

多分辨率下资源感知的图像目标自适应缩放检测

张开强¹ 蒋从锋¹ 程小兰¹ 贾刚勇¹ 张纪林² 万 健³

1 杭州电子科技大学计算机学院 杭州 310018

2 杭州电子科技大学网络空间安全学院 杭州 310018

3 浙江科技学院信息与电子工程学院 杭州 310023

(zkqcs@hdu.edu.cn)

摘要 边缘视频处理可以降低云平台视频处理系统的视频传输时延、视频处理开销和存储成本,但是视频参数(分辨率、帧率等)的多样性容易导致边缘视频处理的效果不尽人意。通常,在图像预处理阶段会先对图像进行缩放变换再进行后续处理,以保障图像处理的最佳效果,但是在视频监控等具有不确定性的场景中对所有分辨率的图像直接成倍缩小容易降低目标检测率。基于以上问题,把图像水平像素点和垂直像素点的缩放倍数记作图像缩放因子,对于不同分辨率的视频数据,分析了图像缩放因子对视频数据处理效果的影响,提出了图像缩放因子动态设置方案。该方案以系统性能指标(服务器端系统功耗和内存使用率)为视频处理性能指标(人脸检测率)的约束条件,获取该分辨率下人脸检测率最优时对应的图像缩放因子。实验结果表明,对于不同分辨率的视频数据,图像缩放因子动态设置方案可以在保证视频处理性能的基础上,减少系统功耗和内存使用率,提高视频处理效率。

关键词: 边缘计算;视频处理;视频参数;图像缩放;人脸检测

中图法分类号 TP391

Resource-aware Based Adaptive-scaling Image Target Detection Under Multi-resolution Scenario

ZHANG Kai-qiang¹, JIANG Cong-feng¹, CHENG Xiao-lan¹, JIA Gang-yong¹, ZHANG Ji-lin² and WAN Jian³

1 School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

2 School of Cyber Science and Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

3 School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

Abstract Edge video processing can reduce the video transmission delay, video processing overhead and storage cost in the video processing system of cloud platform, but the diversity of video parameters (resolution, frame rate, etc.) will lead to unsatisfactory effect of edge video processing. Usually, in the stage of image pre-processing, image scaling and transformation will be applied to ensure the best effect of image processing. However, in uncertain scenes such as video monitoring, the target detection rate can be reduced largely by directly scaling down the resolution of all the images. According to the problems mentioned above, this paper chooses the scaling ratio of the horizontal and vertical pixels as image scaling factor. For the video data with different resolutions, it analyzes the influence of the image scaling factor on the performance of video data processing and puts forward the dynamically-setting scheme of image scaling factor. In this scheme, the system performance index (system power consumption and memory utilization of the server-side) is taken as the constraint condition of video processing performance index (face detection rate), to get the image scaling factor corresponding to the optimal face detection rate with this resolution. Experimental results show that for video data with different resolutions, the dynamic setting scheme of image scaling factor can reduce system power consumption and memory utilization, improve video processing efficiency and capability while guaranteeing video processing performance.

Keywords Edge computing, Video processing, Video parameters, Image scaling, Face detection

1 引言

有了更进一步的期望。为了适应万物互联时代的变化,边缘计算应运而生。在物联网范例中,具有强大计算能力的大量异构设备将相连,但其中很多设备的本地计算资源不足以运

随着物联网的快速发展,用户对数据处理服务的实时性

到稿日期:2020-12-14 返修日期:2021-03-01 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金面上项目(61972118,61972358),浙江省重点研发计划项目(2019C01059)

This work was supported by the General Program of National Natural Science Foundation of China(61972118,61972358) and Zhejiang Key Research and Development Project of China(2019C01059).

通信作者:蒋从锋(cjiang@hdu.edu.cn)

行复杂的应用程序。为了打破上述资源瓶颈,可以将资源需求较大的任务(尤其是视频处理任务)从资源受限的移动设备卸载到附近具有更多计算资源的服务器上执行。计算卸载是根据实际的应用场景指定不同的卸载策略,从而在系统能耗和系统性能之间达到平衡^[1]。具体的卸载策略包括:何时卸载、卸载到何处、如何卸载以及卸载哪一部分任务等^[2-9]。

虽然边缘端各节点之间的拓扑关系不及云中心复杂,但是边缘服务器和边缘节点之间也会存在多对多的关系,即一个边缘视频处理服务器会处理来自多个边缘节点的视频流,一个边缘节点的视频流也会被多个边缘服务器处理。本文主要考虑单个边缘服务器处理多路边缘节点数据的情况,这种一对多的处理关系对边缘服务器的性能要求更高。此外,视频参数的多样性也会对边缘服务器的性能提出挑战。视频参数的多样性体现在两个维度,分别是视频参数的种类以及视频参数的取值范围。通常情况下,视频参数有分辨率、帧率、码率、采样率和视频格式等;而视频参数的取值范围取决于视频参数的种类,例如帧率可以取 25~55 fps 之间的任意值。

