doi:10.11835/j.issn.1000-582X.2021.07.012

基于 VPU 加速的嵌入式实时人脸检测系统设计与实现

闫 嘉[1a],张跃麟[1b],邓昌健[2]
(1. 西南大学 a.人工智能学院; b. 西塔学院 ,重庆 400715;
2. 电子科技大学 信息与通信工程学院,成都 611731)

摘要:智能设备高昂的设计费用和庞大的计算资源需求成为在便携式、低功耗设备上实现深度学习算法及其应用的主要障碍。文中基于树莓派平台,借助 Intel 的视频处理器(VPU)低功耗加速模块,设计并实现了基于残差特征提取模块 CNN 模型的实时人脸检测系统。结果表明,相较于单纯使用树莓派 CPU 进行计算,文中方法在视频流中检测人脸和人脸关键点提取的实验中分别实现了 18.62 倍和 17.46 倍的加速,在便携式设备中实现快速、实时、在线的人脸检测和特征点提取成为现实,同时为使用便携式、低功耗设备运行深度学习算法提供了一种确实可行的方案。

关键词:嵌入式系统;深度学习;VPU;人脸检测;人脸关键点
中图分类号:TP391.4 **文献标志码**:A **文章编号**:1000-582X(2021)07-115-14

Design and implementation of embedded real-time face detection system based on VPU acceleration

YAN Jia[1a], *ZHANG Yuelin*[1b], *DENG Changjian*[2]
(1a. College of Artificial Intelligence; 1b. WESTA College, Southwest University, Chongqing 400715, P. R. China; 2. School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P. R. China)

Abstract: The high design cost and huge computing resource demand of intelligent devices have become the main obstacles to the implementation and application of deep learning algorithm in portable and low-power devices. In order to solve these problems, in this paper, based on the raspberry PI platform and with the help of Intel video processing unit(VPU) low-power acceleration module, a real-time face detection system based on CNN model with residual feature extraction module was designed and implemented. The experimental results show that compared with using central processing unit(CPU) of raspberry PI alone, the proposed method achieved 18.62 times and 17.46 times acceleration respectively in the experiments of face detection and face alignment detection in video stream. It realized the fast, real-time and online face detection and face alignment extraction in portable devices. Meanwhile, it also provided a feasible scheme for the operation of deep learning algorithm in portable and low power devices.

Keywords: embedded system; deep learning; VPU; face detection; face alignment

收稿日期: 2020-11-04
基金项目:国家重点研发计划资助项目(2018YFB1306603);国家自然科学基金资助项目(61672436)。
Supported by National Key Research and Development Program of China(2018YFB1306603) and National Science Foundation of China(61672436).
作者简介:闫嘉(1983—),男,副教授,硕士生导师,主要从事机器学习和模式识别研究,(E-mail)yanjia119@163.com。

近年来，计算机视觉不断蓬勃发展，卷积神经网络(CNN)[1-2]作为深度学习算法中重要组成部分，在图像分类[3]，物体检测[4-5]以及图像重建[6-7]中已经大显身手，例如，活体检测、自动驾驶、无人超市等[8-10]。目前，计算机视觉的应用，例如，人脸检测和人脸关键点提取，主要运行于具有强大计算能力的图形处理器(GPU)[11]的云端服务器，即先将图片上传到服务器，运算之后再传回本次图片检测的结果，但这种方法需要大量的计算资源和高昂的设计费用；由于网络带宽的限制，上传图像和下载响应结果都有很高的时延，在实现本地视频流中实时检测人脸和关键点提取就变得尤为困难。此外，上传到云端的数据存在个人隐私被泄漏的安全风险；或当使用场景在地下停车库、穿山隧道等信号盲区时，云端计算的方式的就显得力不从心。

在实时目标检测方面，Ren 等[12]提出 Faster R-CNN，第一次提出使用“锚框”选择候选框，并结合边框回归、特征图共享的思想，采用非极大值抑制(NMS)、感兴趣区域(RoI)池化的方式识别物体，在 GPU 上实现了实时的物体检测；此后，Redmon 等[13-15]提出了 yolo 网络，其检测速度相较于 Faster R-CNN 成倍增长。作者在此后又提出了改进后的 yolo v2 和 yolo v3 网络，提高了准确度，减少了检测用时。在实际应用中，Faster R-CNN 虽然性能比较稳定，但是其非端到端的“两步”网络结构需要消耗大量的计算资源，增加了检测耗时；而 yolo 网络虽然更快，但在一些比较严苛复杂的环境中，并不能达到产品需求。为了能够结合这 2 种网络的优点，扬长避短，Wei 等[16]提出了 SSD 网络，在不同尺度的特征图上，同时实现物体识别检测和物体边界框回归。其端到端和多特征图的网络结构，使其检测精度媲美 Faster R-CNN，检测速度几乎和 yolo 相当。即使是在图像比较分辨率小的情况下，SSD 都能够达到更高的检测精度。随着卷积层的增多，计算的时间也会更长，不利于便携式设备的实时应用。此后，谷歌大脑(Google Brain)团队提出了 efficient net 的网络模型，并以此提出了一种多维度混合的模型放缩方法，同时兼顾模型的速度与精度[17]，然而这种方法需要针对特定的使用场景进行针对性的参数调整，极大延长了实际应用的开发周期。

在卷积神经网络实现人脸关键点提取方向上，Sun 等[18]提出了基于级联卷积神经网络的人脸特征点定位方法，使用多个级联的 CNN 网络，根据“由粗到精”的思想逐步实现左右眼、鼻尖和左右嘴角的精确定位。Face＋＋研究者以这篇文章为基础，采用四级级联网络结构，结合“由粗到精”的思想，实现包含眉毛、眼睛、鼻子、嘴巴共计 51 个内部关键点，人脸轮廓共计 17 个外部关键点，总计 68 个关键点的精确检测[19]。此外，Zhang 等[20]提出一种多任务级联卷积神经网络(MTCNN)，同时解决人脸检测和 5 个人脸关键点定位问题。而使用神经网络实现人脸特征点提取的科研者，大多使用“由粗到精”的思想，采用的多级网络结果也大大增加了计算时间的训练成本。Feng 等[21]则在 2017 年提出了一种人脸特征点提取专用的 loss 方程：Wing loss，这是第一篇在人脸关键点检测任务上对损失函数进行分析的文章，使其对小误差更加“友好”，实验结果也表明，相比于传统的 L1，L2 以及 smooth L1 方程，Wing loss 误差更小。

在嵌入式硬件平台上，树莓派作为一种基于 ARM 架构的开发板，能够支持运行 Linux 操作系统，支持 1 080 P高清视频解码，CPU 可以稳定工作在 1 GHz，已经广泛用于教育[22]、智能遥控设备[23]、物联网开发以及智能家居[24-25]等领域。1 GHz 的 CPU 运算频率并不能满足深度学习应用的计算资源需求，需要借助专用的视觉加速模块，如现场可编程门阵列(FPGA)或通用加速模块。

