

## 基于 SoC FPGA 和 CNN 模型的动作识别系统设计

陈旭东<sup>1</sup>, 周昱琪<sup>2</sup>, 李梦杰<sup>1</sup>, 陈章进<sup>1,3</sup>

(1. 上海大学 微电子研究与开发中心, 上海 200444;

2. 上海大学 纳米科学与技术研究中心, 上海 200444; 3. 上海大学 计算中心, 上海 200444)

**摘要:** 动作识别是重要的机器视觉研究领域之一。设计实现基于 SoC FPGA 和 CNN 模型的动作识别系统。设计实现流水线型 LK 光流计算、基于 HOG 与 SVM 的行人检测模块; 设计实现动态行人检测与静态行人检测结果的融合算法, 计算出目标人物所在区域; 利用该区域的视频图像和光流场数据, 输入 CNN 模型计算得到目标人物的动作识别结果; 设计指令集架构的 NPU 单元用于实现 CNN 模型计算。整个系统基于 DE10-Nano 开发板进行软硬件协同开发, 能够识别“站立”、“行走”、“挥手”和“下蹲”等动作。该系统具有较高的识别率和较强的设计灵活性, 方便 CNN 模型的扩展与修改。

**关键词:** 动作识别; FPGA; CNN; LK 光流; NPU

**中图分类号:** TN791; TP301

**文献标识码:** A

**DOI:** 10.16157/j.issn.0258-7998.182476

**中文引用格式:** 陈旭东, 周昱琪, 李梦杰, 等. 基于 SoC FPGA 和 CNN 模型的动作识别系统设计[J]. 电子技术应用, 2019, 45(2): 97-104, 108.

**英文引用格式:** Chen Xudong, Zhou Yuqi, Li Mengjie, et al. Design of an action recognition system based on SoC FPGA and CNN model[J]. Application of Electronic Technique, 2019, 45(2): 97-104, 108.

## Design of an action recognition system based on SoC FPGA and CNN model

Chen Xudong<sup>1</sup>, Zhou Yuqi<sup>2</sup>, Li Mengjie<sup>1</sup>, Chen Zhangjin<sup>1,3</sup>

(1. Microelectronic Research and Development Center, Shanghai University, Shanghai 200444, China;

2. Research Center of Nano-Science and Nano-Technology, Shanghai 200444, China;

3. Computer Center, Shanghai University, Shanghai 200444, China)

**Abstract:** Action recognition is an important area in computer vision. An action recognition system based on SoC FPGA and CNN model is presented. A pipelined LK optical flow module and a pedestrian detection module based on HOG and SVM are realized. The results of detection of dynamic pedestrian and static pedestrian are merged to recognize the area where the person is. The image and optical flow in the area are both input into the CNN model, to recognize what action it is. An ISA-NPU is proposed to realize the computation of the CNN model. The system is realized in Altera DE10-Nano Kit. With the co-operation of hardware and software, the system can recognize actions such as stand, walk, wave and squat. The system is accurate in recognition and flexible for expansion and modification of the CNN model.

**Key words:** action recognition; FPGA; CNN; LK optical flow; NPU

## 0 引言

基于视觉的动作识别方法可以分为两大类<sup>[1]</sup>, 其一是采用人工设计的特征进行识别的方法, 比如基于密集轨迹<sup>[2]</sup>、基于空时滤波器与图模型<sup>[3]</sup>、基于关节信息和流形学习的方法<sup>[4]</sup>、基于 HMM 的复杂动作建模<sup>[5]</sup>等; 其二是采用深度网络的方法, 比如基于空时卷积神经网络<sup>[6]</sup>、Two-stream 卷积网络<sup>[7-8]</sup>、基于关节信息和 LSTM 的空时注意力模型<sup>[9]</sup>等。虽然使用深度学习可以减少人工设计特征的工作量, 且在图像识别等领域有极佳的表现<sup>[10]</sup>, 但是却有运算量大的问题。对此, 可以通过模型改进<sup>[11-12]</sup>、压缩<sup>[13]</sup>、量化<sup>[14]</sup>等方式减少运算量, 也可以使用 GPU<sup>[15]</sup>、

ASIC<sup>[16]</sup>和 FPGA<sup>[17-18]</sup>等硬件系统进行加速运算。

本文采用 CNN 模型对视频流中的目标任务进行动作识别。为了减少运算量, 首先利用基于光流信息和静态图像处理的行人检测方案定位场景中的目标, 而后使用 CNN 网络对目标区域的图像和光流进行处理, 最终给出动作识别结果。该系统在 DE10-Nano 开发板上进行开发与验证, 在 FPGA 端实现流水线型 LK 光流计算、基于 HOG 和 SVM 的行人检测、指令集架构的 NPU 处理单元, 在 HPS 端实现目标区域融合求解、NPU 单元调用, 两者共享 DDR 内存, 采用 AXI 总线实现片内通信。

