计算机科学
COMPUTER SCIENCE

视觉情境感知驱动的虚拟机器人交互系统

刘宇博, 郭斌, 马可, 邱晨, 刘思聪

引用本文

刘宇博, 郭斌, 马可, 邱晨, 刘思聪. 视觉情境感知驱动的虚拟机器人交互系统[J]. 计算机科学, 2023, 50(9): 260-268.

LIU Yubo, GUO Bin, MA Ke, QIU Chen, LIU Sicong. Design of Visual Context-driven Interactive Bot System [J]. Computer Science, 2023, 50(9): 260-268.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230200167

视觉情境感知驱动的虚拟机器人交互系统

刘宇博　郭　斌　马　可　邱　晨　刘思聪
西北工业大学计算机学院　西安 710129
(redcontritio@qq.com)

摘　要　虚拟机器人是能与人交互的智能软件,通常具有实时性、交互性等特点。文中以视觉情境感知驱动的虚拟机器人为主题,从轻量级目标检测模型及压缩、实时关键帧提取、系统优化和交互策略4个方面展开探究,在边缘的资源受限平台上构建强实时性、高交互性、高度可扩展的虚拟机器人系统。具体而言,在轻量级目标检测模型及压缩方面,首先探究不同主干网络下SSD模型的性能与精度,随后对基于VGG16网络的SSD模型进行int8量化与剪枝,在精度损失不超过0.1%的前提下,帧率比原模型提高187%。在实时关键帧提取方面,使用边缘特征强度和HOG特征进行视频流预筛选,降低系统压力,等效减少90%的推理时延。在系统优化方面,采用微服务化降低冷启动时延约98%。在交互策略方面,使用含计时器的状态机对情境进行建模以实现情境驱动,并采用语音形式完成人机交互的输出。

关键词:资源受限;轻量级模型;模型压缩;目标检测;情境驱动
中图法分类号　TP391

Design of Visual Context-driven Interactive Bot System

LIU Yubo,GUO Bin,MA Ke,QIU Chen and LIU Sicong
School of Computer Science,Northwestern Polytechnical University,Xi'an 710129,China

Abstract　Bots are intelligent software that can interact with people,and usually have the characteristics of real-time and interactivity. This paper takes the bots driven by visual context awareness as the theme,and explores from four aspects:lightweight target detection model and compression,real-time key frame extraction,system optimization,and interaction strategy,and builds strong real-time on edge resource-constrained devices. A flexible,highly interactive and highly scalable bots system. Specifically,in terms of lightweight target detection models and compression,we first explore the performance and accuracy of different lightweight target detection models,and compress the SSD model based on the VGG16 network to find a suitable compression strategy. Compression on the latest SSD model can increase the frame rate by 187% compared with the original model,under the premise that the accuracy loss does not exceed 0.1%. In terms of real-time key frame extraction,the input video stream is prescreened to reduce system pressure,which is equivalent to reducing inference delay by 90%. In terms of system optimization,the use of microservices reduces the cold start delay by about 98%. In terms of interaction strategy,a state machine with timer is used to model the situation to achieve situation-driven,and the output of human-computer interaction is completed in the form of speech.

Keywords　Resource-constrained,Lightweight model,Model compression,Object detection,Context-driven

1　引言

机器人通常指能半自主或全自主工作的智能机器,其能够辅助或替代人类完成部分工作,强调其与现实环境的交互能力。虚拟机器人则略有不同,它并不强调与现实环境的交互,而更重视与人的交互,满足人的情感需求与信息获取需求。随着人工智能技术的不断发展,虚拟机器人已经被越来越多地应用在日常业务中。其中,虚拟机器人依赖各种人机交互形式,如音频、视频或通过图形界面进行文本输入输出等。而移动边缘设备等也以其便携性、交互性等特性成为虚拟机器人的理想部署平台。以树莓派部署平台为例,其通常设计多模态人机交互软硬件(如摄像头获取视频作为输入,采用音频作为输出方式)与用户进行交互。

在整个交互任务中,最基础和首要的就是对虚拟机器人所处的情境进行建模,让机器人对当前情境进行识别、区分,从而为后续机器人能在特定情境下实施特定行为做铺垫。

到稿日期:2023-02-22　返修日期:2023-07-01
基金项目:国家杰出青年科学基金(62025205);国家自然科学基金(62032020,61725205,62102317)
This work was supported by the National Science Fund for Distinguished Young Scholars(62025205) and National Natural Science Foundation of China(62032020,61725205,62102317).
通信作者:郭斌(guob@nwpu.edu.cn)

以视觉情境为例,机器人观察到的环境状况(如室内地图、用户手势)就是机器人需要建模的视觉情境,而根据应用需求所利用的具体技术包含目标检测、室内地图重构等视觉算法模型。在完成虚拟机器人所处情境的建模后,机器人即可根据"情境-行为"映射完成各种服务应用,如语音导览、智能家居等。

此外,上述虚拟机器人往往在便携的移动边缘平台上部署。然而,这些平台通常面临多进程造成的内存占用与长时间运行导致的电量损耗等问题。由于这些平台的存储和计算资源有限,并且具有动态复杂变化的特性,因此对虚拟机器人的部署模型、调度服务、交互框架等系统设计提出了轻量化和运行时自适应的需求。在这种情况下,传统静态的模型部署设计、服务应用定制往往会造成各种资源浪费,导致系统难以做到长时间稳定部署,设备寿命缩短,维护难度增大。

综上,本文基于边缘移动平台,以视觉情境感知驱动的虚拟机器人部署为具体目标,展开轻量自适应、持续高效的交互系统设计。具体地,在视觉情境建模上,结合紧凑模型设计和模型压缩技术部署轻量级视觉识别模型;在交互框架上,采用视频关键帧选取、微服务化系统优化和基于状态机的情境驱动策略等,完善整个虚拟机器人交互系统的搭建。本文旨在探索和提供移动边缘平台上虚拟交互机器人系统的轻量自适应通用开发方案,促进模型压缩、系统优化等主流技术在物联网实际应用中的落地实现和结合。