视频参数的多样性会加大视频处理的难度,降低视频处理的性能。本文实验中,边缘服务器的视频处理应用程序是基于 OpenCV 的人脸检测,整个过程分为离线的模型训练、在线的图像预处理和人脸检测 3 个阶段。图像分辨率对每个阶段的影响都不可忽略,它会直接影响视频处理的效果(人脸检测率),也会间接影响边缘服务器的性能指标(CPU、内存和功耗等)。对于所有分辨率的视频流而言,视频分辨率的范围跨度大,导致边缘服务器系统性能和视频处理性能很难同时达到最优。本文的目的是,对于不同分辨率的视频流,边缘服务器尽可能在自身系统性能和视频处理性能之间达到平衡。

为了保证后续图像处理的最佳性能,在图像预处理阶段对输入图像进行缩放。本文把图像水平像素点和垂直像素点的缩放倍数记作图像缩放因子(Scaling Factor)。图像缩放因子的大小对实时视频处理系统的视频分析性能和其他系统性能均有影响。如果只考虑图像处理效果,则使用原始大小的图像效果最佳。由于系统内存和 CPU 的限制,导致直接处理高分辨率的原始图像会加剧系统负担,消耗大量的计算资源。不仅如此,边缘节点采集、发送高分辨率视频流会产生更多能耗,而且传输高分辨率的视频流也需要更大的网络带宽。

在不考虑实际应用场景的实验环境中,输入图像可以成倍缩小。但是,实时图像采集一般应用于视频监控等具有不确定性的生活场景中,因此直接成倍缩小图像并不可取。如本文实验中的最大视频分辨率为 1920×1080 ,直接缩小为原来的 $1/4$ 后为 480×270 ,缩放后的图像虽然会大大缩短图像处理时间,但并没有发挥出高分辨率视频的优势,如果在监控场景中使用,将降低监控效果以及目标识别的准确率。

由上述分析可知,为了节省边缘服务器的系统资源,在图像预处理阶段需要对图像进行缩放,但是不可直接成倍缩小。此外,视频分辨率具有多样性而且参数范围跨度很大,为了保证所有分辨率视频的处理性能,每种分辨率应该对应一个合理的缩放参数,而不是统一设置。因此,合理的做法是,针对每一种输入视频流,系统自动设置一个合理的缩放倍数,在满足系统性能参数(CPU、内存、功耗)阈值的情况下最优化视频

处理性能参数(帧接收率、人脸检测率)。本文把上述方案记作动态图像缩放因子设置方案,该方案的具体实现将在第 3 节中详细介绍。

2 相关工作

2.1 边缘视频处理

借助边缘计算的优势,边缘视频处理可以解决云中心视频处理中面临的很多问题。就近处理可以减少网络传输的开销,增强数据的安全性,还可以降低视频数据的传输时延,解决视频处理的实时性问题。但是,由于边缘服务器的资源有限,视频的实时处理也会产生诸多问题。

Bilal 等^[10]介绍了交互式媒体边缘计算的潜力和在多媒体应用以及视频流转码中使用边缘计算具有节省成本、带宽以及降低能耗的好处。Long 等^[11]提出了一种边缘计算框架,其中的关键挑战是如何将移动设备形成最佳的视频处理组,并将视频块分派给适当的视频处理组。Bailas 等^[12]主要探讨了边缘视频处理能力,表明基本的图像处理操作也可以在低功耗的网关设备上。关于视频参数的多样性问题,Tran 等^[13]设想 MEC(Mobile Edge Computing)网络中的一种按需视频流的协作联合缓存和处理策略,旨在增强广泛使用的自适应比特率流媒体技术。Jiang 等^[14]为了提高图像的缩放质量,提出了一种自适应图像缩放算法。当原图像像素点处于边缘区域时,沿 Sobel 算子确定的边缘方向进行自适应插值,当原图像像素点处于非边缘区域时,采用双线性插值法进行插值。Wang 等^[15]为了提高视频中人脸检测的检测速度,采用回归分析方法预测连续视频中人脸中心位置坐标,并通过调整区域宽度系数来确定人脸区域位置,从而提出了一种人脸检测加速算法。

Shi 等^[16]针对传统视频监控的不足,给出了基于边缘计算的新型视频监控系统的展望。Ge 等^[17]为了降低基于云平台的视频监控传输时延、开销和存储成本等,提出了一种基于边缘计算的视频监控框架。

2.2 边缘视频分析平台

为了解决时延问题,Mangiante 等^[18]提出 LAVEA。它是一个建立在边缘计算平台上的系统,可以卸载客户端和边缘节点之间的计算任务,与附近的边缘节点协作,在靠近用户的地方提供低延迟的视频分析。Hu 等^[19]提出一种基于边缘的混合人脸识别系统,即在边缘处计算人脸标识符,在云端进行人脸匹配,这种分工模式避免了大量的视频传输。Zhang 等^[20]提出一个分布式框架,即基于边缘的实时绑架者跟踪平台,通过跟踪绑匪的车辆来实现绑匪跟踪应用程序,从而让周边设备进行实时视频分析。Zhang 等^[21]提出一种面向边缘计算的目标追踪应用部署策略。Sun 等^[22]提出一个基于移动边缘计算的视频流处理平台 mVideo,该平台可以通过协调边缘和云端的计算资源来进行视频处理。Long 等^[23]提出一种边缘计算框架,借助资源较为丰富的边缘移动设备来对延迟比较敏感的多媒体 IoT 任务进行协同处理。