## 1 动作识别原理

### 1.1 LK 光流法

$I(x, y, t)$  表示时刻  $t$  拍摄下的视频图像中坐标  $(x, y)$  点的灰度值, 根据 LK 光流约束<sup>[19]</sup>, 光流场的计算满足下列式子:

$$u^*, v^* = \arg \min_{u, v} \sum_{\Omega} (I_x u + I_y v + I_t)^2 + \lambda (u^2 + v^2) \quad (1)$$

式中,  $u = \partial x / \partial t$ ,  $v = \partial y / \partial t$  分别为光流场在  $x$  和  $y$  方向上的分量,  $I_x$  和  $I_y$  表示图像在  $x$  和  $y$  方向上的梯度, 而  $I_t$  则表示在时间方向  $t$  上的梯度,  $\Omega$  则是坐标  $(x, y)$  的某个较小的邻域, 参数  $\lambda$  是正则化因子。

求解上述优化问题, 可以得到其最优解的形式如下:

$$\begin{bmatrix} u^* \\ v^* \end{bmatrix} = (A + \lambda I)^{-1} B \quad (2)$$

$$\text{式中, } A = \begin{bmatrix} \sum_{\Omega} I_x^2 & \sum_{\Omega} I_x I_y \\ \sum_{\Omega} I_x I_y & \sum_{\Omega} I_y^2 \end{bmatrix}, B = \begin{bmatrix} \sum_{\Omega} I_x I_t \\ \sum_{\Omega} I_y I_t \end{bmatrix}。$$

### 1.2 基于 HOG 和 SVM 的行人检测

通常使用滑动窗口对原始图像进行 HOG 特征提取, 再对 HOG 特征向量进行分类识别, 达到行人检测的目的。图像中  $(x, y)$  坐标点的像素值为  $I(x, y)$ 。该点的像素梯度的大小和方向可以计算为  $M(x, y)$  和  $\Theta(x, y)$ 。将梯度的方向  $\Theta(x, y)$  划分到  $N$  个区间, 每个区间占据  $\pi/N$  角度值, 如图 1 所示。

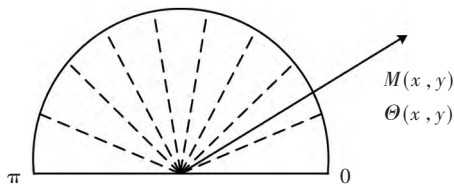


图 1 梯度方向直方图统计示意图

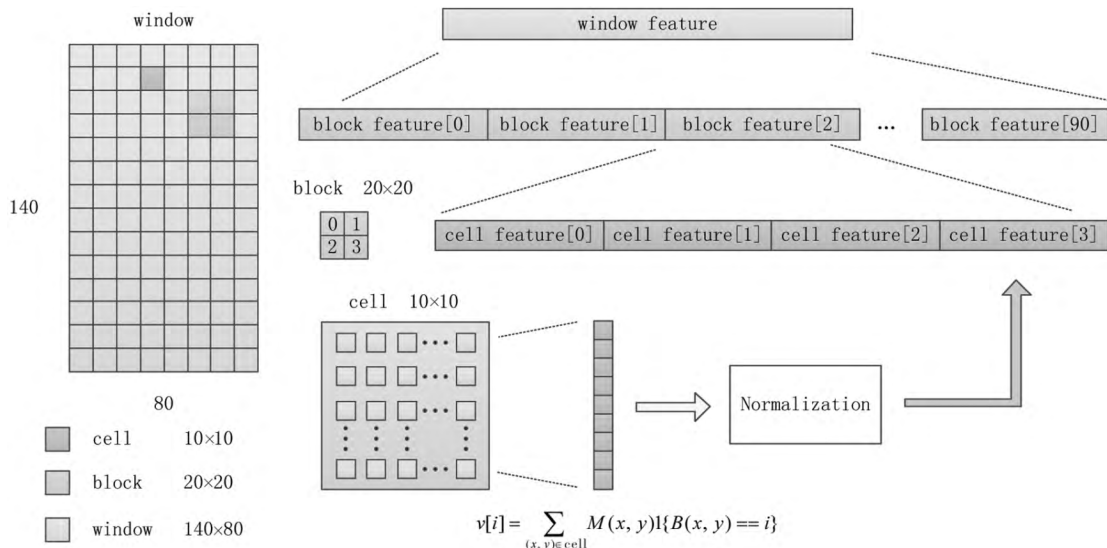


图 2 HOG 特征提取流程示意图

其中:

$$\begin{aligned} I_x(x, y) &\approx I(x+1, y) - I(x-1, y) \\ I_y(x, y) &\approx I(x, y+1) - I(x, y-1) \\ M(x, y) &\approx \sqrt{I_x(x, y)^2 + I_y(x, y)^2} \\ \Theta(x, y) &= \tan^{-1}(I_y(x, y) / I_x(x, y)) \% \pi \\ B(x, y) &= \text{floor}(\Theta(x, y) / \pi \times N) \end{aligned}$$