本文主要根据上述两部分内容模块进行划分和介绍。

1)结合紧凑模型设计和模型压缩技术,设计视觉情境驱动的虚拟机器人目标检测模型,探索交互系统中的轻量化机理,并针对不同设备算力和不同模型开销(如计算量、模型尺寸)等,进行不同压缩规格的尝试,获得了一系列适用的模型变体。

2)交互系统实现,包括系统性能调优与虚拟机器人的情境驱动交互策略。其中,系统性能调优包括通过实时关键帧提取改进实时性,使用微服务机制缩短系统冷启动时间,通过网络接口通信保障可扩展性。情境驱动阶段则是在目标检测结果的基础上,基于状态机方式进行情境触发,随后在预置情境知识的帮助下进行响应,以音频的形式与用户进行交互。

2 相关技术

2.1 虚拟机器人技术

虚拟机器人[1]是基于自然语言处理的智能会话系统[2],融合了多种人工智能技术。其特点是仅依赖于软件,而对硬件设备的依赖度较低,仅需要基本的输入输出接口,更多的是用于和人的交互方面。本文的研究目的即是开发基于视觉的虚拟机器人系统,并使其具备与用户的交互能力。

2.2 目标检测技术

目标检测作为计算机视觉领域的一项重要任务,其目标是提供图像中某类视觉对象(如人、动物或汽车)的具体位置。其模型可以分为以 RCNN 模型为代表的二阶段模型和以 SSD 模型、YOLO 模型为代表的单阶段模型。由于单阶段模型所需阶段数量较少,因此其检测速度通常比二阶段的目标检测算法更快,但代价是精度略低[3]。本文选用 SSD 模型作为后续研究的基础。

2.3 轻量化模型设计

轻量化模型设计旨在在不损失精度的前提下,减少网络参数数量,从而使模型的参数更少,算力消耗更低,更容易部署在资源受限的平台上。

现有的 3 种典型的轻量化模型分别是 SqueezeNet[4],MobileNet[5]和 ShuffleNet[6]。SqueezeNet 开创性地提出了 Fire[4]模块,包含 squeeze 层与 expand 层,squeeze 层即 1×1 的卷积层,而 expand 层则是 1×1 与 3×3 的卷积层,两者配合以减小模型体积。MobileNet 则是采用深度可分离卷积达到减少模型参数数量与加快计算速度的目的,其将传统卷积操作拆分为深度卷积与逐点卷积。ShuffleNet 使用分组卷积[6]和通道卷积[6]代替传统卷积,以减少参数量,从而加快速度,其改进点在于使用通道混洗来代替 1×1 卷积。

2.4 深度模型压缩技术

复杂度越高的模型往往有着更高的精确度,但其巨大的内存占用、高额的计算资源消耗限制了其在各种硬件平台上的应用。因此,针对复杂模型的模型压缩随之产生。模型压缩最常用的方法是参数修剪[7-8]和量化[7-10],其主要思路是提出一种参数重要性的计算方式,随后将不重要的连接和滤波器进行裁剪,以减少模型的冗余。

2.5 视频关键帧技术

视频帧通常存在大量的冗余,通过视频关键帧提取[11],能够从平均价值低、数量大的图像帧序列中提取出平均价值高、数量少的关键帧,从而降低视频信息分析的输入量,减小处理压力,同时,高质量的图像也有助于保证分析的有效性。

基于传统算法的高性能视频关键帧提取的常用方式包括随机抽样提取、基于图像特征[12]、基于运动分析[13]等。由于随机抽样方式的效果明显较差,因此通常仅将其作为对比项,本文主要使用基于图像特征的关键帧提取。

图像特征通常包括颜色特征和梯度特征,前者包括颜色直方图、颜色矩等;后者则基于颜色梯度进行计算,包括方向梯度直方图和图像边缘。边缘是像素快速变化的区域,通常基于图像梯度进行计算。在二维的灰度图中,最常用的边缘检测算子有 Sobel 算子[14]、Canny 算子[15]等。Sobel 算子能够较快地对边缘进行粗糙检测,而 Canny 算子基于高斯核进行梯度计算,能够获得更精细的边缘。

3 交互系统轻量级目标检测模型设计

作为虚拟机器人视觉处理的关键部分,轻量化目标检测模型需要针对平台性能进行设计,以提高虚拟机器人的实时性,从而改善交互体验。

在目标检测模型的设计过程中,模型轻量化主要包括对卷积的加速和修改网络结构。根据卷积操作的数学原理可知,对卷积操作的优化加速存在理论下界,但通过重新设计计算的顺序,就可以使运算量大幅减小。

本章将先对目标检测的主流模型进行介绍,结合本文的研究内容选择待研究模型,并对其不同变体进行性能与准确率的测试。随后针对不同的深度模型压缩方法,对基于 VGG16 网络的 SSD 模型进行处理,并将未压缩模型与经过

各种压缩方式处理过的模型进行对比。最后基于本章的设计与尝试进行实验，实验结果验证了本文的研究成效。

3.1 模型轻量化机理验证

目标检测最常用的模型包括 RCNN 系列、SSD 系列与 YOLO 系列，本节首先对这几种模型分别进行介绍，随后结合相关工作进行不同主干网络的 SSD 模型的参数修剪实验，探究适合受限环境的轻量化模型设计方案。

RCNN 系列是最早将卷积神经网络引入目标检测领域的网络，其作为典型的两阶段模型，首先生成候选区域并将候选区域缩放到相同的尺寸，随后将这些相同尺寸的图像送入 CNN 网络中进行特征提取，并基于提取到的特征使用 SVM 进行分类，最后对所有候选区域进行非极大值抑制及边界框回归以获得最终的目标检测结果。尽管 RCNN 模型精度较高，但其模型大小与推理时延都不尽人意。

YOLO 是典型的单阶段网络，即在一个网络中完成对特征的提取与分类、边界框回归操作。最常用的是第三代模型，即 YOLO v3，该模型使用 3 种不同的网格划分尺度，能同时提取原始图像的不同尺度特征。此外，该模型采用多标签分类以从同一个物体检测出多个标签。