3 资源感知的图像自适应缩放

3.1 图像自适应缩放方案思想

在具有实际应用意义的视频监控场景中,图像缩放因子

的值不能随意给出。图像缩放因子会影响视频处理性能,如在人脸识别的应用场景下,图像缩放因子会影响到帧接收率和人脸检测率。此外,图像缩放因子还会影响边缘服务器的系统性能指标,如内存使用率和系统功耗等。若将视频设置为统一的图像缩放因子,则会导致各种性能指标参差不齐。因此,在进行实时视频处理时,对不同分辨率的视频设置不同的图像缩放因子可以保障各个性能指标达到较优水平。

本文提出的动态图像缩放因子设置方案主要分为离线和在线两部分。离线部分的方案如下:首先给出4个性能指标,其中包含2个视频处理性能指标(分别是帧接收率(FRR)和人脸检测率(FDR)),以及2个系统性能指标(分别是系统内存使用率(Memused)和服务器端系统功耗(Power));然后,列举5个图像缩放因子整数值1~5。如果需要得到更加精确的结果,可以缩小图像缩放因子的取值范围,同时给出系统性能指标的阈值范围,即假设内存使用率不可大于90%,系统功耗不可大于120W。具体步骤如下:

(1)对于11种分辨率的视频数据,经实验得出帧接收率、人脸检测率、服务器端功耗和服务器端系统内存使用率4个性能指标随图像缩放因子变化的分布情况。

(2)根据(1)中的实验结果,通过曲线拟合的方式获得4个性能指标关于图像缩放因子的相关关系。

(3)以系统性能指标的阈值范围为视频处理性能指标的约束条件,即以服务器端功耗和内存使用率的阈值范围为值域,分别计算相应的图像缩放因子定义域,然后取两个定义域的交集作为图像缩放因子的定义域,最后在此定义域范围内寻找人脸检测率最大时的图像缩放因子作为该分辨率最优的图像缩放因子。

(4)形成视频分辨率和图像缩放因子的一一映射关系。

在线部分根据分辨率和图像缩放因子的映射关系,为输入的视频流设置图像缩放因子。对于每一种分辨率视频数据,获取最佳图像缩放因子的方案的流程如图1所示。由第3.2节中的图像自适应缩放方案验证部分可知,服务器端的功耗和内存使用率与帧接收率的单调趋势一致,即不可作为帧接收率的约束条件。

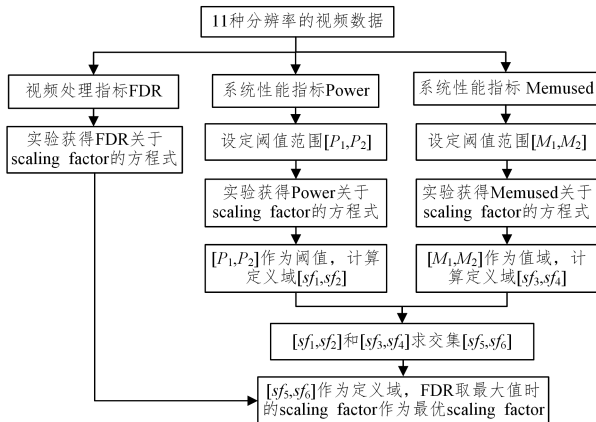


图1 获取最佳图像缩放因子的方案流程图

Fig. 1 Scheme flow of obtaining the best image scaling factor

大量CPU资源,同时庞大的视频流数据也会占据大量存储资源。此外,视频传输通过流媒体协议进行网络传输,尽管存在成熟的视频压缩算法,但是实时视频流传输仍然会消耗大量网络资源。因此,在实时视频处理系统中,从边缘节点到网络再到边缘服务器会消耗大量资源。本文在动态图像缩放因子设置方案的基础上扩展出了资源(网络资源与边缘节点的电能资源)感知的动态图像缩放因子设置思想。

在实时视频处理系统中,当视频帧率不变时,随着视频分辨率的提高,消耗的系统资源会相应地增加,严重时由于帧堆积会出现丢帧现象。此外,网络环境也会加剧系统丢帧。在无线网环境中,由于信号干扰,即便是传输低分辨率视频流系统也会出现丢帧的情况,而传输高分辨率视频流会出现更加严重的丢帧现象。因此,针对网络不稳定这一因素,本文提出了基于网络感知的动态图像缩放因子设置方案来提高视频处理的综合性能。其思想为:在服务器端进行视频流实时处理的过程中,如果检测到导致丢帧的原因是网络中的信号干扰,可动态地将图像缩放因子设置成较小值,从而提高图像处理的精确度。

