备传输至服务器的时间. 根据实验结果, 服务器的计算能力较强, 影响服务器总速度的瓶颈为视频传输成本较高, 而客户端本地计算则不存在传输速度限制, 但其硬件设备计算能力较弱, 无法进行快速视频分析.

表3 不同设备处理视频的耗时 单位: s

	模型	数据处理	视频分析	总时长
服务器	模型2	0.384	0.166	0.55
客户端	模型1	0.164	0.947	1.111
	模型2	0.164	1.291	1.455
	模型3	0.164	2.501	2.665

3.4 能耗实验

在一个边缘周期D_n内, 客户端的能耗曲线如图5所示.

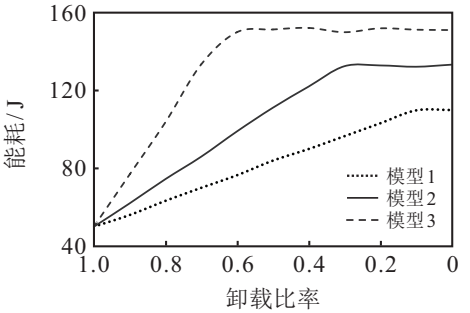


图5 不同卸载率的能耗

当卸载率降低至一定程度后,留在本地处理的视频块数量较多,客户端已处于满负荷,此时即使继续降低卸载率,客户端也无法再处理更多的视频块,故能耗也不再增长. 通过对比可知,同样是满负荷状态,在使用大模型的情况下能耗会更高. 因此,对于客户端而言,使用大模型的成本较高,延迟和能耗都会增加,需占用较大的系统资源换取精度的提升.

4　仿真实验与分析

4.1　实验环境

根据实际场景,基于3.1节所述的配置,本次实验选择一台云服务器和3台边缘设备进行仿真. 仿真实验环境为: 使用BDD100K数据集中的50个test交通视频,其每帧大小为1 280×720,每秒30帧; 客户端的初始卸载率为0.9; 视频块长度取$t = 1$ s,边缘周期取$D_n = 10$ s,云结算周期为60 s,即60 s更新一次卸载策略,系统一共运行10 min,即进行10次云结算. 客户端初始参数设定如表4所示.

表4　实验初始设置

	客户端1	客户端2	客户端3
偏好权重	$\eta_1 = 1$	$\eta_2 = 1$	$\eta_3 = 1$
CNN模型	$m_1^0 = 1$	$m_2^0 = 2$	$m_3^0 = 3$

FastVA[11]采用一种卸载思路,即尽量将任务卸载至云服务器计算以提高精度,剩余任务留在本地处理或丢弃,作为对比,EdgeVison[14]中提到了一种尽可能把任务留在本地处理的卸载方式. 以上两种方式都直接基于客户端进行独立决策,本文提出的方式则基于云服务器进行控制卸载决策,为了验证本文提出的方式具有一定优越性,首先将上述3种方式进行对比分析,具体如下:

1) 卸载优先: 客户端根据服务器负荷率,优先将视频卸载至服务器;

2) 本地优先: 客户端根据自身负荷率,优先将视频留在本地处理,减少视频卸载;

3) 阶段优化算法: 服务器根据算法调整所有客户端卸载策略,最大程度提高系统收益.

4.2　结果与分析

4.2.1　负荷分析

在系统运行持续时间内,每1 min进行一次结算,读取设备负荷率变化,如图6所示.

卸载优先的策略是增加卸载,使服务器长时间维持满负荷,而客户端负荷会相对较低; 本地优先的策略是减少卸载,使本地维持满负荷,而服务器负荷会相对较低; 本文算法的策略是尽可能使客户端与服务器之间的负荷维持于80%附近,以获得更高的收益. 实际上,设备超负荷运行会导致视频块丢失,对交通监控造成较大损失,3种卸载方式的视频丢失情况如图7所示.