将检测窗口 (window) 划分成数个元胞 (cell)。在本设计中, 窗口尺寸设定为  $140 \times 80$ , 可被划分成  $14 \times 8$  个  $10 \times 10$  的元胞。首先, 对这  $14 \times 8$  个  $10 \times 10$  的元胞, 依次统计每个元胞里面的梯度直方图; 将其通过规则化模块, 得到最终的 cell 特征; 通过组合  $2 \times 2$  的元胞特征, 可以得到块特征; 最后, 将窗口中的块特征依次排列, 得到最终的  $91 \times 36 = 3276$  维度的窗口特征, 如图 2 所示。

在提取 HOG 特征后, 可以利用 SVM 进行分类识别。本文选择使用线性 SVM 进行分类判别。其判别的依据如式(3)所示。

$$y = \text{sign}(\mathbf{w}^T \mathbf{x} + b) \quad (3)$$

其中,  $\mathbf{w}$  和  $b$  分别是 SVM 的权值和偏置,  $\mathbf{x}$  则是输入的 HOG 特征。最后输出  $y = +1$  说明是正样本 (即行人); 输出  $y = -1$  则说明是负样本 (即非行人)。

### 1.3 动态窗口与静态窗口的融合

使用光流信息和静态图像信息都可以对视频流中的行人进行检测<sup>[20-21]</sup>。通过实验发现, 使用基于 HOG 和 SVM 的静态图像检测方法不易受到物体运动影响, 但是误检率较高; 而基于光流信息的动态检测方法误检率较低, 但是一旦物体长时间静止或者缓慢运动就难以定位目标。因此, 本文提出将两种检测方法相融合的方案, 如图 3 所示。

融合光流和行人检测结果的窗口融合算法流程如下:

- (1) 将 LK 光流结果加框处理, 得出动态行人的框  $B_d$

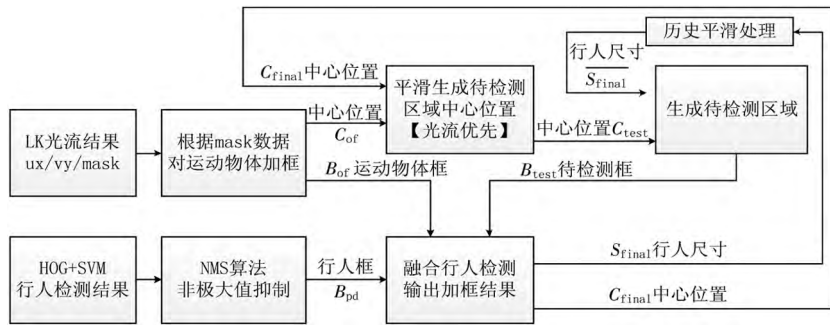


图3 融合光流和行人检测结果示意图

以及动态框的中心位置  $C_{of}$ 。

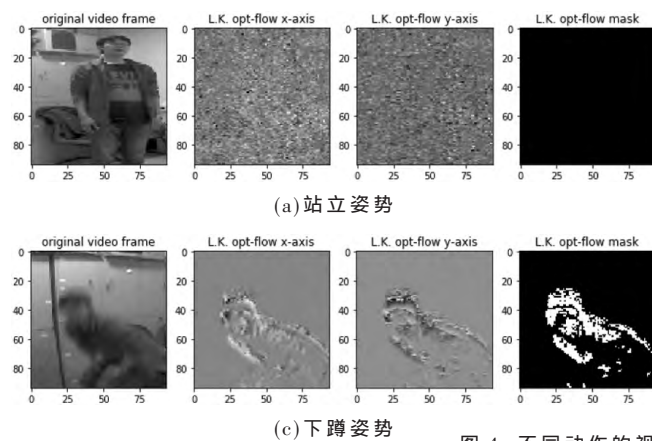
(2)如果  $B_{of} \neq \text{NULL}$ , 即 LK 光流计算结果表明动态框存在, 利用  $C_{final}$  和  $C_{of}$  进行插值平滑, 得到待检测框中心为  $C_{test} = \alpha C_{of} + (1 - \alpha) C_{final}$ ; 否则如果  $B_{of} = \text{NULL}$ , 说明动态框不存在, 那么待检测框的中心为最近一次检测中心, 即  $C_{test} = C_{final}$ 。