图2给出了图像缩放因子分别为1.0(图像大小不变)和4.0(图像水平像素点数和垂直像素点数是原图像的1/4)时,服务器端的系统功耗随着视频分辨率提高的变化情况。经分析可知,两种情况下服务器端的系统功耗都随着视频分辨率的提高而增加,但是可以明显看出图像缩放因子为4.0的总体趋势低于图像缩放因子为1.0的情况。图像缩放因子为1.0时,随着分辨率的提高,系统功耗递增趋势不明显的原因是内存资源不足,导致系统丢帧。由此可知,图像缩放因子的大小对边缘服务器端的系统功耗有很大的影响。观察图2中的数据可知,当分辨率为 960×540 、图像缩放因子为1.0时,服务器端的系统功耗比图像缩放因子为4.0时高出81.13%。因此,可以采用功耗感知的动态图像缩放因子设置方案,在保障视频处理性能的前提下尽可能减少边缘服务器端的系统功耗。其思想为:当边缘服务器检测到本地的电能资源充沛时,使用较小的图像缩放因子来提高视频处理性能;当本地电能资源稀缺时,可以使用较大的图像缩放因子,在保障系统性能的前提下提高视频处理的综合性能。

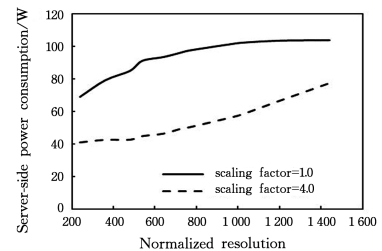


图2 不同分辨率配置下的服务器端功耗

Fig. 2 Server-side power consumption in different resolution configurations

3.2 图像自适应缩放方案验证

(1)离线部分的第一步。探究对于不同分辨率的视频数

实时视频处理是一种典型的CPU密集型任务,它会消耗

据,当使用不同图像缩放因子时,边缘服务器的系统功耗、内存使用率、帧接收率和人脸检测率(由于有帧缓冲队列系统几乎不存在丢帧的情况,关于 FRR 实验的实验环境是无帧缓冲队列系统)4 个性能指标的变化情况。4 个性能指标随着视频分辨率和图像缩放因子的变化而变化的情况如表 1—表 4 所列,对应的可视化图如图 3—图 6 所示。

表 1 不同缩放因子和分辨率配置下的边缘服务器端的系统功耗

Table 1 Power consumption of edge server-side with different scaling factors and resolution configurations

(单位:W)

Normalized resolution	scaling factor				
	1	2	3	4	5
240	68.90	42.92	40.22	40.89	40.79
360	79.05	47.02	42.09	42.46	42.10
480	84.94	51.49	45.63	42.57	42.38
540	91.12	51.56	47.30	44.74	43.39
640	93.39	61.61	53.24	46.24	45.45
720	95.99	66.16	54.72	49.06	45.90
768	97.52	68.46	56.39	50.23	47.21
960	101.27	92.04	67.05	56.00	51.62
1024	102.30	93.88	69.88	58.25	54.48
1200	103.50	98.70	83.46	66.38	58.77
1440	103.79	99.36	95.10	77.16	68.40

表 2 不同缩放因子和分辨率配置下的边缘服务器端的内存使用率

Table 2 Memory usage of edge server-side with different scaling factors and resolution configurations

(单位:%)

Normalized resolution	scaling factor				
	1	2	3	4	5
240	10.72	10.23	10.88	10.75	10.36
360	9.60	9.73	9.52	9.45	9.38
480	11.67	10.51	9.92	9.78	9.72
540	13.04	11.07	10.21	9.99	9.93
640	17.12	11.78	10.61	10.39	10.26
720	26.45	12.66	11.12	10.65	10.49
768	31.42	13.56	11.43	10.88	10.71
960	54.11	14.47	12.12	11.61	11.23
1024	63.95	15.61	12.70	11.94	12.28
1200	84.30	19.35	13.86	12.84	13.13
1440	86.66	32.11	16.07	14.29	14.24

表 3 不同缩放因子和分辨率配置下的人脸检测率

Table 3 Face detection rate with different scaling factors and resolution configurations

(单位:%)

Normalized resolution	scaling factor				
	1	2	3	4	5
240	99.01	0	0	0	0
360	99.81	99.80	0	0	0
480	98.62	99.21	0	0	0
540	98.04	100	84.41	0.79	0
640	79.80	99.80	99.80	4.32	0
720	65.82	99.61	100	20.58	0
768	58.24	99.21	99.80	98.04	16.27
960	39.80	98.83	99.02	100	73.19
1024	34.90	99.41	99.22	99.61	100
1200	39.14	78.26	99.41	98.04	98.83
1440	40.36	70.18	99.42	100	100

表 4 不同缩放因子和分辨率配置下的帧接收率

Table 4 Frame reception rate with different scaling factors and resolution configurations

(单位:%)