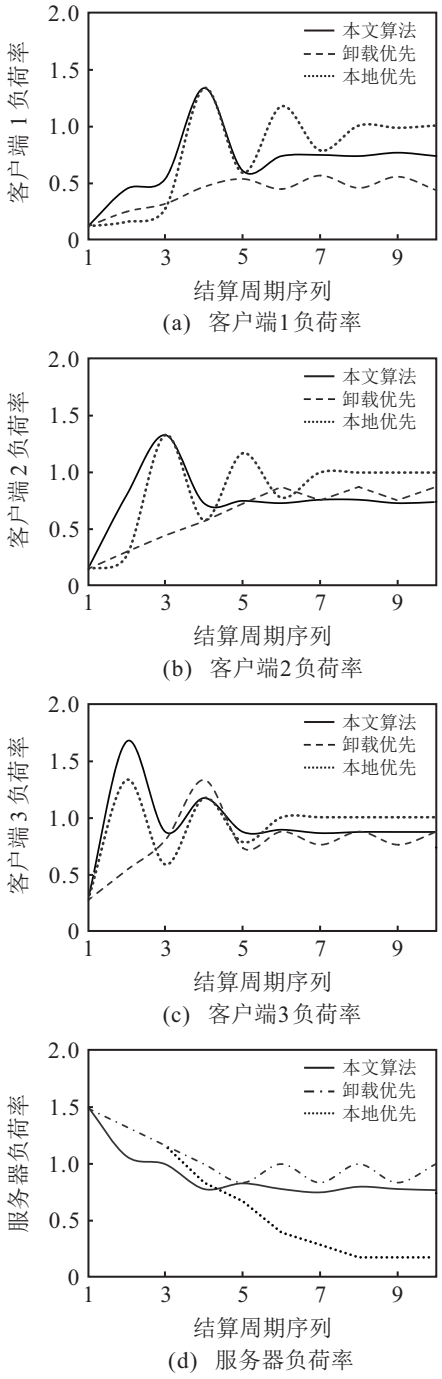


图 6　系统各节点负荷率

基于卸载优先的方式容易导致系统不稳定,原因是云端接入客户端较多,而卸载优先的方式是通过本地客户端对服务器负荷率的判断来单独决定是否增加或减少任务卸载,当大多数客户端做出相同策略时,比如都认为服务器空闲而选择增加卸载任务至服务器,便容易导致服务器拥挤甚至超负荷.

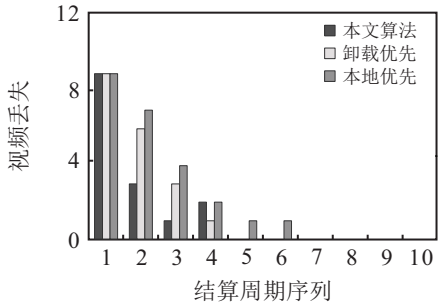


图 7 丢失视频块数量

基于本地优先的方式则致力于在尽量减少卸载的前提下提高模型精度,因此其客户端的负荷率相对偏高,但这在客户端性能较弱时,会导致算法逐渐偏向较小的CNN模型,导致视频精度下降.

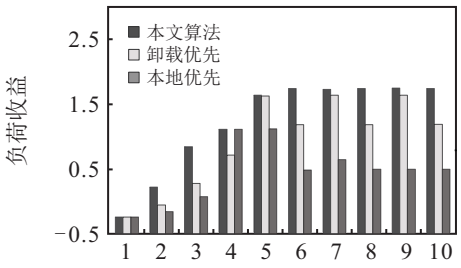
本文采用的阶段优化算法基于云端对全局设备状态的掌控,可根据所有边端和云端的性能状态实时做出最优任务划分. 相较于其他方法,本文的算法更为全面,便于统筹调控资源,有利于系统稳定.

4.2.2 收益分析

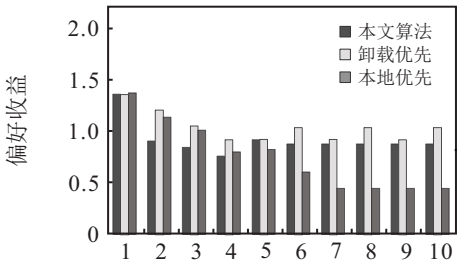
收益主要由负荷收益与偏好收益组成,负荷收益即客户端负荷收益和服务器负荷收益,偏好收益即能耗与精度的加权混合收益,此处能耗与精度的偏好权重相同,即$\eta_i = 1$. 为简化分析,此处将3个客户端的收益进行平均,结果如图8所示.