在近年来 FPGA 实现加速的研究中，为使 FPGA 针对特定的网络进行优化设计[26]，达到高速运算的需求，刘谦让等[27]在 2018 年利用 CNN 卷积神经网络中存在着较多稀疏特征的特性，充分挖掘 CNN 卷积运算稀疏性的特点，结合 FPGA 并行矩阵乘法器的实现方法，能够实现相比于传统 CNN 加速器 19％的提速。而张榜等[28]再次提出使用数据量化的方式将浮点数转为定点，降低运算开销，并提出从 FPGA 发起数据访问的架构，避免 FPGA 性能下降的问题，在实际实验中，性能功耗比达到了 6.81 GOPS/W。使用 FPGA 进行定制开发，需要了解整个算法的流程以及数据的输入输出形式，结合 FPGA 的特点设计相应的加速结构[29]；由于高端 FPGA 价格昂贵，研发需要大量的人力物力进行支撑，因而在成本预算和研发时间上，并不适合快速的开发和应用场景。为能实现使用神经网络加速模块的快速开发和使用，Intel 公司自行研发了 MYRIAD 加速芯片，芯片的核心部件是向量计算单元，其小于 1 块硬币的尺寸使其可以容纳在 1 个类似于 U 盘的加速棒中，并兼容 Window 以及 Linux 操作系统。作为一种通用图形处理加速模块，最新研发的 MYRIAD X 图

形处理芯片能够提供大约 1 TFlops 的计算能力，搭建该芯片的视觉计算棒 NCS2 平均功耗仅约为 1 W，同时，结合 Intel 发布的 OpenVINO 视觉处理包，开发者能够快速构建神经网络加速器，推动嵌入式人工智能产品的开发和应用。

基于上述分析基础，设计采用以树莓派 4B 为系统平台，结合 Intel 的视觉神经棒，采用基于残差模块搭建的卷积神经网络模型，在树莓派上实现实时检测人脸位置以及提取 83 个人脸关键点，并额外训练了另外 3 个网络模型作为对比。首次提出了以树莓派 4B 为平台使用视觉神经棒对基于残差模块的卷积神经网络进行加速的方法，并对其运行效果与另外 3 个网络模型进行了对比分析。基于低功耗的加速模块和嵌入式的开发平台，可以为当前在嵌入式设备中应用在线深度学习提供一种可行的方案，也可应用于无人机的自动避障导航设计、无人车的行为规划、关键场所的人脸识别等，具有较大的发展空间和商业价值。

1　人脸检测和关键点提取网络

为验证设计方案的可行性，总共设计了 2 个卷积神经网络：一个为人脸检测网络，用于实时检测人脸的位置和大小；另一个为人脸关键点提取网络，在实时检测出人脸位置后，提取出该人脸的人脸关键点，并计算出这些关键点在原图中的位置。

1.1　人脸关键点网络

鉴于 SSD 既结合了 Faster R-CNN 锚框的思想，即先验框[30]，也结合了 yolo 采用“端到端”的网络结构。文中对该网络加以改进，提出基于残差模块提取特征的 SSD。该网络使用残差模块进行特征提取，利用多个不同尺度的特征图产生多种不同的特征映射，并对应每一个特征映射到原图的位置，使用一个卷积过滤器来评估人脸的边界框和人脸的得分。

文中所设计的 SSD 网络不使用全连接层，而是使用由多个卷积层堆叠而成的特征提取骨架产生不同尺度的特征图，而后直接使用过滤器来过滤特征图产生的候选框。由于使用的 VPU 所搭载的图形处理芯片针对卷积运算进行了优化，所以部署以卷积层为主要结构的网络至 VPU 可更好地体现 VPU 对图像处理的加速效果。因在网络设计时，卷积层过多可能出现梯度弥散的问题，文中引入了残差模块[29]，其主要思想是在网络中增加了“直连通道”，相比于以往的网络大多是对输入做非线性变换得到输出，残差模块则通过直接将输入信息直连到输出，保护信息的完整性，整个网络只需要学习输入、输出相差的那一部分，进而简化学习目标和难度，如图 1(a)所示。根据实际需求，设计采用了 2 种残差模块，如图 1(b)所示。其中，conv1，2S 表示卷积核大小为 1x1，步长为 2，卷积类型为 SAME 的二维卷积。

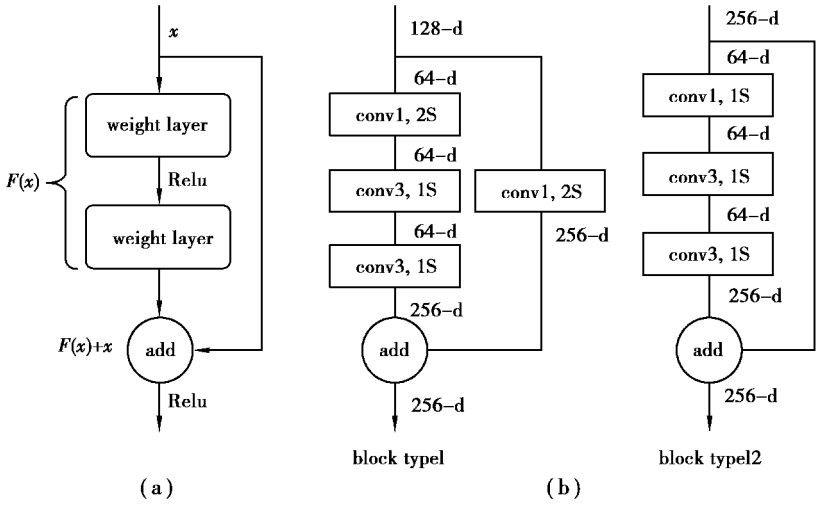

图 1　残差模块

Fig. 1　Residual blocks

残差模块的设计原则为：

1)当残差模块输出特征相比于输入特征的大小减半时(即在残差支路和同等映射支路同时存在一个卷积步长为2或者存在一个2×2的池化层)，输出特征的通道数相比于输入特征数量翻倍。在这种情况下，同等映射的方式为：使用卷积核大小为1×1，卷积步长为2的卷积核对输入特征进行卷积运算，使同等映射的输出特征的通道数量为输入特征的2倍，如block type1所示；

2)对于残差模块输出特征和输入特征的大小相同的层(即模块中所有的卷积步长为1)，输出特征相比于输入特征具有相同通道数，如block type2所示。

在此基础上，根据SSD网络的实际需求，使用残差模块搭建的特征提取单元，如图2所示。每一个特征提取单元输出一个不同尺度的特征图，用于预测人脸的位置和每个人脸预测框的得分。

根据上述设计的残差模块以及特征提取单元，设定人脸检测网络的图片大小为384×512像素，设计基于SSD网络框架的人脸检测网络，如图3所示。残差模块的数量越多，得到的特征图能够反映原始图片中更加深入的细节，例如，物体的边、角、形状等信息[30]。但考虑到嵌入式系统的实时性和可用性，只需要经过5个block type 1和4个block type 2残差模块得到初始的feature map即可；再次经过2个特征提取单元feature extractor提取更加深层次的特征，最后将有3个不同尺度的特征图用于人脸位置和得分的预测。