SSD 系列模型的思路是在多层卷积网络中，将各层的输出共同作为特征，从而获得不同尺度的特征。最后，基于此前获取的不同尺度特征集合进行检测，用不同尺度特征对不同尺度目标进行识别。

由于 RCNN 推理时延显著高于其余两者，且 YOLO 模型的网络结构固定，因此本节首先简要介绍模型轻量化机理，随后基于 SSD 模型对不同主干网络进行测量与验证。

3.1.1 模型轻量化机理

本文涉及的网络轻量化机理主要包括瓶颈层(Bottle-Neck Layer)、深度可分离卷积(Depthwise Separable Convolution)与倒残差块(Inverted Residual Block)。

1)瓶颈层

瓶颈层是 ResNet 块[16]的关键结构，其最早提出于 Inception 网络[17]中。为了解决大量卷积操作带来的运算成本，使用 1×1 卷积以减少通道数量，降低运算成本，其结构如图 1 所示。以参数为 3×3×256 的传统卷积为例，将原有的卷积更换为含有瓶颈层的 ResNet 块，能使参数量减少 61%，从而降低网络的训练与推理成本。

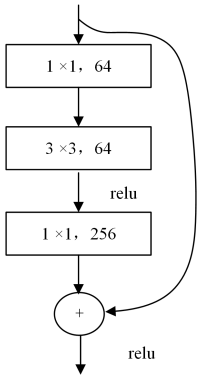


图 1 瓶颈层结构

Fig. 1 Structure of bottleneck layer

2)深度可分离卷积[5]

深度可分离卷积最早出现在 MobileNet 网络结构中，其采用了两段式的卷积运算方式，如图 2 所示，采用两次卷积代替原来的跨通道卷积。

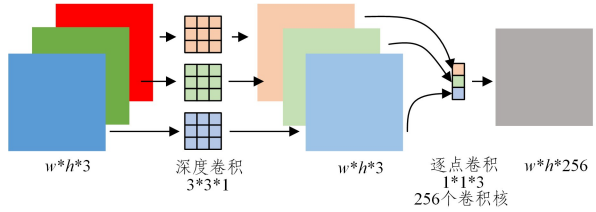


图 2 深度可分离卷积

Fig. 2 Depthwise separable convolution

以参数为 3×3×256 的传统卷积为例，将传统卷积更换为深度可分离卷积，能使参数量减少 65.5%。

3)倒残差块

倒残差块最早提出于 MobileNet v2 网络[18]中，其结构如图 3 所示。为了尽可能保留信息，倒残差块对残差块进行了 3 处改进，首先采用 1×1 卷积层进行特征的扩张，随后使用深度可分离卷积进行中间的卷积操作，最后移除输出端瓶颈层的 relu 操作，使其保留线性，从而避免信息的损失。

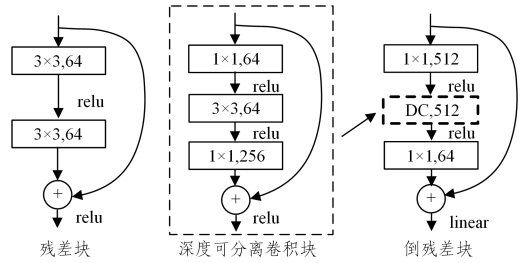


图 3 倒残差块结构示意图

Fig. 3 Structure of inverted residual block

倒残差块解决了深度可分离卷积带来的精度损失问题，在性能与精度间取得了较好的均衡。

3.1.2 不同主干网络的对比

SSD 模型通常使用以 VGG16 或 ResNet 作为特征提取的主干网络，前者是 SSD 首次发布时的网络结构，后者则通常被称为 RetinaNet。SSD 模型的优点在于运算速度快，且主干网络只用于提取特征，因此结构耦合度较低。

SSD 模型所用的 VGG16 主干网络由 5 个块构成，块之间使用最大池化层进行处理。由于 VGG16 主要用于图像分类，因此最后使用全连接与 softmax 层对分类进行预测。要将 VGG16 应用于目标检测网络主干，只需要提取特征的能力，因此应用时主要进行了 3 处改动：1)为了在主干网络的最后进行特征提取，移除了原有的全连接层 FC6 与 FC7，改用卷积 conv6 与 conv7，移除了 FC8 层；2)将池化层的核从 2×2 改为 3×3，步长改为 1；3)由于已经移除了全连接层，因此不再需要为避免全连接层过拟合的 dropout 层。

易于验证，以 VGG16 为主干网络的 SSD 模型中，95%以上的运算都发生在主干网络中。因此对其更换更轻量级的主干网络，能够大幅减少 SSD 模型的总运算量。

3.2 深度模型压缩

3.2.1 剪枝

VGG16 网络的参数远多于 ResNet50 与 MobileNet，但精度相近，因此 VGG16 网络有着更高的冗余度，在本节中将对以 VGG16 为主干的 SSD 模型进行参数修剪，并测试其在不同稀疏度下的精度表现。

表 1 列出了 SSD 模型中的 VGG16 结构，其前 4 个块参数量只占总体的 8.47%，如果对所有层进行相同比例的稀疏，将会严重影响前 4 个块的精度。更合理的方式是对全局权重进行统一衡量，对参数较多、冗余较大的层进行更强的修剪，对参数较少的层则保留更多。

表 1 SSD 模型中的 VGG16 结构

Table 1 Structure of VGG16 in SSD

模块名	层编号	输入特征	输出特征	参数量	计算量(MFLOPS)
block1	vgg.0	3×300×300	64×300×300	1792	296.63
	vgg.2	64×300×300	64×300×300	36928	6328.13
block2	vgg.5	64×150×150	128×150×150	73856	3164.06
	vgg.7	128×150×150	128×150×150	147584	6328.13
block3	vgg.10	128×75×75	256×75×75	295168	3164.06
	vgg.12	256×75×75	256×75×75	590080	6328.13
	vgg.14	256×75×75	256×75×75	590080	6328.13
block4	vgg.17	256×38×38	512×38×38	1180160	3249.00
	vgg.19	512×38×38	512×38×38	2359808	6498.00
	vgg.21	512×38×38	512×38×38	2359808	6498.00
block5	vgg.24	512×19×19	512×19×19	2359808	1624.50
	vgg.26	512×19×19	512×19×19	2359808	1624.50
	vgg.28	512×19×19	512×19×19	2359808	1624.50
conv6	vgg.31	512×19×19	1024×19×19	4719616	3249.00
conv7	vgg.33	1024×19×19	1024×19×19	1049600	722.00