通过图8(a)可以观察到:本文算法策略相对其他算法策略的最大优势在于对负荷平衡的把控. 在偏好收益方面,基于卸载优先的策略能取得较好的收益,主要是因为卸载到服务器的视频较多,从而客户端的能耗相对较低,而服务器的能耗是不被计算在内的,虽然这么做能提高客户端的能耗和精度收益,但服务器长时间保持满负荷会降低稳定性,影响系统安全. 基于本地优先的算法在本环境下获得的收益较低,这是因为在客户端性能较弱时,该算法会逐渐偏向较小的CNN模型,导致视频精度下降,同时客户端较高的负荷率又导致能量消耗较高.

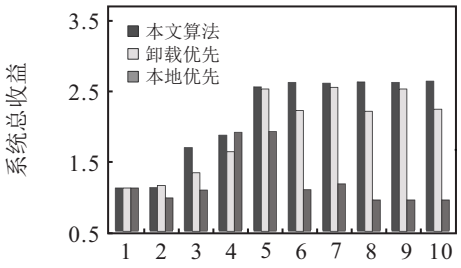
综合而言:卸载优先的方式理论上适用于单客户端对单服务器的网络结构,可以最大化精度与能耗的收益,又无需关注多客户端做出相同决策时导致的服务器负荷波动,但在本文的多客户端场景下不适用;本地优先的方式理论上适用于服务器性能不强或客户端数量较多的情况,可以减少每个客户端上传至服务器的视频数据,最小化服务器压力,但无法应用于有节能需求的场景,具有一定的局限性;基于阶段优化算法决策的方式既可以做到稳定系统,又能做到精度能耗相对平衡,重要的是,本文方法还能通过



(a) 负荷收益对比



(b) 偏好收益对比



(c) 系统总收益对比

图 8 收益比较

调整偏好权重控制决策偏向,且具有一定扩展能力,后续实验将对此进行验证分析.

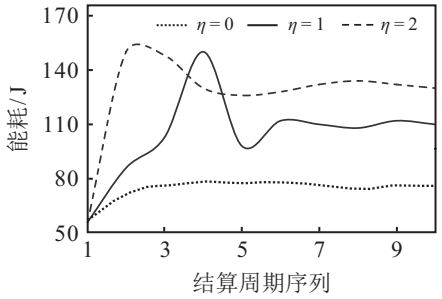
4.2.3 偏好与扩展性分析

首先分析,当系统中客户端1存在偏好要求,即权重η取不同值时的情况,如图9所示,$\eta = 0$代表偏好节能,$\eta = 1$代表无偏好,$\eta = 2$代表偏好精度.

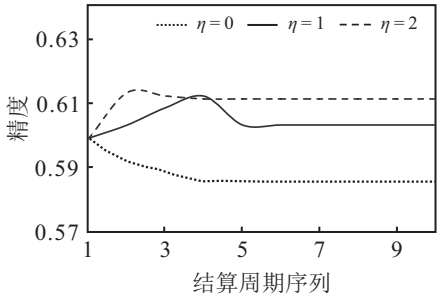
对比图9(a)与图9(b):当$\eta = 0$时,客户端偏向节能,此时精度会相对降低;当$\eta = 2$时,客户端偏向高精度,此时能耗相对较高. 通过图9(b)可以看出:当$\eta = 2$时,算法会让客户端采用大CNN模型进行视频分析,虽然总体上精度有所提升,但提升幅度较小,这是由于客户端的性能较弱,换用较大模型后,客户端能处理的视频块数量减少,导致精度提升有限,在未来的工作中,或许可以让云服务器也部署多种CNN模型,以更好地协同满足客户端精度需求.

最后,为了验证本文算法的扩展能力,在原有3个客户端基础上,额外增加客户端,增加的客户端均为无偏好客户端且初始CNN模型为2,即$\eta = 1, m = 2$. 根据系统中客户端数量的不同,系统在运行期间

的收益变化如图10(a)所示;图10(b)则展示了系统中包含不同数量客户端时收益的构成,该收益为整个系统运行期间的平均收益.