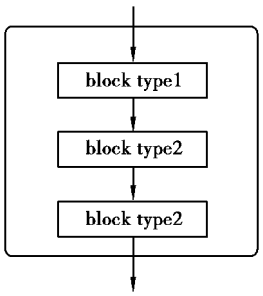

图2 特征提取单元

Fig. 2 Feature extractor

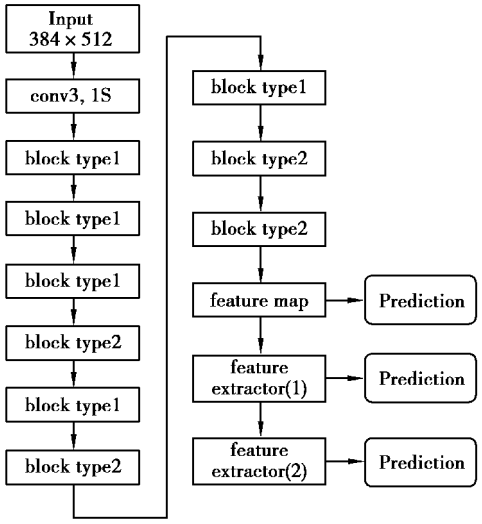

图3 人脸检测网络结构

Fig. 3 Face detection network structure

1.2 关键点提取网络

使用SSD结构搭建人脸检测网络后，可以继续对检测到的人脸进行关键点提取，检测关键点分别包括脸部、眼睛、眉毛、嘴唇、鼻子轮廓共计83个关键点。人脸关键点提取网络在卷积特征图后连接的是全连接层，即内积层，并非在卷积特征图后连接的卷积层，即卷积过滤器。因为卷积层输出特征的一个点对应的感受野是原图上的一个区域，而全连接层每一个输出节点所对应的感受野则是整个图像，能够更好地学习到人脸关键点的特征和该人脸关键点输出位置之间的联系。而人脸关键点提取网络的搭建依旧采用节1.1中所述的残差模块，其网络结构示意图，如图4所示，其中fc表示全连接层(full connection layers)，设定该网络的输入图片的大小为128×128像素。

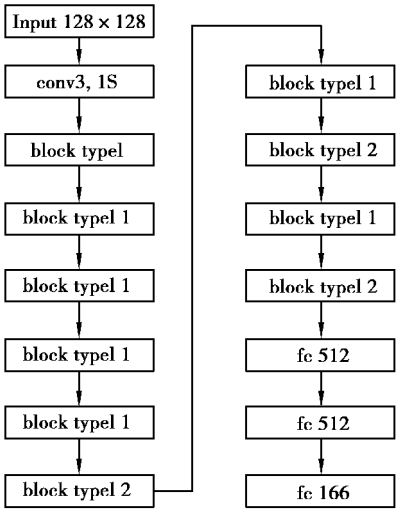

图4 人脸关键点提取网络结构

Fig. 4 Network structure of facial key points extraction

2　系统设计

为了能够合理调用软件和硬件资源，充分发挥视觉神经棒的功能，文中基于 python 设计了一套整体系统用于实现实时人脸检测和人脸关键点提取。根据人脸实时检测的功能需求分析，进行人脸检测的每一帧可以从 3 个渠道获取：摄像头、本地视频和本地图片。而进行推理运算的设备可以分为 2 类：CPU 和视频处理器(VPU)；此外，人脸关键点提取需要在一张图片数据中首先找到人脸的位置，因此，人脸检测是人脸关键点提取的必要条件，系统界面如图 5 所示。

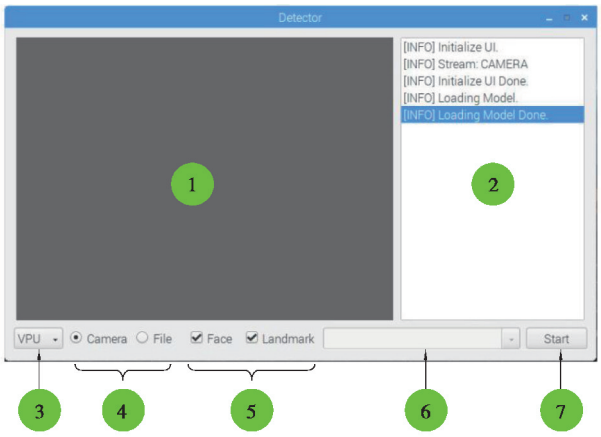

图 5　设计系统界面

Fig. 5　System interface of the design

图 5 中的标注分别为：①图像显示区域。实时显示每一帧的图像以及检测结果。②状态显示区域。显示当前程序运行状态，如初始化 UI、加载模型、切换设备等。③设备选择。选择推理模型运行的设备，即选择基于 CPU 或者 VPU 运行。④选择图像帧来源。每一帧图像获取的来源，支持使用摄像头或者本地媒体文件。⑤选择人脸检测和人脸关键点提取。在勾选后执行检测任务。⑥选择本地媒体文件。当使用本地媒体文件作为图像数据来源时，用于选择待检测的媒体文件。⑦检测开始或检测停止。切换“开始检测”和“停止检测”的状态。

设计的人脸检测软件系统的整体工作流程，如图 6 所示。为 3 个主要步骤：第一，初始化 UI 界面和各个控件的功能；第二，将训练好的人脸检测网络和人脸关键点提取网络分别加载到 CPU 运行环境和 VPU 运行环境；第三，打开摄像头后获取视频流，读取每一帧图像，实时检测人脸的位置并提取相应的人脸关键点，并显示检测结果。

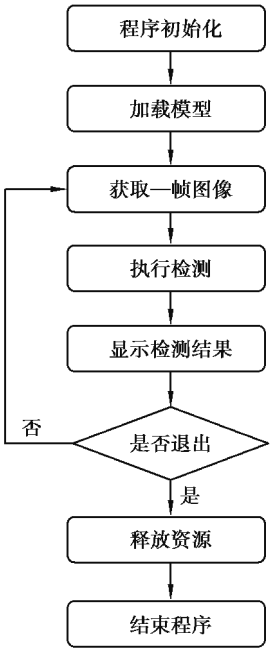

图 6　软件系统整体流程

Fig. 6　The overall process of software system

3　训练和实际测试

系统涉及的卷积网络模型由 tensorflow(python)深度学习框架进行搭建，使用 GPU(GTX 960M)进行并行加速训练，得到的网络模型使用 OpenVINO 工具包转换为视觉神经棒可加载的推理模型，并将整个系统部署到树莓派 linux 系统运行。为了能够体现出残差模块的引入对文中神经网络 SSD 的改进效果，使用相同数据集训练了未经改动的使用 VGG 网络作为骨架的原版 SSD300 网络，通过对比 SSD300 与文中 SSD 网络的运行效果即可得出残差模块的作用。此外，为了使实验现象更具说服力和代表性，还使用相同数据集额外训练了 yolo v4 和 yolo v4-tiny 2 个较为新颖的网络模型与文中 SSD 网络进行对比实验。额外选取的 2 个网络输入图大小为 416×416，与 SSD 网络的输入图(512×

384)大小相近,均不使用全连接层,且均采用多尺度预测的结构,与SSD网络具有一定的相似性。

3.1 数据集

训练样本数据来自2 000人的175 700张包含了亚洲人脸的样本图片,而验证样本数据来自于其他200人的18 025张相同大小的人脸图片。每一张图片大小为400×400像素,每个人有从50张到100张不等的不同场景,不同姿态的图片样本,人脸位置和关键点由Face++标注。每一张人脸图片有且仅有一个人脸数据文件,其中包含了所有的人脸框和其对应的83点人脸关键点的数据。