Normalized resolution	scaling factor				
	1	2	3	4	5
240	100	100	100	100	100
360	100	100	100	100	100
480	100	100	100	100	100
540	100	100	100	100	100
640	100	100	100	100	100
720	84.71	100	100	100	100
768	74.31	100	100	100	100
960	52.94	100	100	100	100
1024	53.14	100	100	100	100
1200	39.80	85.29	100	100	100
1440	36.27	76.08	100	100	100

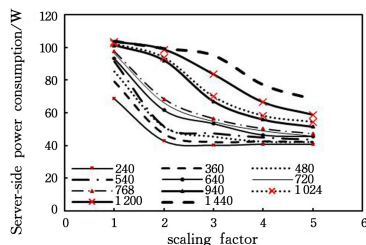


图 3 边缘服务器端系统功耗

Fig. 3 Edge server-side system power consumption

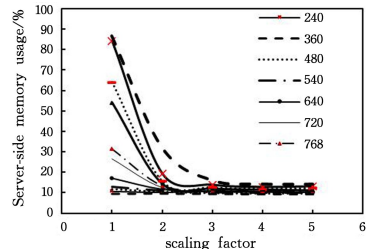


图 4 边缘服务器端内存利用率

Fig. 4 Edge server-side memory usage

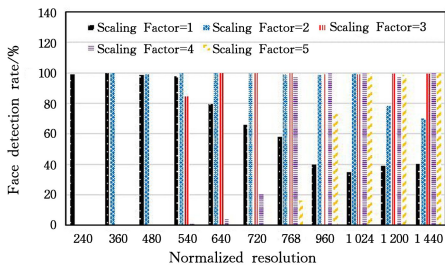


图 5 人脸检测率

Fig. 5 Face detection rate

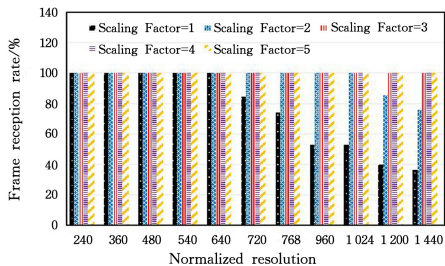


图 6 帧接收率

Fig. 6 Frame reception rate

由图 3 可知,对于所有分辨率的视频而言,服务器端的系统功耗均随图像缩放因子的增大而减少。由图 4 可知,当分辨率为 240(下文皆用归一化后的分辨率表示原始分辨率)时,系统内存使用率随着图像缩放因子的增加几乎没有变化。但是,当分辨率大于 240 时,服务器端的内存使用率均随着图像缩放因子的增大而降低。

由图 5 可知,对于所有分辨率的视频流而言,随着图像缩放因子的增加,人脸检测率有 3 种变化趋势。当视频分辨率较小时,如 240,360,人脸检测率随着图像缩放因子的增大而降低,且在数值上差距很大,人脸检测率最高接近 100%,而最小仅为 0。随着视频分辨率的提高,当分辨率为 480,540,640,720,768,960 时,人脸检测率随着图像缩放因子的增大呈先提高后下降的趋势;当分辨率较大时,如 1 024,1 200,1 440,人脸检测率随着图像缩放因子的增大而提高。

由图 6 可知,随着视频分辨率的提高,无帧缓冲队列系统会出现丢帧。此外,由图 6 还可知,对于出现丢帧的分辨率,帧接收率随着图像缩放因子的增加而提高,当图像

缩放因子等于 3 时,对于所有分辨率的视频流,系统不再出现丢帧。

由上述实验结果可知,随着图像缩放因子的增大,系统内存使用率和系统功耗呈递减趋势,帧接收率呈现递增趋势,即帧接收率、系统内存使用率、系统功耗 3 个指标的性能变化单调性一致。因此,服务器端系统功耗和内存使用率不适合做帧接收率的约束条件。此外,本文已经从帧缓冲队列层面解决系统丢帧问题。

(2)离线部分的第二步。对于特定分辨率的视频流,以服务器端系统功耗和内存使用率为人脸检测率的约束条件,以人脸检测率最优时所对应的图像缩放因子为该分辨率对应的图像缩放因子。首先,以曲线拟合的形式建立人脸检测率、服务器端系统功耗以及内存使用率关于图像缩放因子的函数关系式,结果如表 5 所列。然后,通过内存使用率和系统功耗两个系统性能指标的阈值分别求得图像缩放因子的定义域区间。最后,计算人脸检测率在两个定义域交集内取得最大值时的图像缩放因子,该缩放因子即为结果。

表 5 不同分辨率配置下 FDR , $Power$ 与 $Memused$ 关于 scaling factor 的函数关系式

Table 5 Functional relation of FDR , $Power$ and $Memused$ on scaling factor with different resolution configurations