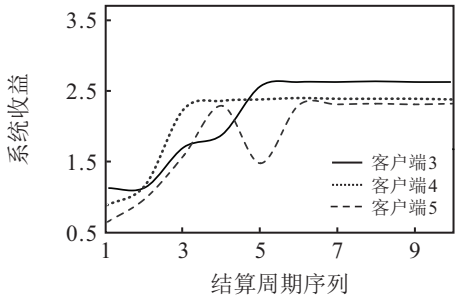


(a) 客户端1不同权重下能耗

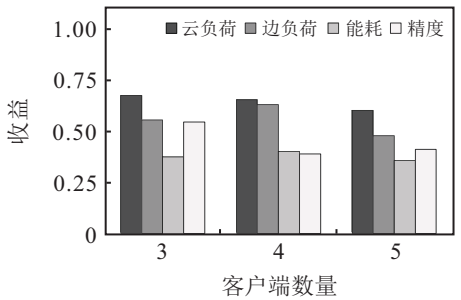


(b) 客户端1不同权重下精度

图 9　偏好权重η对决策结果的影响



(a) 扩展客户端的系统收益



(b) 系统运行过程中的平均收益

图 10　扩展客户端的收益状况

由图10可以看出,在客户端数量增加时,阶段优化算法依然可以做到系统负荷稳定,并使当前系统收益最大化,具有较强的扩展能力. 但值得注意的是,在客户端增加后,系统的整体收益在降低,这是由于系统的算力有限,客户端数量过多会导致算法不得不采用较小的CNN模型,因此不能无限制地增加客户端数量.

5　结　论

本文针对多端实时交通视频分析的场景,采用卸载计算的方式,设计云边卸载框架,提出了一种基于数据驱动的阶段优化卸载算法,能够自适应地平衡多客户端算力需求,并通过基础实验和仿真实验模拟多客户端与单服务器组成的系统运行状态. 实验结果表明,相对于其他卸载方案,本文提出的方案可以较好地平衡多端之间的资源竞争,更好地完成视频分析任务,减少视频数据丢失. 此外,本框架还综合考虑到视频分析精度和能耗等现实因素,能根据不同客户端之间的需求偏好配置计算资源,控制多客户端的稳定运行,并具有较强的扩展性,能针对客户端数量的不同自适应调整卸载策略,为多端云边系统提出一种可行的任务卸载方案.

考虑到真实交通应用场景中,往往存在客户端在系统运行期间的偶然接入或断开,这可能会给系统造成不稳定性. 虽然本文提出的卸载方案具有较强的扩展性,但目前还无法做到在系统运行期间加入或断开某些客户端后继续维持系统稳定. 在未来的工作中,可以针对上述问题展开具体研究.

参考文献(References)

[1] 徐先峰, 杨凡, 刘状壮, 等. 基于两级筛选机制及深度学习组合模型实现短时交通流预测[J]. 控制与决策, 2023, 38(1): 84-92.
(Xu X F, Yang F, Liu Z Z, et al. Combination model of short-term traffic flow prediction based on two-level screening mechanism[J]. Control and Decision, 2023, 38(1): 84-92.)

[2] Lingani G M, Rawat D B, Garuba M. Smart traffic management system using deep learning for smart city applications[C]. The 9th Annual Computing and Communication Workshop and Conference. Las Vegas, 2019: 101-106.

[3] 金沙沙, 龙伟, 胡灵犀, 等. 多目标检测与跟踪算法在智能交通监控系统中的研究进展[J]. 控制与决策, 2023, 38(4): 890-901.
(Jin S S, Long W, Hu L X, et al. Research progress of detection and multi-object tracking algorithm in intelligent traffic monitoring system[J]. Control and Decision, 2023, 38(4): 890-901.)

[4] Jiang Z G, Shi X T. Application research of key frames extraction technology combined with optimized faster R-CNN algorithm in traffic video analysis[J]. Complexity, 2021, 2021: 1-11.

[5] Othmani M. A vehicle detection and tracking method for traffic video based on faster R-CNN[J]. Multimedia Tools and Applications, 2022, 81(20): 28347-28365.

[6] Yao C R, Liu W T, Tang W Q, et al. EAIS: Energy-aware adaptive scheduling for CNN inference on high-performance GPUs[J]. Future Generation Computer Systems, 2022, 130: 253-268.

[7] Tang E Q, Minakova S, Stefanov T. Energy-efficient and high-throughput CNN inference on embedded CPUs-GPUs MPSoCs[C]. International Conference on Embedded Computer Systems. Cham, 2022: 127-143.