此外,假如训练的样本过于单一,例如,在样本中人脸的位置始终处于图像的中间部分,这样的样本训练出来的模型,也只能在图像的中间部分能够比较准确地检测人脸,这种网络便失去了泛性。本次模型的训练过程中,使用数据增强的方式,一来增加训练的数据量,从而提高网络的泛化能力;二来也可以增加噪声数据,提高模型的鲁棒性。数据增强的方式采用了:随机水平翻转;随机顺时针-20°~20°;随机等比缩放0.5~1.2倍;在保证人脸不超出图像范围的情况下随机平移,以及随机图像背景,如图7所示。

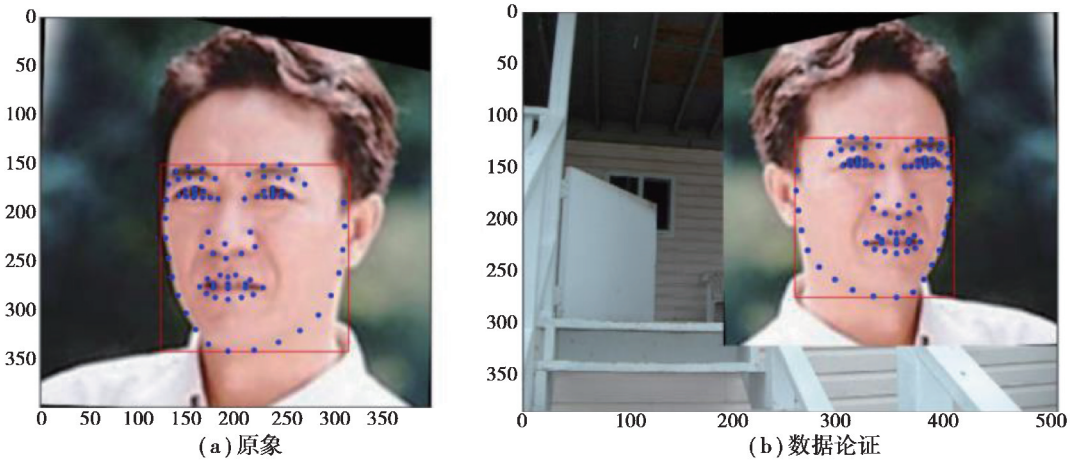

图7 随机数据增强示例

Fig. 7 Example of random data enhancement

3.2 训练网络模型

在网络模型的训练过程中采用了指数衰减学习率方法,将SSD网络及关键点提取网络的初始学习率设置为0.000 1,衰减因子设置为0.95,每批次输入样本数量分别设置为1和20,使用GPU分别训练60 000次和50 000次。作为对比组的其他3个网络,将yolo v4和yolo v4-tiny的初始学习率分别设置为0.001和0.002 6,在每批次输入32个样本的条件下均训练5 000次;将SSD300的初始学习率设置为0.000 01,每批次输入16个样本,进行10 000次训练。

为了客观表示并评价设计的有效性和实用性,首先建立了模型需要的人脸数据集,以查准率和召回率作为人脸检测模型的评价指标,见式(1)和式(2);并以平均像素误差作为人脸关键点提取的评价指标,其中,precision和recall分别为查准率和查全率,TF和NF分别表示检测出人脸中,检测正确和检测错误的数量,而GT表示真实的人脸数量。

$$\text{precision}=\frac{\text{TF}}{\text{TF}+\text{NF}},\tag{1}$$

$$\text{recall}=\frac{\text{TF}}{\text{GT}}。\tag{2}$$

在人脸检测网络中,设定检测到人脸框相比于真实框的最大交并比大于0.8时,认定该预测框检测到了正确的人脸结果,否则检测到的是错误的结果或者预测人脸框的偏移量较大,人脸检测网络及对比实验网络在测试集中的表现结果如表1所示,其中,SSD网络的查准率为90.41%,查全率为89.0%;SSD300的查准率为73.42%,查全率为72.52%;yolo v4和yolo v4-tiny的查准率分别为94.93%和84.03%,查全率分别为93.90%和81.96%,人脸检测网络表现良好。此外,对比文中SSD网络与原版SSD300网络的测试效果可发

现，残差模块的引入极大地改进了性能，查准率和查全率均提高了近 17 个百分点。

表 1　人脸检测网络测试效果

Table 1　Test results of face detection network

model	GT	TF	NF	precision	recall
SSD	18 025	16 043	1 701	0.904 1	0.890 0
SSD300	18 025	13 071	4 730	0.734 2	0.725 2
yolo v4	18 025	16 925	904	0.949 3	0.939 0
yolo v4-tiny	18 025	14 773	2 808	0.840 3	0.819 6

为分析人脸关键点网络的性能，使用包含 200 人，18 025 张图的验证集上，对每一张图片随机顺时针旋转－20°～20°，再次进行 20 000 次关键点的提取测试，统计了误差大小和误差场次之间的关系。如图 8 所示，$N(\text{error}_i)$表示在像素误差为 error 条件下的图片数量，其条件概率为 $p(N\mid\text{error}_i)$，mean value 表示验证集上所有图片的像素误差的加权均值为

$$\text{mean}=\sum_i \text{error}_i\times p(N_i\mid\text{error}_i)。\tag{3}$$

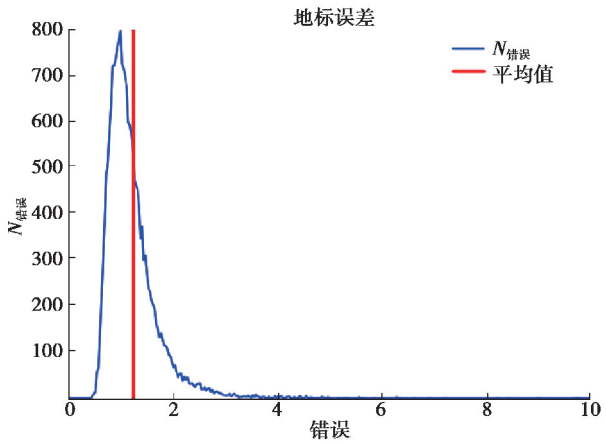

图 8　人脸关键点提取网络测试效果

Fig. 8　Test results of facial key points extraction network

3.3　CPU 和 VPU 性能对比

设计中，树莓派 CPU 的运算精度为 32 位浮点数，而视觉神经棒 VPU 的运算精度为 16 位浮点数，为对比运算精度对设计的影响，给出了树莓派 CPU 和视觉神经棒 VPU 的 SSD 网络人脸检测和人脸特征点提取的效果对比，如图 9 所示，其中的(a)和(b)、(c)和(d)分别为使用树莓派 CPU、视觉神经棒 VPU 的推理预测结果。图 10 和图 11 分别为 SSD300 网络、yolo v4 和 yolo v4-tiny 网络的人脸检测和人脸特征点提取效果。其中，图 10 的(a)和(b)、(c)和(d)分别为 SSD300 网络使用树莓派 CPU、视觉神经棒 VPU 的推理预测结果，图 11 的(a)和(b)、(c)和(d)分别为 yolo v4-tiny 网络使用树莓派 CPU 和视觉神经棒 VPU 的推理预测结果。由于 yolo v4 网络结构复杂且规模较大，使用树莓派 CPU 运行会因内存不足而失败，因此 yolo v4 只使用 VPU 进行人脸检测和人脸特征点提取，如图 11(e)和(f)所示。在人脸位置检测和人脸关键点提取的实际测试中，使用 CPU 和 VPU 的预测结果受到计算精度的影响很小，预测的位置基本相同。