3.2.2 量化

在现有的网络结构中，网络的参数类型均为 float32 类型，尽管可以精确表达过程中的运算，但在实际运算过程中，由于网络权重通常在较小的特定范围内，因此可以用更稀疏的值进行表示，即量化操作，将连续的权值进一步稀疏化与离散化。

最常用的量化方式是使用 int8 进行量化，即使用 int8 作为实际权重的一种映射表达方式。当权重的值不够稀疏时，进行 int8 量化可能会大幅降低精度，因此，作为一种折中方案，我们提出了 float16 量化。降低浮点数的位长对精度的影响微乎其微，但其能使模型的尺寸减半。特别地，当设备指令集不支持 float16 时，也可以自动将其转换成 float32 类型以保证运算的正确。

4 面向资源受限移动平台的虚拟机器人交互系统实现

在实际环境中，虚拟机器人面对复杂的资源状况，面对算力与内存都相对较差的树莓派平台，如何让整个虚拟机器人系统在满足平台资源限制的情况下，尽可能地达到性能与精度的最佳均衡点，是本文面对并亟待解决的关键问题之一。

使用的模型需要根据具体的资源场景进行选取，以便系统能够更稳定地运行。本章将基于目标检测模型进行封装，从系统角度对外围设计进行介绍，并以部署指标为导向进行系统调优，从而实现完整的可交互虚拟机器人。图 4 展示了系统优化对良好用户体验的重要性，并给出了目标检测模型的优化路径与作用方式。

除了模型对系统的性能会产生影响外，系统仍有其他的性能指标，如冷启动速度等。在后面的第 4.3 节中，将阐述该性能指标的改进方式，并在第 4.4 节中展示改进的效果，从而对本文的系统设计工作成效进行验证。图 5 给出了设计后的系统架构。

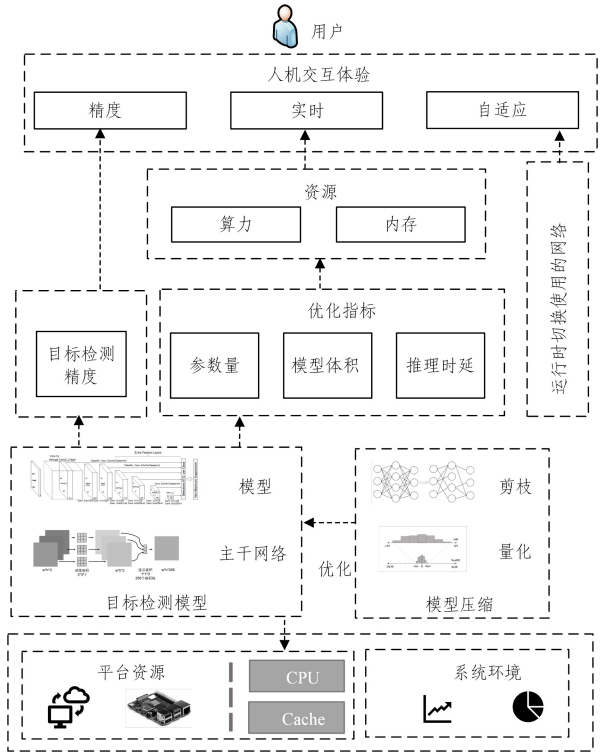


图 4 系统优化目标检测模型以提升交互体验

Fig. 4 System optimization to improve performance

由于本文所用的树莓派设备上不存在显示屏，因此采用音频作为唯一的输出方式，系统将语音送入音频输出接口，从而实现与用户的交互。在这个过程中，首先需要考虑如何针对前述流程的目标检测结果，并在此前反馈的时间序列基础上，设计适当的反馈内容，使之表现出智能性。除了交互内容的设计，交互的频率、语音生成方式等也需要进行设计，从而提高智能性与交互性。

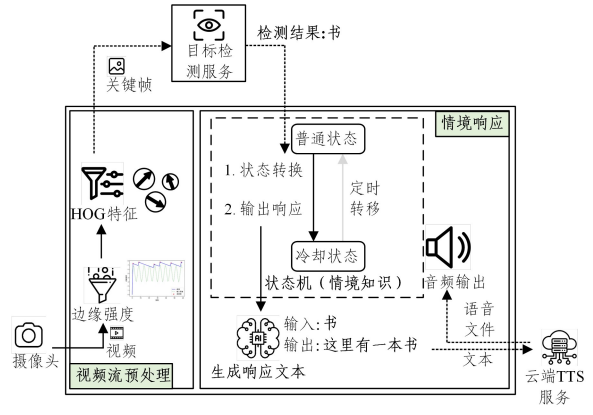


图 5 系统架构示意图

Fig. 5 System architecture diagram

4.1 基于多重特征的实时关键帧提取

主流的视频关键帧技术[19]往往是基于已有的视频离线进行，而人机交互系统对实时性有着更高的需求。因此，许多成熟的关键帧检测算法不能直接应用。此外，由于聚类依赖完整的片段结构，只能针对完整视频，因此实时性较差。