[8] Wang X K, Yang L T, Xie X, et al. A cloud-edge computing framework for cyber-physical-social services[J]. IEEE Communications Magazine, 2017, 55(11): 80-85.

[9] Gao Z F, Zhang H Y, Dong S Z, et al. Salient object detection in the distributed cloud-edge intelligent network[J]. IEEE Network, 2020, 34(2): 216-224.

[10] Ran X K, Chen H, Zhu X D, et al. DeepDecision: A mobile deep learning framework for edge video analytics[C]. IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Honolulu, 2018: 1421-1429.

[11] Tan T X, Cao G H. FastVA: Deep learning video analytics through edge processing and NPU in mobile[C]. IEEE INFOCOM 2020-IEEE Conference on Computer Communications. Toronto, 2020: 1947-1956.

[12] Qian B, Wen Z Y, Tang J Q, et al. OsmoticGate: Adaptive edge-based real-time video analytics for the Internet of things[J]. IEEE Transactions on Computers, 2023, 72(4): 1178-1193.

[13] Wang X Z, Gao G Y, Wu X H, et al. Dynamic DNN model selection and inference off loading for video analytics with edge-cloud collaboration[C]. Proceedings of the 32nd Workshop on Network and Operating Systems Support for Digital Audio and Video. New York, 2022: 64-70.

[14] Dong Y Q, Gao G Y, Wang R, et al. Collaborative video analytics on distributed edges with multiagent deep reinforcement learning[J/OL]. 2022, arXiv: 2211.03102.

[15] Tung T Y, Gündüz D. DeepWiVe: Deep-learning-aided wireless video transmission[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(9): 2570-2583.

[16] Nurrohman A, Abdurohman M. High performance streaming based on H264 and real time messaging protocol (RTMP)[C]. The 6th International Conference on Information and Communication Technology. Bandung, 2018: 174-177.

[17] Schreier R M, Rothermel A. A latency analysis on H.264 video transmission systems[C]. 2008 Digest of Technical Papers-International Conference on Consumer Electronics. Las Vegas, 2008: 1-2.

[18] Kumar N, Mishra N. Load balancing techniques: Need, objectives and major challenges in cloud computing — A systematic review[J]. International Journal of Computer Applications, 2015, 131(18): 11-19.

[19] Sreenivas V, Prathap M, Kemal M. Load balancing techniques: Major challenge in cloud computing — A systematic review[C]. 2014 International Conference on Electronics and Communication Systems. Coimbatore, 2014: 1-6.

[20] Zhang L, Zhang Y Q, Wu X M, et al. Batch adaptative streaming for video analytics[C]. IEEE INFOCOM 2022-IEEE Conference on Computer Communications. London, 2022: 2158-2167.

[21] Cook J, Ramadas V. When to consult precision-recall curves[J]. The Stata Journal: Promoting Communications on Statistics and Stata, 2020, 20(1): 131-148.

[22] Robertson S. A new interpretation of average precision[C]. Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, 2008: 689-690.

[23] Yue Y S, Finley T, Radlinski F, et al. A support vector method for optimizing average precision[C]. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, 2007: 271-278.

[24] Xu H Z, Gao Y, Yu F, et al. End-to-end learning of driving models from large-scale video datasets[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 3530-3538.

[25] Chen M Q, Yu L J, Zhi C, et al. Improved faster R-CNN for fabric defect detection based on Gabor filter with genetic algorithm optimization[J]. Computers in Industry, 2022, 134: 103551.

作者简介

温震宇 (1987−), 男, 教授, 博士, 从事大数据、分布式系统等研究, E-mail: zhenyuwen@zjut.edu.cn;

胡慧峰 (1996−), 男, 硕士生, 从事云边系统的研究, E-mail: 2112003095@zjut.edu.cn;

钱滨 (1993−), 男, 博士生, 从事分布式系统、人工智能等研究, E-mail: b.qian3@ncl.ac.uk;

洪榛 (1983−), 男, 教授, 博士生导师, 从事物联网应用及安全、网络和数据安全等研究, E-mail: zhong1983@zjut.edu.cn;

俞立 (1961−), 男, 教授, 博士生导师, 从事网络化控制、鲁棒控制等研究, E-mail: lyu@zjut.edu.cn.