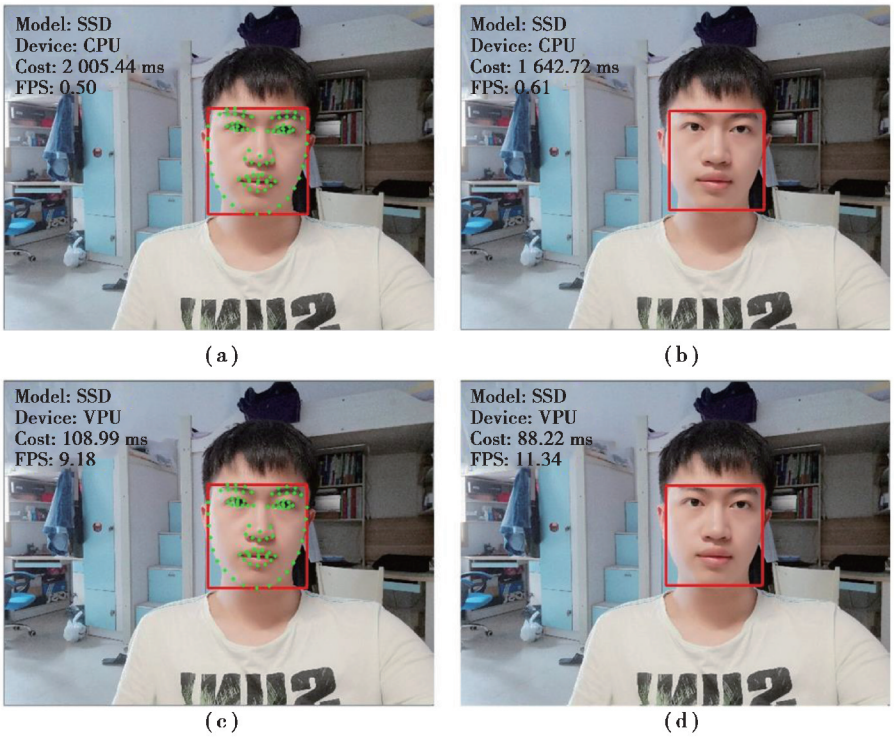

图 9 CPU 和 VPU 对比(SSD)

Fig. 9 Comparison of CPU and VPU(SSD)

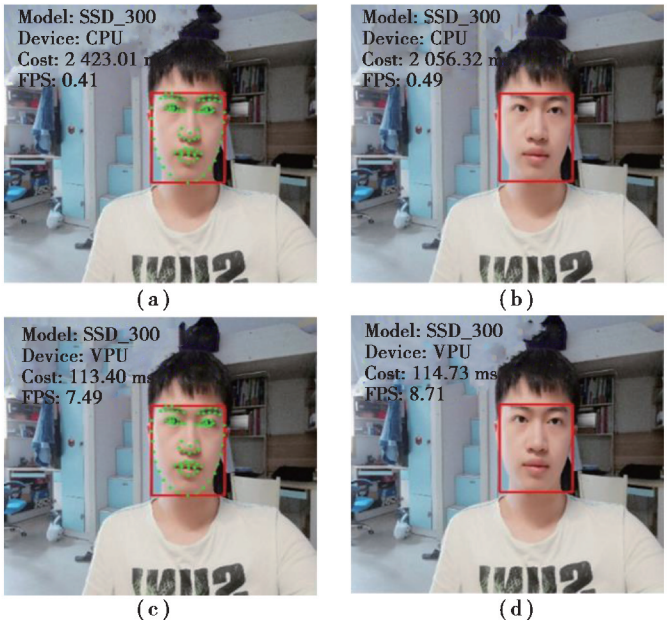

图 10 CPU 和 VPU 对比(SSD300)

Fig. 10 Comparison of CPU and VPU(SSD300)

在此基础上,为体现使用视觉神经棒 VPU 进行加速运算对实时检测的重要性,在效果对比的基础上也增加了运算性能的对比。根据图 9～图 11 的表现结果,总结为表 2 所示,其中 T1、T2 分别表示同时执行人脸检测、人脸关键点提取耗时和仅执行人脸检测耗时。从表 2 可以明显看出,在便携式低功耗终端设备的实际计算应用中,不管是同时进行人脸检测和人脸关键点提取,还是单独进行人脸检测或人脸关键点提取,VPU 相比于 CPU 来说运行不同人脸检测网络模型的速度都取得较大的提高。

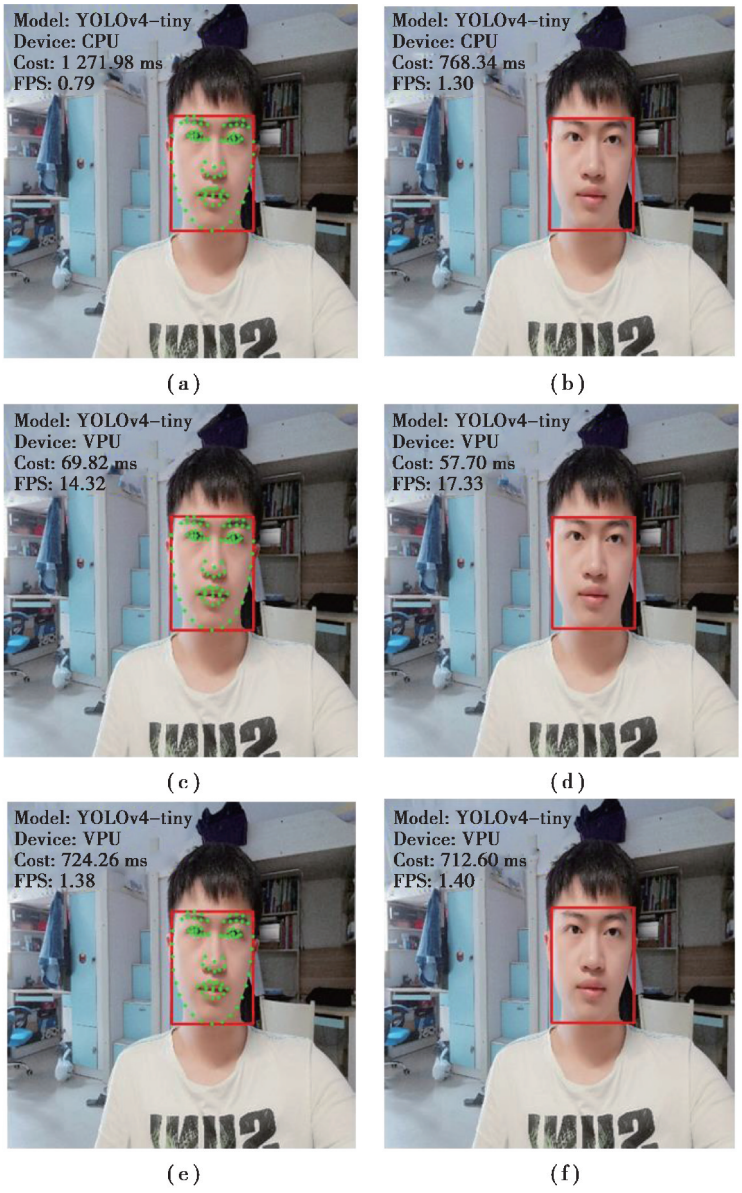

图 11　CPU 和 VPU 对比(yolo v4 & yolo v4-tiny)