Normalized resolution	Functional relation		
	FDR	$Power$ consumption	$Memused$
240	$y = -19.8026x + 79.210$	$y = -5.824x + 64.220$	$y = -0.0414x^2 + 0.229x + 10.358$
360	$y = -29.941x + 129.745$	$y = -7.846x + 74.080$	$y = -0.0186x^2 + 0.0394x + 9.622$
480	$y = 7.002x^2 - 71.658x + 177.520$	$y = -9.403x + 81.614$	$y = 0.1893x^2 - 1.5987x + 13.034$
540	$y = -5.251x^2 + 1.979x + 108.474$	$y = -10.230x + 86.311$	$y = 0.3186x^2 - 2.6414x + 15.268$
640	$y = -10.295x^2 + 36.26x + 61.210$	$y = -11.123x + 93.361$	$y = 0.8121x^2 - 6.3838x + 22.25$
720	$y = -13.468x^2 + 59.746x + 26.118$	$y = -11.729x + 97.550$	$y = 2.0236x^2 - 15.5344x + 38.618$
768	$y = -17.703x^2 + 97.71x - 24.085$	$y = -11.886x + 99.621$	$y = 2.64x^2 - 20.25x + 47.31$
960	$y = -12.206x^2 + 80.03x - 23.658$	$y = -13.533x + 114.199$	$y = 5.74x^2 - 43.302x + 87.474$
1024	$y = 13.039x + 47.508$	$y = -13.123x + 115.126$	$y = 7.1079x^2 - 53.348x + 105.154$
1200	$y = 13.914x + 40.994$	$y = -12.173x + 118.677$	$y = 9.6394x^2 - 72.7207x + 140.826$
1440	$y = 14.911x + 37.258$	$y = -9.297x + 116.655$	$y = 8.8043x^2 - 69.0917x + 143.102$

(3)离线部分的第三步和第四步。根据第二步的结果,以系统性能指标的阈值范围为视频处理性能指标的约束条件,

计算出每种分辨率对应的最佳图像缩放因子,建立图像缩放因子与分辨率之间的对应关系,如表 6 所列。

表 6 不同分辨率配置下 FDR 取最大值时的缩放因子

Table 6 Scaling factor at the maximum FDR with different resolution configurations

Normalized resolution	240	360	480	540	640	720	768	960	1 024	1 200	1 440
scaling factor	1.00	1.00	1.00	2.00	1.76	2.22	2.76	3.28	5.00	5.00	5.00

(4)在线部分。修改视频处理预处理阶段的代码,根据表 6 建立图像缩放因子与分辨率之间的对应关系。

算法 1 表示资源感知的图像自适应缩放算法。算法 1 的第 5 行—第 19 行表示根据图像自适应缩放方案设置图像缩放因子。函数 $networkAwareness()$ 和 $powerAwareness()$ 分别表示基于网络感知和功耗感知的动态图像缩放因子设置方法(本文只给出思想,具体内容请参考第 2.1 节),两种方法均属于在线方案,具有实时性。在实时视频处理系统中,由于网络资源不稳定,边缘设备的电能也具有不确定性。如果由于信号干扰导致系统大量丢帧,则 $networkAwareness()$ 会返回较小的图像缩放因子,从而尽可能提升视频分析的性能。如果系统电能呈偏低趋势,则 $powerAwareness()$ 会返回较大的图像缩放因子,从而节约系统电能。由上述分析可知,网络感

知的缩放因子调整是为了在系统丢帧的情况下尽可能提升视频分析的性能,而功耗感知的缩放因子调整是为了在系统电能资源匮乏的情况下节约系统资源。因此,后者的优先级更高(算法 1 中的第 24 行,功耗感知的结果放在最后进行比较)。

算法 1 资源感知的图像自适应缩放算法

```

input: imag
output: sf // scaling factor
1. function getScalingFactor(imag)
2.   row ← imag.rows // Get image horizontal pixels
3.   col ← imag.cols // Get image vertical pixels
4.   P ← √(row × col) // Normalized resolution
5.   if P ≤ 480 then

```

```

6.     sf←-1.0
7.   else if P≤540 then
8.     sf←-2.0
9.   else if P≤640 then
10.    sf←-1.76
11.  else if P≤720 then
12.    sf←-2.22
13.  else if P≤768 then
14.    sf←-2.76
15.  else if P≤960 then
16.    sf←-3.28
17.  else
18.    sf←-5.0
19.  end if
20.  sf_networkAwareness←networkAwareness()
21.  sf_powerAwareness←powerAwareness()
22.  if sf≥sf_networkAwareness then
23.    sf←sf_networkAwareness
24.  else if sf≤sf_powerAwareness then
25.    sf←sf_powerAwareness
26.  end if
27.  return sf
28. end function

```

4 实验与性能分析

本文研究了视频分辨率的多样性对预处理阶段图像缩小倍数(缩放因子)的影响,提出了关于图像缩放因子的动态设置方案,本节对该方案进行了实验验证和性能分析。

本实验给出的性能指标有人脸检测率(Face Detection Rate, FDR)、帧接收率(Frame Reception Rate, FRR)、系统功耗(Power)以及系统内存使用率(Memory Usage, Memused)。