4.1.1 基于边缘的动态阈值法

本小节将细致介绍基于边缘特征的动态阈值法设计流程，并对其有效性进行分析。

在传统图像处理中，图像边缘主要用于图像分割[20]。但在研究中，我们发现边缘特征也可以用来作为图像清晰度的衡量标准。由于本文进行边缘特征提取的目的是衡量图像质量，对边缘质量要求低，因此选择 Sobel 算子进行处理，采用边缘强度灰度图的全局均值作为全局边缘强度指标。对于相同的场景，图像内容越清晰，全局边缘强度均值越大；图像内容越模糊，全局边缘强度均值越小。该衡量方式存在一个缺陷，即全局边缘强度均值的绝对值与场景内容相关。场景内容越复杂，全局边缘强度均值就越大。因此，需要采用一种自适应的策略对全局边缘强度均值进行处理。

本文选择的自适应策略是，首先定义阈值 s 与衰减率 d。每当一帧被选择时，就将阈值 s 更新到被选择帧的全局边缘强度均值。此外，为了避免阈值的单调提高，我们设置阈值会随时间基于衰减率 d 等比例衰减，从而保证关键帧的检出率。

4.1.2 基于 HOG 特征余弦距离的相似度阈值

在通过边缘特征初步筛选了清晰的帧后，程序可以获得一个稳定的低帧率图像流。但这个图像流仍然存在一定问题，即存在大量的冗余信息。如果能通过特征对图像相似度进行衡量，就能检测出其中的冗余信息。HOG 特征[21]，即方向梯度直方图特征，主要用于对图像的全局信息进行描述。在获得了图像的 HOG 特征后，由于 HOG 特征已经进行过归一化，因此可以认为其模长固定，使用向量夹角的余弦值能表示两个 HOG 特征的相似程度。在实际运算过程中，选择上一个被选取帧作为 HOG 特征对比的对象，与当前帧进行对比。

4.2 基于状态机的情境驱动策略

完成系统设计与系统性能分析后，就需要基于此前的流程结果对响应进行设计，使之体现出智能特性。

根据检测结果生成文本的过程有较为朴素的设计方式是直接指出在场景中发现的目标。但考虑到智能性，另一种更好的方式是基于过去一段时间的响应来生成蕴含上下文情境的文本。为了避免系统设计复杂度的过分膨胀，选择采用状态机的方式对生成的文本进行规则的预定义，通过恰当设计状态机的状态转换策略来改善系统效果。

本文以检测结果作为状态机的事件，也作为文本生成，即动作的参数。此外，考虑到文本生成密度不宜太高，需要对交互频率进行约束，原因是文本生成密度太高会导致响应密度过高，进而降低交互舒适度，因此本文设计了两种约束方式，一种是每次生成文本后进入待冷却状态，通过计时器触发冷却事件，使其回复到初始可用状态。该方式的优点在于易于实现，能够通过限制最高对话频率来一定程度地提高交互性。另一种方式则是每次生成文本时，同时进入对应状态的待冷却状态，以使对话频率与对话内容相关，便于针对不同情况决定交互策略。

4.3 交互系统优化

通过实时关键帧选取器，能保证帧的传入频率不会过高，并提取出高质量、低相似度的关键帧。通过关键帧选取后，本系统尝试对以 MobileNet 为主干网络的 SSD 模型进行目标检测工作，并结合具体资源状况，使用量化等技术对模型进行加速，从而提高系统的效率。

未压缩模型在树莓派设备上运行速度较快，在关键帧选取器运作时，可以保障未压缩模型的实时运行。但要使用未压缩模型，仍然存在两个问题，即冷启动速度与首次检测速度过慢。

模型加载时，需要从磁盘(内存卡)中读取模型，并加载到内存中对其结构进行解释。网络结构越复杂，读取所花费的时间就越多。例如，以 MobileNet 为主干网络的 SSD 模型，在冷启动过程中几乎会占用完整的 CPU，可以认为已经达到了树莓派平台的最佳性能。使用该模型与量化后的轻量级模型加入系统时，记录启动参数，如表 2 所列。

表 2 模型的启动性能

Table 2 Model startup performance

	冷启动用时/s	初次检测用时/s	后续检测用时/s	虚拟内存占用/MB
未压缩模型	173.9	12.90	0.47	1.31
压缩模型	5.4	0.37	0.30	0.06

从表 2 可以看到，最大的时间成本在于冷启动与初次检测，且未压缩模型的冷启动用时达到了压缩模型的 32 倍。以未压缩模型为例，初次检测用时达到了后续检测用时的 27 倍。

要降低该系统的冷启动成本，最常见的做法是构建微服务架构，即从系统中根据功能划分成若干小的服务模块，各个服务之间互相协调、互相配合。

根据对数据的处理流程，该系统可以分为 3 个阶段，即视频流输入与预筛选、轻量化目标检测以及最后的情境驱动输出响应。应用系统分析的方法，能发现轻量化目标检测阶段是成本最高的模块，因此需要对其进行抽离，使之成为独立的轻量化目标检测服务。图 6 给出了微服务优化前后的系统架构对比。

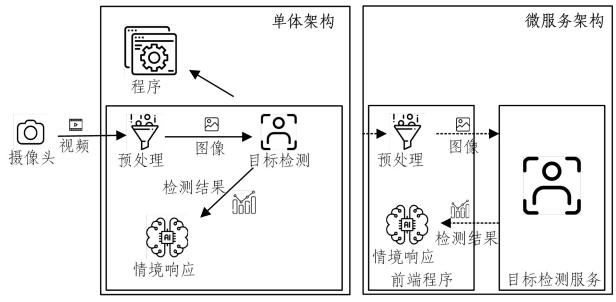


图 6 系统的单体架构与微服务架构

Fig. 6 Monolithic architecture and microservice architecture of system

本文在实践中采用 Docker 对轻量化目标检测模块进行

封装与部署工作，并将其加入开机自启动项目中，使得项目的前端冷启动用时控制在 2s 以内，减少项目冷启动时间约 98%。

5 实验验证

5.1 轻量化主干网络对比

本文使用 VOC2007 与 COCO 数据集作为目标检测效果的验证集，并使用 mAP，AP50 等作为精度的衡量指标。

在分别完成以 ResNet50 网络与 MobileNet v2 网络为主干的 SSD 模型训练后，在 COCO 数据集与 VOC2007 数据集上分别进行检验。检验结果如表 3 所列。