Fig. 11　Comparison of CPU and VPU(yolo v4 & yolo v4-tiny)

具体来说，使用相同像素和大小的图片，同时进行人脸检测和人脸关键点提取，VPU 相比于 CPU 的加速比最高可达 18.40 倍；在单独检测人脸位置任务时，VPU 相比于 CPU 加速比最高可达 18.62 倍；除去人脸检测消耗的推理时间，单独执行 83 个人脸关键点提取时，对应加速比最高可达 41.55 倍。值得注意的是，使用 VPU 能够成功运行 CPU 无法运行的 yolo v4 网络，充分说明了 VPU 不仅能够加速网络运行，还能使资源有限的嵌入式设备得以成功部署较复杂的网络。通过分析不难发现，使用 VPU 进行加速，仅能使 yolo v4 实现较低帧数，这说明现有方法还存在网络选择不当的缺点，在下一步的实验中应选择更轻量的网络。

表2 CPU和VPU性能对比

Table 2 Performance comparison of CPU and VPU

model	精度	CPU/1.2GHz	VPU/NCS2	加速比
		32bit Float	16bit Float	—
	功耗	3.37 W	1 W	—
SSD	T_1	2 005.44 ms	108.99 ms	18.40
	T_2	1 642.72 ms	88.22 ms	18.62
	T_1-T_2	362.72 ms	20.77 ms	17.46
SSD300	T_1	2 423.61 ms	133.46	18.16
	T_2	2 056.35 ms	114.75 ms	17.92
	T_1-T_2	367.26 ms	18.71 ms	19.63
yolo v4	T_1	—	724.26 ms	—
	T_2	—	712.60 ms	—
	T_1-T_2	—	11.66 ms	—
yolo v4-tiny	T_1	1 271.98 ms	69.82 ms	18.28
	T_2	768.34 ms	57.70 ms	13.32
	T_1-T_2	503.64 ms	12.12 ms	41.55

对表2中的结果进行横向对比可以发现,网络使用VPU运行消耗的时间与卷积层数量呈正相关关系。对比SSD网络与yolo v4-tiny网络,前者卷积层数量为后者的1.83倍,运行耗时(T_2)为后者的1.53倍。对比SSD网络与yolo v4网络,后者卷积层数量为前者的2.58倍,运行耗时(T_2)则为前者的8.08倍。由此可知,虽然VPU可加速神经网络的运行,但当网络规模达到一定水平时,运行耗时便会成倍增加,所以在VPU的实际使用中仍需控制模型体积。对比SSD与SSD300的表现,SSD300运行速度慢于SSD,可能与原版SSD300网络同时使用了6个不同尺度的特征图进行预测有关。进一步观察对比yolo v4-tiny和SSD网络使用VPU运行的加速比(T2)可知,加速比大致随网络层数增加而提高。由于yolo v4网络的层数比SSD网络的层数要多,虽然其加速比无法直接通过测量得出,但可大致猜测yolo v4使用VPU运行能够获得比SSD网络更高的加速比,即VPU对神经网络的加速效果随网络规模的增加而提升。

此外,使用树莓派CPU进行推理运算时,树莓派需要满载运行,运行功耗约为3.37 W,而使用VPU的运算功耗为1 W,仅为树莓派满载功耗的29.7%。通过对比可知,视觉神经棒VPU在视觉加速处理上有着较大的优势,而且功耗更低,速度更快,更适合在便携式低功耗终端设备的计算应用。

综合表1和表2可知,设计的基于残差模块的SSD网络无论是在耗时还是在准确度上相比原版SSD300网络都有了较大的提升,证明了残差模块的引入极大地改进了SSD300网络的性能。比较SSD与yolo v4-tiny可以发现,SSD以较复杂的网络结构实现了更高的精度,并且只带来了耗时的小幅度增加。此外,对比SSD与yolo v4的表现可以发现,虽然SSD在准确度上稍差于yolo v4,但其运行速度却显著快于后者,可见SSD在速度和精度上达到了更好的平衡,能以更短的耗时实现较好的效果。充分说明了文中设计方法的优越性以及创新性,在综合考虑运行速度和准确度的情况下能达到优于现有网络的效果。

3.4 实际场景测试

根据3.3节的结论,因嵌入式CPU受到计算资源的限制,使用视觉神经棒进行加速运算是一种可行的方案。在上述结论的基础上,进一步给出了在实际场景下,使用视觉神经棒进行加速运算的表现效果,图12为视频流中提取每一帧并使用SSD网络进行检测的结果。图12(a)所展示的是同时检测2个人脸并分别提取人脸关键点,耗时109.75 ms,检测耗时并没有因为人脸的增多而增加过大,且人脸预测框的大小、位置和关键点的位置都很符合实际;图12(b)展示了当有遮挡物遮挡了人脸后,该网络模型仍然能够检测出人脸的位置,但是提取的部分关键点因为遮挡的原因和实际关键点有部分误差;图12(c)中,当人处于侧脸时,依旧能够检测人脸位置,并较为准确地提取人脸关键点;图12(d)中则展示了人在张嘴,人脸拉长的情况下,依旧能够准确判断人

脸位置，较为准确地表现出人脸关键点的位置，并没有因为一些复杂表情而受到影响。图 13 为使用 SSD300 网络进行实时检测的结果，使用与图 12 相同的姿势，图 13 的(a)、(b)、(c)、(d)分别为多脸、遮挡、侧脸和张嘴的检测结果，可见其运行效果良好。图 14 为使用 yolo v4 和 yolo v4-tiny 进行视频流检测对比实验的结果，使用了与图 12 相同的姿势进行检测，图 14(a)和图 14(e)为分别使用 2 个网络同时检测 2 个人脸并分别提取人脸关键点，图 14(b)和图 14(f)为 2 个网络在人脸被遮挡时的表现情况，图 14(c)和图 14(g)展示了 2 个网络存在一定误差的情况下分别成功检测出侧脸，图 14(d)和图 14(h)则展示了当人在张嘴时的检测情况，这 2 个网络在不同的帧率下同样实现了较好的效果，充分证明了 VPU 加速功能的普适性。