4.1 实验环境

为了验证真实场景下图像缩放因子动态设置方案的有效性,本实验使用树莓派作为边缘节点、台式主机作为边缘服务器搭建了实时视频流处理的边缘计算平台。树莓派型号为 Raspberry Pi 3 Model B+, 摄像头为 800 万物理像素。台式机的 CPU 配置为 Intel (R) Core (TM) i7-7700CPU @ 3.60GHz(4core), 内存 16 GB, 其中虚拟机为 8 核, 内存为 1GB。此外,使用万用表 UT71E 测量服务器功耗。软件环境主要为:边缘节点端的视频捕获工具 Rspivid 和转码工具 VLC, 边缘服务器端用于人脸检测的函数库 OpenCV 3.1.0。

实验搭建的实时视频流处理边缘计算平台的实物连接拓扑结构如图 7 所示。边缘节点为 3 个树莓派组成的集群, 边缘服务器是台式机中宿主机环境为 64 位 Windows10 的 Ubuntu 虚拟机, 系统架构是基于 Linux 的高并发模型, 可以同时为多路边缘节点提供实时视频流处理服务。实验网络环境为有线连接。边缘节点和边缘服务器间的视频流传输示意图如图 8 所示。在实验过程中, 边缘节点首先主动与服务器建立连接, 边缘节点端 Rspivid 将从摄像头捕获的视频输入 VLC 转码成 H.264 视频流, 以码流格式为 TS (Transport Stream) 的形式经压缩打包后输出到自定义端口, 边缘服务器

通过访问边缘节点的地址和指定端口来接受实时视频流, 然后进行解码、人脸检测、结果分析等操作, 直到边缘节点停止推流。

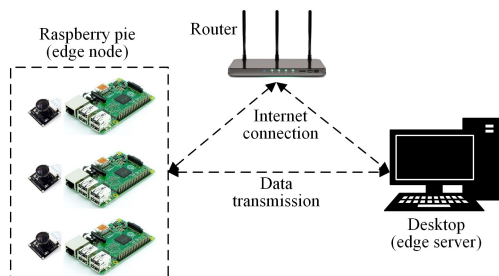


图 7 节点之间的连接拓扑图

Fig. 7 Connection topology between nodes

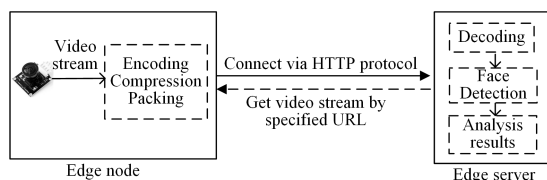


图 8 视频流传输示意图

Fig. 8 Diagram of video stream transmission

实验环境光线稳定, 仅考虑一个边缘节点推流至边缘服务器进行视频流处理, 目标物距离摄像头正前方 1m。对于测试帧接收率这一性能指标的实验, 边缘服务器端应用程序使用的是无帧缓冲队列系统(原因是帧缓冲队列系统几乎不存在丢帧问题), 对于测试其他性能指标的实验, 则使用的是有帧缓冲队列系统。

4.3 结果分析

本文提出的动态图像缩放因子设置方案的实验结果(4 个性能指标随着视频分辨率的提高而变化的情况)如表 7 所列。分析表中的数据可知, 所有分辨率视频的帧接收率均是 100%, 人脸检测率接近 100%。相比表 1 中的数据, 所有分辨率的系统性能指标分布呈现稳定且偏低趋势。由上述分析可知, 本文提出的图像缩放因子的动态设置方法是可行的, 即在系统性能允许的前提下, 11 种分辨率的视频数据都保持了较好的帧接收率和人脸检测率。

表 7 不同视频分辨率配置下的 4 个性能指标

Table 7 Four performance indicators with different video resolution configurations

Normalized resolution	Power/W	Memused/%	FDR/%	FRR/%
240	63.11	10.72	99.21	100
360	77.98	9.60	98.82	100
480	89.36	11.67	98.83	100
540	59.19	11.07	100	100
640	73.83	10.76	99.21	100
720	66.13	11.55	100	100
768	59.85	10.22	99.80	100
960	65.33	11.87	99.80	100
1024	52.49	12.28	99.61	100
1200	61.84	13.13	100	100
1440	69.40	14.24	100	100

结束语 本文主要研究视频参数的多样性对视频处理中

预处理阶段的影响。预处理阶段会将图像适当缩小再进行处理,并将图像缩小倍数记作图像缩放因子。对于不同分辨率的视频,应该设置不同的图像缩放因子,从而保障各自最佳的视频处理性能。在视频预处理阶段,与直接成倍降低分辨率的方法相比,图像缩放因子动态设置方案可以在保证视频处理性能的基础上减少系统功耗和内存使用率,从而提高视频处理效率和能力。

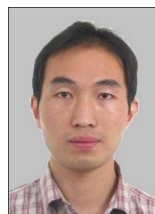
实验部分验证了图像缩放因子动态设置方案的可行性。实验结果表明,对于所有分辨率的视频,该方案均可在系统性能允许的前提下保障较优的视频处理性能。但是,本文实验中涉及到的性能指标较少,实验中图像缩放因子的划分间隔较大,范围也可进一步扩展。因此,本文涉及的相关实验规模均可扩展。此外,视频参数都是间隔性取得的,涉及到的视频参数具有局限性,在未来的工作中有待进一步研究。