表 3 目标检测网络指标对比

Table 3 Comparison of target detection network metrics

主干网络	模型大小/MB	推理时延/ms	COCO	VOC
ResNet50	110.2	109	59.2	76.8
MobileNet v2	44.0	44	34.5	67.6

对比 ResNet50 网络与 MobileNet v2 网络的指标可以看出，模型推理时延与其精度呈负相关。ResNet50 尽管有着更高的精度，但推理时延也较高，不利于在资源受限的平台上进行推理。通过横向对比可以看出，COCO 数据集的目标检测难度更大，这是由于 VOC 数据集的待检测物体通常在画面中间且占据空间较大，而 COCO 数据集有着更多的类别，加大了中小尺度任务的占比，因此更容易出现漏检、交并比过低的现象。

在不同平台上分别对两者的实际推理时延进行测试，可以得到如图 7 所示的模型推理时延指标。

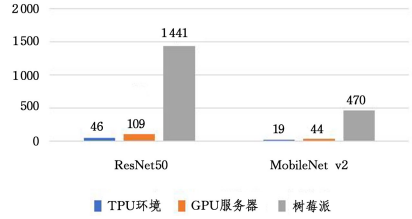


图 7 模型在不同平台的运行时延对比

Fig. 7 Comparison of runtime latency on different platforms

通过主干网络对比实验可以看出，对于本文的需求，以 MobileNet v2 为主干的 SSD 模型有着显著的优势。

5.2 模型压缩方式验证

在确定了使用的网络后，需要对不同的模型压缩方式进行分析以确定合适的方法。由于 VGG16 主干网络有较大的冗余度，因此采用 VGG16 进行模型的压缩方式验证。

首先采用不同的稀疏度对 VGG16 主干网络进行非结构化剪枝，采用 0%，20%，40%，60% 与 80% 的稀疏度进行剪枝，并且为了避免在部分区域发生过度的剪枝，本文使用全局剪枝进行处理，随后对压缩的模型权重体积、VOC 数据集检测精度等指标进行检验，结果如表 4 所列。从表 4 可以看出，当稀疏度不高于 60% 时，其对精度的影响较小。但对比 AP50 与 AP75 的下降速度可以看出，当稀疏度下降时，检测得到的边界框精度也不断降低，从而导致交并比阈值升高，精度严重降低。

表 4 VGG 主干网络的不同稀疏度与精度

Table 4 Different sparsity and accuracy of VGG backbone network

稀疏度/%	压缩权重体积/MB	AP50/%	AP75/%	AP95/%
0	93.01	89.67	83.95	3.15
20	79.62	89.44	83.51	2.88
40	63.80	87.78	80.76	1.30
60	46.26	67.36	52.02	0.36
80	27.70	0.62	0.01	0.00

在 VGG16 网络的不同稀疏度下，首先选择 float16 类型对 VGG16 主干网络进行量化，得到的结果如表 5 所列。

表 5 VGG16 的模型 float16 量化结果

Table 5 float16 quantification of VGG model

稀疏度/%	AP50/%	AP75/%	AP95/%
0	87.46	79.80	1.90
20	87.44	79.76	1.90
40	87.29	79.50	1.30
60	66.28	48.76	0.36
80	0.62	0.01	0.00

由表 5 可以看出，float16 对模型的精度影响较小，但由于平台没有对 float16 运算的原生支持，因此权重体积并不比原有的模型小，也不能带来运算速度上的提升。出于以上考虑，本文对 int8 量化位宽进行实验，采用无校准、等比例缩放两种量化后处理方式对模型进行处理，以 VOC 数据集作为测试集，在 GPU 服务器端对帧率、推理时延与精度进行测量。

观察表 6 可以发现，int8 位宽量化对模型精度影响较小，且模型性能提升较大。使用量化公式进行逐层的等比例缩放，虽然量化过程需要花费更多时间，但量化后模型具有更好的性能。

表 6 不同量化方式对比

Table 6 Comparison of different quantitative methods

量化类型	处理方式	帧率	推理时延/ms	AP50/%
未量化模型		13.98	72	83.66
int8	无	17.49	57	83.65
	等比例缩放	40.18	25	83.66

通过对 VGG16 网络进行非结构化剪枝可以发现，其参数存在高度的冗余，因此可以对其进行模型的压缩。本文通过对 VGG16 主干网络进行不同稀疏度的剪枝，验证了其冗余程度较高，并说明了剪枝对模型权重体积的优化效果。此后，本文首先通过不同稀疏度的 float16 量化，表明 float16 优化的有效性及其与稀疏度的关系。但由于 float16 在大多数平台上没有原生的指令支持，需要扩展至 float32 运算，不能体现其性能的改进，因此又使用 int8 进行模型量化，采用不同的后处理方式对模型进行调优，分别在未量化模型的基础上将帧率提高了 25% 和 187%，且保持精度基本一致，损失不超过 0.1%。

5.3 实时关键帧提取

基于 Sobel 算子获取全图的边缘特征后，对边缘特征的灰度图进行全局平均，从而获得单个标量浮点数。此外，还需要设置更高的更新阈值，使其能容忍后续一段时间的边缘强度逐渐上升，不至于在相邻帧内连续选取质量相近的帧。

在实际测试中，也选择与此相同的参数，即使用 0.9 作为衰减系数，选择 1.1 作为更新阈值，选择每秒 10 帧作为每秒帧数。在正常环境下，以步行速度进行测试，持续 1min，计算秒均帧选取数，通过多次实验来获取平均值。测量结果如表 7 所列。

表 7 动态阈值法的秒均帧选取数

Table 7 Average number of frames per second of dynamic threshold method

实验序号	选取帧数	秒均帧选取数
1	56	0.933
2	58	0.967
3	60	1.000
4	59	0.983
5	58	0.967
6	59	0.983
平均	58.3	0.972