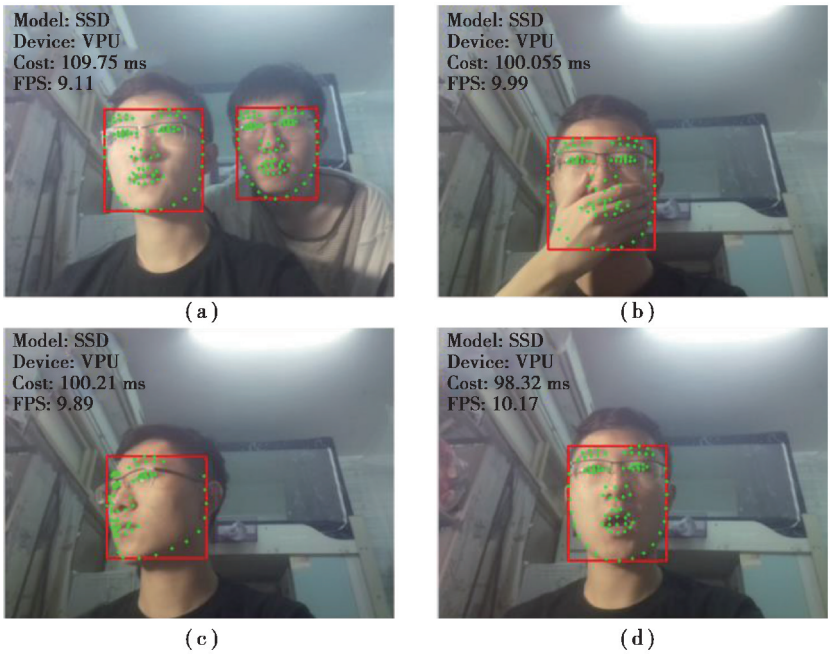

图 12　实际场景测试(SSD)

Fig. 12　Actual scenario test(SSD)

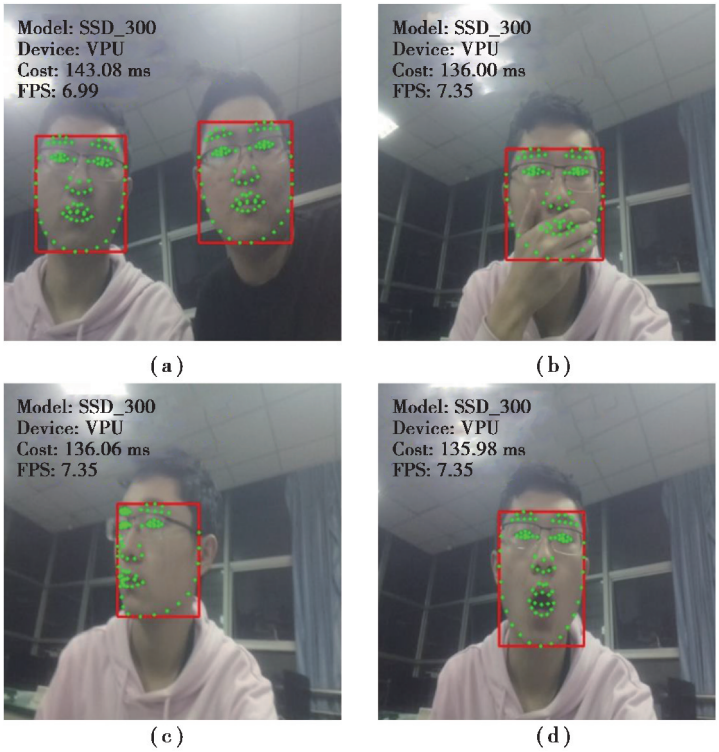

图 13　实际场景测试(SSD300)

Fig. 13　Actual scenario test(SSD300)

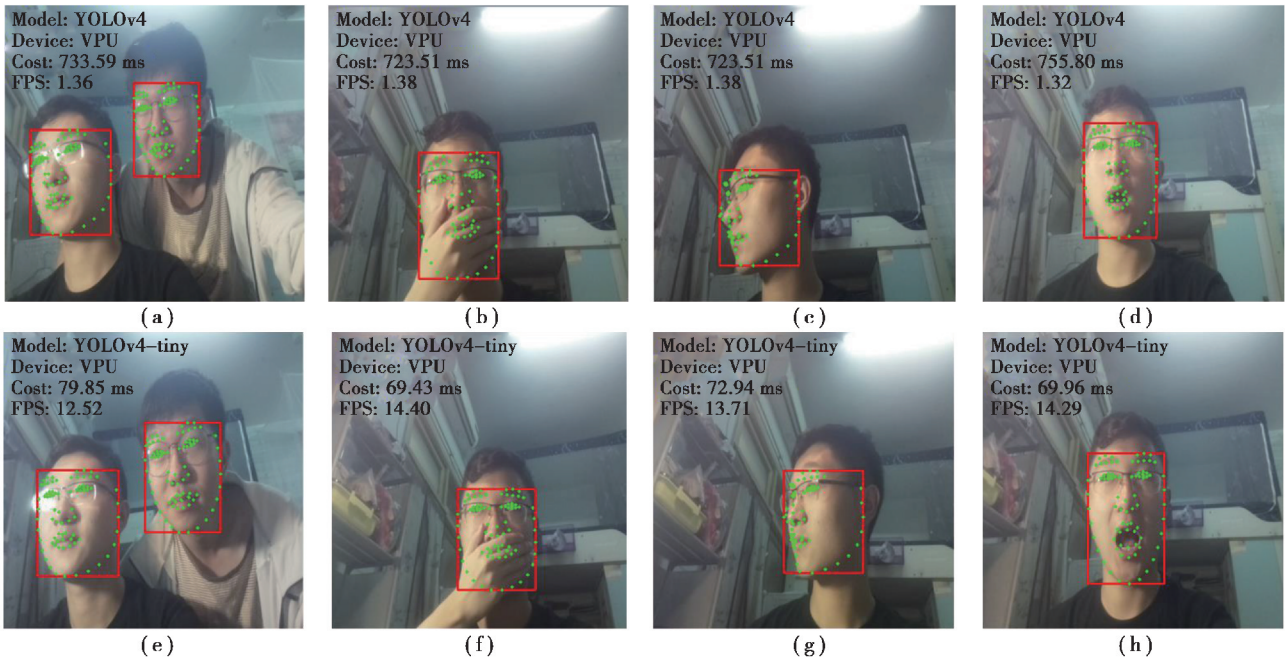

图 14 实际场景测试(yolo v4 & yolo v4-tiny)

Fig. 14 Actual scenario test(yolo v4 & yolo v4-tiny)

4 结 论

使用视觉神经棒 VPU 加速计算模块,对于嵌入式系统中部署计算机视觉应用有着重要的作用和价值。文中设计并训练了 2 种网络:人脸检测网络和人脸关键点提取网络,额外选取了其他 3 种网络作为对比。经过优化训练部署到树莓派后,使用视觉神经棒 VPU,在同时执行人脸检测和关键点提取、单独执行人脸检测和单独执行关键点提取任务时,相比于单独使用嵌入式 CPU 进行运算,加速比分别可达 18.40,18.62,17.46 倍,说明了设计方案的可行性和有效性,在此基础上,可以进一步在嵌入式系统上实现情绪分析、情绪识别、人物核验等功能,具有广阔的应用前景。此外,树莓派系统使用的是内存卡,其读写速度在一定程度上限制了树莓派和视觉神经棒的数据传输速率。下一步的研究重点包括:1)进一步实现基于嵌入式系统的在线情绪识别、情绪分析等功能;2)使用一种内存读写速度更快的嵌入式平台,以充分发挥视觉神经棒的性能;3)优化视频流中对每一帧图像的人脸和人脸关键点的预测和跟踪,利用当前帧的结果预测下一帧,加快检测速度和效率。

参考文献:

[1] Le Cun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[2] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.