(3)对  $S_{final}$  进行时间域上的平滑处理, 得到待检测框的尺寸  $S_{test} = \overline{S_{final}}$ 。

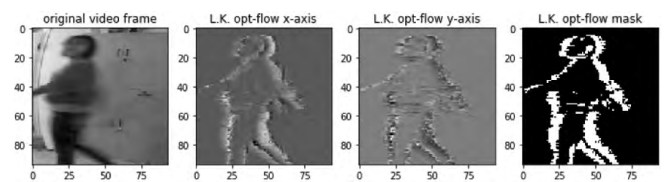
(4)利用  $C_{test}$  和  $S_{test}$  得出待检测框为  $B_{test}$ 。

(5)对基于 HOG 和 SVM 的行人检测打框结果, 进行 NMS 非极大值抑制, 得到静态行人 (相对粗糙) 的框  $B_{pd}$ 。

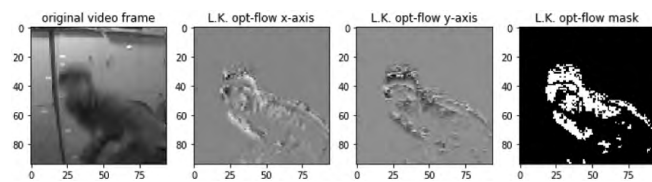
(6)如果  $\text{IoU}(B_{pd}, B_{test}) = \frac{S(B_{pd} \cap B_{test})}{S(B_{pd} \cup B_{test})} > \delta$ , 即静态行人框  $B_{pd}$  和待检测框  $B_{test}$  有较大的重合范围, 说明静态检测结果正确, 此时需要将静态框  $B_{pd}$  与动态框  $B_{of}$  进行融合得到最后的  $B_{final} = B_{of} \cup B_{pd}$ , 并且更新行人框的中心为  $C_{final}$ , 尺寸为  $S_{final}$ 。



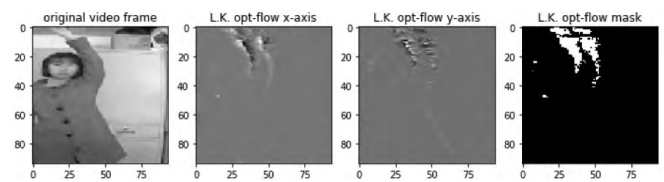
(a) 站立姿势



(b) 行走姿势



(c) 下蹲姿势



(d) 挥手姿势

图4 不同动作的视频图像与光流数据示意图

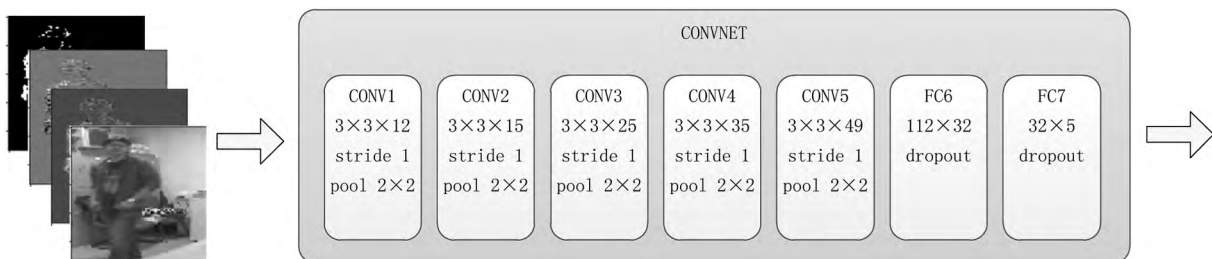


图5 使用CNN模型识别目标人物的动作

通过上述窗口融合算法, 可以得到目标人物在视频帧中的位置。针对这个区域的视频图像和光流数据进行采集, 对于不同的动作有不同的数据形态。如图4所示, 每个动作对应的四张图分别是原始视频图像、光流  $x$  轴分量、光流  $y$  轴分量和显著运动检测掩膜。