实验测得，秒均帧选取数达到了 0.972，即约为每秒有 1 帧通过选取，能满足本文的需求。

由于 HOG 特征计算速度比边缘特征提取速度慢，因此本文中先使用基于边缘特征的动态阈值对图像进行粗筛选，随后在已经获得的较清晰图像中，用 HOG 特征来描述其相似度，避免了在相同画面中模糊的快速减弱导致较短时间内持续有帧被选取。

在实际环境中，单独使用 HOG 特征过滤，采用 0.05 作为阈值，选择 10 帧作为每秒帧数，在正常环境下以步行速度进行测试，持续 1 min，计算秒均帧选取率，通过多次实验计算平均值。测量结果如表 8 所列。

表 8 HOG 特征的秒均帧选取数

Table 8 Average number of frames per second of HOG

实验序号	选取帧数	秒均帧选取数
1	372	6.20
2	379	6.32
3	358	5.97
4	380	6.33
5	378	6.30
6	367	6.12
平均	372.3	6.21

实验测得，每秒平均帧选取率达到了 0.972，即约为每秒有 1 帧通过选取，对于本文需求而言过高，但能够有效地跳过画面内容相近的帧。

基于此前两种算法各自的效果与运行成本，本文选用两种关键帧选取并用的方式，以期同时继承两者的优点而不影响系统速度。

由于复合策略效果难以简单分析，因此在现实环境中，选取一段时间的边缘特征强度变化情况与 HOG 特征差异度进行分析。

从图 8 可以看到，在边缘强度上升的阶段，即使使用了略高的阈值，仍然会有 1 秒内选取 3 帧的情况发生。但由于相邻图像 HOG 特征相似，因此可以跳过处理。使用 0.05 作为 HOG 特征余弦距离阈值后，可以看到最终选取的帧数极为稀疏。多次测试发现，实际平均秒均帧选取量约为 1 帧。

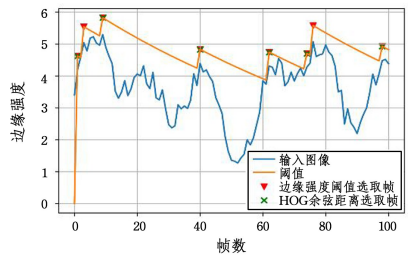


图 8 复合方法的选取效果

Fig. 8 Selection effectiveness of composite methods

将边缘特征强度动态阈值法与 HOG 特征相似度阈值进行综合，可以得到一个速度极快且效果较好的关键帧选取器，这也将作为此后工作流程的基础。

5.4 系统设计优化

本节对系统设计进行了性能优化，并简单阐述了基于状态机的情境驱动策略。

通过对系统进行设计，占用的总资源量没有发生变化，但通过微服务改造与将首次检测用时合并入微服务的冷启动阶段，大幅减少了系统的启动时间与平均检测用时。尽管系统仍然存在摄像头等资源的初始化，但该改进仍然大幅缩短了系统的冷启动时间，在未压缩模型基础上，改进前后的相关参数如表 9 所列。

表 9 系统优化前后参数对比

Table 9 Parameter comparison before and after system optimization

	冷启动时间/s	总内存占用/MB	首次检测用时/s
未优化系统	173.9	40.0	2.9
优化后系统	5.5	55.3	0.6

从表 9 可以看到，对系统进行分离与微服务化能够缩短系统的启动时间，但虚拟层的增加也导致系统内存占用增加。由于该系统的内存相对较为宽裕，因此选择以空间换时间的策略，将实时性作为最重要的指标对系统进行优化。

5.5 系统实现

通过将此前的设计与实现串联，就可以得到最终的交互系统。首先系统将以恒定帧率对实时视频进行采集，随后使用 4.1 节中提到的实时关键帧提取技术对清晰且有效的帧进行选取。图 9 展示了现实场景中系统关键帧提取阶段的数据流。

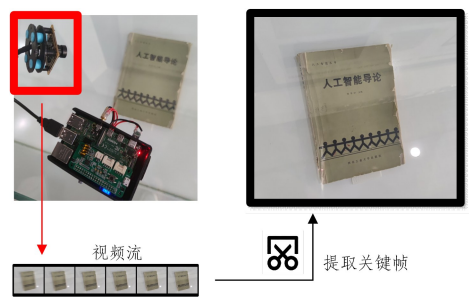


图 9 系统关键帧提取阶段

Fig. 9 Keyframe extraction phase of system

此后，在获取的关键帧图像基础上，将关键帧发送给目标检测微服务，以对画面中的目标进行检测。在目标检测服务

完成检测后，将图像中检测到的目标回传给前端程序。图 10 给出了该目标检测微服务的调用过程。

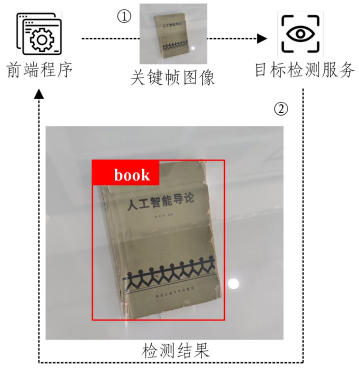


图 10 调用目标检测微服务

Fig. 10 Invoke object detection microservice

完成目标检测后，就可以基于目标检测的结果进行情境响应。由于视频流预处理和情境响应比较轻量级，因此均在前端程序中实现以减少工作量。

系统将首先基于目标检测结果，对情境感知的状态机进行驱动，使其发生状态的转换，以避免短时间内发生多次响应，破坏交互体验。状态机发生状态转移时，会产生副作用，首先基于目标检测结果产生对应的响应文本。在测试用例中，使用书作为目标检测结果，系统能够生成“这里有一本书”，最终调用音频输出接口进行输出。图 11 给出了系统基于目标检测结果产生情境响应，并以音频输出结果的流程。