[3] Yan Z C, Zhang H, Piramuthu R, et al. HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition[C]// 2015 IEEE International Conference on Computer Vision(ICCV). December 7-13, 2015, Santiago, Chile. IEEE, 2015: 2740-2748.

[4] Gidaris S, Komodakis N. Object detection via a multi-region and semantic segmentation-aware CNN model[C]// 2015 IEEE International Conference on Computer Vision(ICCV). December 7-13, 2015, Santiago, Chile. IEEE, 2015: 1134-1142.

[5] Bappy J H, Roy-Chowdhury A K. CNN based region proposals for efficient object detection[C]// 2016 IEEE International

Conference on Image Processing(ICIP). September 25-28, 2016, Phoenix, AZ, USA. IEEE, 2016: 3658-3662.

[6] Lin Y F, Li J B, Wang H J. DCNN-GAN: reconstructing realistic image from fMRI[C]// 2019 16th International Conference on Machine Vision Applications(MVA). May 27-31, 2019. Tokyo, Japan. IEEE, 2019.

[7] 张圣祥，郑力新，朱建清，等. 采用深度学习的快速超分辨率图像重建方法[J]. 华侨大学学报(自然科学版)，2019，40(2)：245-250.
Zhang S X, Zheng L X, Zhu J Q, et al. Fast super-resolution image reconstruction method using deep learning[J]. Journal of Huaqiao University(Natural Science), 2019, 40(2): 245-250.(in Chinese)

[8] 许晓. 基于深度学习的活体人脸检测算法研究[D]. 北京：北京工业大学，2016.
Xu X. Research on deep learning based face lineness detection algorithm[D]. Beijing: Beijing University of Technology, 2016.(in Chinese)

[9] 王科俊，赵彦东，邢向磊. 深度学习在无人驾驶汽车领域应用的研究进展[J]. 智能系统学报，2018，13(1)：55-69.
Wang K J, Zhao Y D, Xing X L. Deep learning in driverless vehicles[J]. CAAI Transactions on Intelligent Systems, 2018, 13(1): 55-69.(in Chinese)

[10] 朱柳依. 结合模板匹配与单样本深度学习的货架商品定位与识别技术研究[D]. 杭州：浙江大学，2018.
Zhu L Y. Research on grocery product detection and recognition technology by template matching and one-shot deep learning[D]. Hangzhou: Zhejiang University, 2018.(in Chinese)

[11] 张印，董兰芳，王建富. 基于 GPU 的人脸检测和特征点定位研究[J]. 电子技术，2014，43(9)：38-42,37.
Zhang Y, Dong L F, Wang J F. Face detection and feature localization based on GPU[J]. Electronic Technology, 2014, 43(9): 38-42,37.(in Chinese)

[12] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[13] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). June 27-30, 2016, Las Vegas, NV, USA. IEEE, 2016: 779-788.

[14] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). July 21-26, 2017, Honolulu, HI, USA. IEEE, 2017: 6517-6525.

[15] Redmon J, Farhadi A. Yolov3: An incremental improvement [EB/OL]. arXive-prints, (2018-04-08)[2020-03-10]. https://arxiv.org/abs/ 1804.02767.

[16] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector[M]// Computer Vision - ECCV 2016. Cham: Springer International Publishing, 2016: 21-37.

[17] Tan M, Le Q V. EfficientNet: rethinking model scaling for convolutional neural networks[J]. International Conference on Machine Learning, 2019.

[18] Sun Y, Wang X G, Tang X O. Deep convolutional network cascade for facial point detection[C]// 2013 IEEE Conference on Computer Vision and Pattern Recognition. June 23-28, 2013, Portland, OR, USA. IEEE, 2013: 3476-3483.

[19] Zhou E J, Fan H Q, Cao Z M, et al. Extensive facial landmark localization with coarse-to-fine convolutional network cascade[C]// 2013 IEEE International Conference on Computer Vision Workshops. December 2-8, 2013, Sydney, NSW, Australia. IEEE, 2013: 386-391.

[20] Zhang K P, Zhang Z P, Li Z F, et al. Joint face detection and alignment using multitask cascaded convolutional networks[J]. IEEE Signal Processing Letters, 2016, 23(10): 1499-1503.

[21] Feng Z H, Kittler J, Awais M, et al. Wing loss for robust facial landmark localisation with convolutional neural networks[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. June 18-23, 2018, Salt Lake City, UT, USA. IEEE, 2018: 2235-2245.

[22] 李文胜. 基于树莓派的嵌入式 Linux 开发教学探索[J]. 电子技术与软件工程，2014(9)：219-220.
Li W S. Research on embedded Linux development teaching based on raspberry PI [J]. Electronic Technology and Software Engineering, 2014(9): 219-220.(in Chinese)

[23] 罗佳伟，孙建梅，徐国旭. 基于树莓派和深度学习技术的智能遥控车的设计与实现[J]. 计算机产品与流通，2018(1)：185-185.

Luo J W，Sun J M，Xu G X. Design and Implementation of intelligent remote control vehicle based on raspberry PI and deep learning technology [J]. Computer Products and Circulation，2018(1)：185-185.(in Chinese)

[24] 宋凯，姚嘉明，李静. 基于树莓派的智能家居控制开关的研究[J]. 电子技术与软件工程，2015(21)：140-141.

Song K，Yao J M，Li J. Research on smart home control switch based on raspberry PI [J]. Electronic Technology and Software Engineering，2015(21)：140-141.(in Chinese)

[25] 刘华，田占生，冯宇飞. 基于树莓派的智能家居语音控制系统[J]. 制造业自动化，2018，40(10)：128-131.

Liu H，Tian Z S，Feng Y F. Intelligent home voice control system based on raspberry Pi[J]. Manufacturing Automation，2018，40(10)：128-131.(in Chinese)

[26] 黄宏敏，张明森，詹瑞典. 基于 FPGA 的 YOLO 网络的分片加速方法[J]. 电子世界，2020(8)：66-67.

Huang H M，Zhang M S，Zhan R D. Sharding acceleration method based on FPGA YOLO Network [J]. Electronics World，2020(8)：66-67.(in Chinese)

[27] 刘勤让，刘崇阳. 利用参数稀疏性的卷积神经网络计算优化及其 FPGA 加速器设计[J]. 电子与信息学报，2018，40(6)：102-108.

Liu Q R，Liu C Y. Calculation optimization for convolutional neural networks and FPGA-based accelerator design using the parameters sparsity[J]. Journal of Electronics and Information Technology，2018，40(6)：102-108.(in Chinese)

[28] 张榜，来金梅. 一种基于 FPGA 的卷积神经网络加速器的设计与实现[J]. 复旦学报(自然科学版)，2018，57(2)：236-242.

Zhang B，Lai J M. Design and implementation of a FPGA-based accelerator for convolutional neural networks[J]. Journal of Fudan University(Natural Science)，2018，57(2)：236-242.(in Chinese)

[29] Xu J. 斯坦福目标检测深度学习指南[J]. 机器人产业，2017(6)：20-26.

Xu J. Stanford's guide to deep learning for target detection [J]. Robotics Industry，2017(06)：20-26.(in Chinese)

[30] He K M，Zhang X Y，Ren S Q，et al. Deep residual learning for image recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR). June 27-30，2016，Las Vegas，NV，USA. IEEE，2016：770-778.

(编辑 陈移峰)