参 考 文 献

- [1] ZHOU B, DASTJERDI A V, CALHEIROS R N, et al. mCloud: A context-aware offloading framework for heterogeneous mobile cloud [J]. *IEEE Transactions on Services Computing*, 2017, 10(5):797-810.
- [2] LIU W, HUANG Y C, DU W, et al. Resource-Constrained Serial Task Offload Strategy in Mobile Edge Computing [J]. *Journal of Software*, 2020, 31(6):1889-1908.
- [3] MA H R, CHEN X, ZHOU Z, et al. Dynamic Task Offloading for Mobile Edge Computing with Green Energy [J]. *Journal of Computer Research and Development*, 2020, 31(6):1889-1908.
- [4] LIU Y, XU C, ZHAN Y, et al. Incentive mechanism for computation offloading using edge computing: a Stackelberg game approach [J]. *Computer Networks*, 2017, 129:399-409.
- [5] ME E, TODD T D, ZHAO D, et al. Energy aware offloading for competing users on a shared communication channel [J]. *IEEE Transactions on Mobile Computing*, 2017, 16(1):87-96.
- [6] CHEN X. Decentralized computation offloading game for mobile cloud computing [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2015, 26(4):974-983.
- [7] CHEN X, JIAO L, LI W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing [J]. *IEEE/ACM Transactions on Networking*, 2016(5):2795-2808.
- [8] JIA M, CAO J, YANG L. Heuristic offloading of concurrent tasks for computation-intensive applications in mobile cloud computing [C]// *Proceedings of the IEEE Conference on Computer Communications Workshops*. 2014:352-357.
- [9] YU B W, PU L J, XIE Y L, et al. Joint Task Offloading and Base Station Association in Mobile Edge Computing [J]. *Journal of Computer Research and Development*, 2018, 55(3):537-550.
- [10] BILAL K, ERBAD A. Edge computing for interactive media and video streaming [C]// *Proceedings of the Second International Conference on Fog and Mobile Edge Computing (FMEC)*. 2017:68-73.
- [11] LONG C, CAO Y, JIANG T, et al. Edge computing framework for cooperative video processing in multimedia IoT systems [J]. *IEEE Transactions on Multimedia*, 2018, 20(5):1126-1139.
- [12] BAILAS C, MARSDEN M, ZHANG D, et al. Performance of video processing at the edge for crowd-monitoring applications [C]// *Proceedings of the 4th World Forum on Internet of Things (WF-IoT)*. 2018:482-487.
- [13] TRAN T X, PANDEY P, HAJISAMI A, et al. Collaborative multi-bitrate video caching and processing in mobile-edge computing networks [C]// *Proceedings of the 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2017:165-172.
- [14] JIANG W, CHEN G, YANG F, et al. Self-adaptive Image Scaling Algorithm Based on Sobel Operator [J]. *Computer Engineering*, 2010, 36(7):214-216.
- [15] WANG D, ZHAO H W, DAI Y, et al. A face detection acceleration algorithm based on regression [J]. *Journal of Chongqing University of Post and Telecommunications (Natural Science Edition)*, 2019, 31(4):550-555.
- [16] SHI W S, SUN H, CHEN Y M. Prospect of new video surveillance system based on edge computing [J]. *China Academic Journal Electronic Publishing House*, 2018, 35(12):60-63.
- [17] GE C, BAI G W, SHEN H, et al. Edge computing-based video surveillance framework [J]. *Computer Engineering and Design*, 2019, 40(1):32-39.
- [18] MANGIANTE S, KLAS G, NAVON A, et al. VR is on the edge: How to deliver 360 videos in mobile networks [C]// *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. 2017:30-35.
- [19] HU P, NING H, QIU T, et al. Fog computing based face identification and resolution scheme in Internet of Things [J]. *IEEE Transactions on Industrial Informatics*, 2017, 13(4):1910-1920.
- [20] ZHANG Q, ZHANG Q, SHI W, et al. Distributed collaborative execution on the edges and its application to AMBER alerts [J]. *IEEE Internet of Things Journal*, 2018, 5(5):3580-3593.
- [21] ZHANG Z, ZHANG X Q, ZUO D C, et al. Research on Target Tracking Application Deployment Strategy for Edge Computing [J]. *Journal of Software*, 2020, 31(9):2691-2708.
- [22] SUN H, YU Y, SHA K, et al. mVideo: Edge Computing Based Mobile Video Processing Systems [J]. *IEEE Access*, 2019, 8:11615-11623.
- [23] LONG C C, CAO Y, JIANG T, et al. Edge Computing Framework for Cooperative Video Processing in Multimedia IoT Systems [J]. *IEEE Transactions on Multimedia*, 2017, 20(5):1126-1139.



ZHANG Kai-qiang, born in 1997, post-graduate, is a student member of China Computer Federation. His main research interests include edge computing and system optimization.



JIANG Cong-feng, born in 1980, Ph.D. professor, is a member of China Computer Federation. His main research interests include edge computing, system optimization, performance evaluation and distributed system benchmarking.