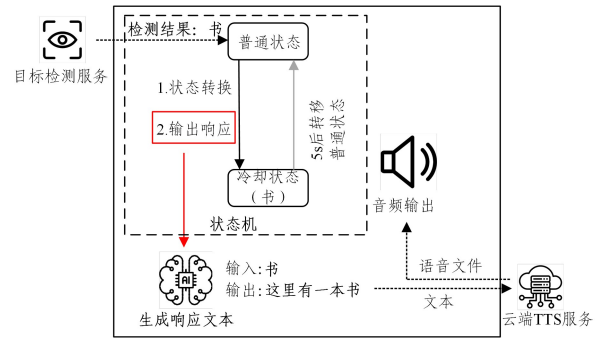


图 11 基于目标检测结果产生情境响应

Fig. 11 Generate contextual responses based on object detection results

现有的系统设计已经具备一定的情境感知能力，具有初步的可用性与交互性，并且有着充分的可扩展空间，包括对其状态机状态进行细化以提高情境区分能力，改进生成响应文本的算法，或通过云端生成响应文本，以提高最终交互语音的智能性等。

结束语 本文的研究内容以轻量化目标检测模型设计为核心，力求在效率、资源占用与精度上取得适合本文的均衡。1)本文尝试了多种不同的 SSD 模型变体，并分别对模型大小、推理时延、精度指标进行评估，并采用多种方式对模型进行压缩，横向对比不同压缩的有效性。以 VGG16 为主干网络的 SSD 模型压缩，使其帧率提高了 187%，且精度没有较大损失。2)针对项目需求，首先将系统根据功能划分模块，将目标检测模块以微服务的形式进行部署，对系统的冷启动时间等指标进行优化。此外，对输入的视频流进行预处理，将原有高频、平均质量低、信息冗余度大的图像流进行过滤，获取低频、平均质量高、信息冗余度小的稀疏图像流，降低了后端模型的处理压力，以满足实时交互的需求。最后，设计情境交互策略，采用状态机对情境进行描述，并根据前阶段获得的目标检测结果生成交互文本，以语音输出的方式表现系统的智能性与交互性。

但本文仍有很多地方有待改进。首先可以通过为系统添加与手机通信的功能，从而提供手机客户端以对系统状况进行监控，在手机端进行模型的运算工作，将手机作为系统输入输出界面等。此外，本文未能充分考虑与测试不同模型，如 CenterNet，YOLO 等。最后，在情境驱动策略方面，可以采用深度模型来处理情境的识别与转换，从而获得更适宜的情境感知与驱动效果。

参 考 文 献

[1] KLOPFENSTEIN L C，DELPRIORI S，MALATINI S，et al. The rise of bots：A survey of conversational interfaces，patterns，and paradigms[C]// Proceedings of the 2017 Conference on Designing Interactive Systems. 2017：555-565.

[2] ALBAYRAK N，ÖZDEMIR A，ZEYDAN E. An overview of artificial intelligence based chatbots and an example chatbot application[C]// 2018 26th Signal Processing and Communications Applications Conference(SIU). 2018：1-4.

[3] ADARSH P，RATHI P，KUMAR M. YOLO v3-Tiny：Object Detection and Recognition using one stage improved model [C]// 2020 6th International Conference on Advanced Computing and Communication Systems(ICACCS). 2020：687-694.

[4] IANDOLA F N，HAN S，MOSKEWICZ M W，et al. SqueezeNet：AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size[J]. arXiv：1602.07360，2016.

[5] HOWARD A G，ZHU M，CHEN B，et al. Mobilenets：Efficient convolutional neural networks for mobile vision applications[J]. arXiv：1704.04861，2017.

[6] ZHANG X，ZHOU X，LIN M，et al. Shufflenet：An extremely efficient convolutional neural network for mobile devices[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018：6848-6856.

[7] LI H，KADAV A，DURDANOVIC I，et al. Pruning filters for efficient convnets[J]. arXiv：1608.08710，2016.

[8] VANHOUCKE V，SENIOR A，MAO M Z. Improving the speed of neural networks on CPUs[C]// Deep Learning and Unsupervised Feature Learning Workshop(NIPS 2011). 2011.

[9] GUPTA S，AGRAWAL A，GOPALAKRISHNAN K，et al. Deep learning with limited numerical precision[C]// International Conference on Machine Learning. 2015：1737-1746.

[10] COURBARIAUX M，HUBARA I，SOUDRY D，et al. Binarized neural networks：Training deep neural networks with weights and activations constrained to +1 or −1[J]. arXiv：1602.02830，2016.

[11] SUJATHA C, MUDENAGUDI U. A study on keyframe extraction methods for video summary[C]// 2011 International Conference on Computational Intelligence and Communication Networks. 2011: 73-77.

[12] KELM P, SCHMIEDEKE S, SIKORA T. Feature-based video key frame extraction for low quality video sequences[C]// 2009 10th Workshop on Image Analysis for Multimedia Interactive Services. 2009: 25-28.

[13] LIU T, ZHANG H J, QI F. A novel video key-frame-extraction algorithm based on perceived motion energy model[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(10): 1006-1013.

[14] SOBEL I, FELDMAN G. A 3×3 isotropic gradient operator for image processing[J]. Pattern Classification and Scene Analysis, 1973: 271-272.

[15] CANNY J. A Computational Approach to Edge Detection [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, PAMI-8(6): 679-698.

[16] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

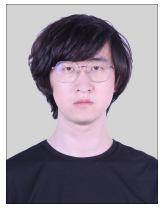
[17] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[18] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4510-4520.

[19] ZHUANG Y, RUI Y, HUANG T S, et al. Adaptive key frame extraction using unsupervised clustering[C]// Proceedings 1998 International Conference on Image Processing. icip98 (cat. no. 98cb36269). 1998: 866-870.

[20] HARALICK R M, SHAPIRO L G. Image segmentation techniques[J]. Computer Vision, Graphics, and Image Processing, 1985, 29(1): 100-132.

[21] DALAL N, AND TRIGGS B. Histograms of oriented gradients for human detection[C]// 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05). 2005: 886-893.



LIU Yubo, born in 2000, postgraduate, is a member of China Computer Federation. His main research interest is multi-modal QA.

GUO Bin, born in 1980, Ph. D, professor, doctoral supervisor. His main research interests include ubiquitous computing, mobile crowd sensing, big data intelligence and so on.

(责任编辑:杨雪敏)