#### 1.4 基于 CNN 的动作识别

计算每一帧视频的光流, 提取行人存在区域的视频图像、光流场  $x$  轴分量、 $y$  轴分量以及显著运动检测掩膜。将这四类图

像信息统一形变到  $94 \times 94$  大小, 并作为 CNN 的输入通道, 交由 CNN 进行卷积层和池化层运算; 最后的结果经过全连接层计算得到姿势识别结果。其中, 卷积核的尺寸统一设置为  $3 \times 3$  大小, 池化核的尺寸统一为  $2 \times 2$ ; 而为了提高 CNN 的泛化性能, 在全连接层使用了 dropout, 并对 CNN 中的卷积核、全连接权值都添加 L2 正则化。使用 CNN 模型识别目标人物的动作如图5所示。

#### 2 动作识别系统的设计与实现

##### 2.1 软硬件协同开发示意图

本文提出的动作识别系统发挥了 SoC FPGA 的软硬件协同处理能力。

将运算量巨大的光流计算、静态行人检测和 CNN 运算放置在 FPGA 端, 充分发挥其并行计算、流水线型处理的运算能力; 而将运算需求较低的窗口融合算法在 HPS 端实现, 使其发挥 C 语言程序设计的便捷性。HPS 和 FPGA 之间通过 AXI 桥实现通信。而为了训练 CNN 模



型,在 HPS 端增加数据采集模块,将样本保存到 ima 文件,通过网络传输到 PC。针对站立、挥手、下蹲和行走四种姿势分别采样,其中存在一定数量的“离群值”,为了更好地训练 CNN,在 MATLAB 上实现了可视化样本筛选程序。所有样本和标签代入 TensorFlow 框架,进行 CNN 模型训练;将训练得到的 CNN 模型参数和 CNN 模型结构传输到 HPS 端,HPS 会将 CNN 参数加载到 DDR 内存,并将 CNN 模型对应的 NPU 指令发送到 FPGA 端的指令集架构 NPU 单元的指令 Cache 中。一旦 HPS 将目标窗口融合完成,可以发送一次 CNN 运算启动的指令,等待 FPGA 端 NPU 运算完成,读取 DDR 中的判别结果。动作识别系统框架图如图 6 所示。

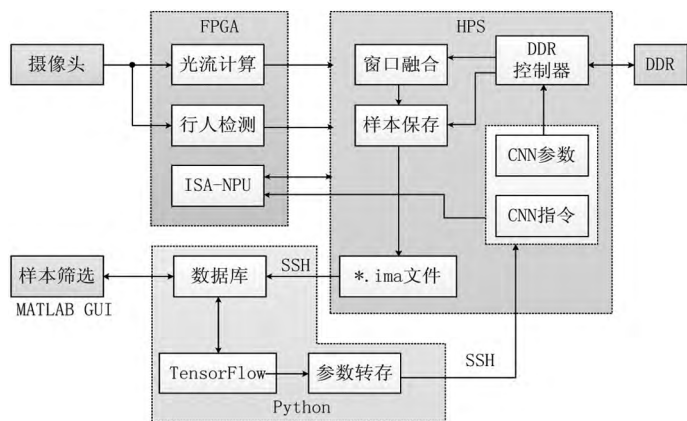


图 6 动作识别系统框架图

## 2.2 硬件系统框架

整个动作识别系统的硬件部分框架的细节图如图 7 所示。DDR 存储空间为 1 GB, 被划分为 7 个部分。其中, Linux 操作系统占用 0~480 MB 空间; 光流计算结果占用

480 MB~512 MB 这 32 MB 空间; 原始视频流数据占用 512 MB~544 MB 空间; 行人检测的结果占用 576 MB~608 MB 空间; 行人加框视频占用 608 MB~640 MB 空间; 而将 640 MB~1 024 MB 空间用于 NPU 运算过程中的数据缓存。

摄像头 MT9D111 的数据进入 FPGA 后,会同时传输到 LK 光流计算模块、行人检测模块,计算结果会缓存到 DDR 中;而通过视频缓存与显示模块,原始视频可以存储到 DDR 中,同时 HDMI 视频输出所需的数据从 DDR 中获取。这些模块对 DDR 的读写优先级较高,且占用 DDR 带宽较大,使用 FPGA-to-SDRAM (F2S) 接口直接和 HPS 上的 SDRAM 控制器交互。

HPS 端的 C 程序需读取行人检测结果, 并使用 NMS 算法对行人检测的框进行聚合、优化。调用 `memcpy()` 将原始视频复制到加框视频的内存空间, 并加上打框的结果; 同时使用通过 HPS-to-FPGA (H2F) 接口传输指令, 启动 FPGA 端 NPU 的运算, 并读取 NPU 运算状态。

NPU 运算时会对 DDR 进行读写访问, 该访问通过 FPGA-to-HPS(F2H)接口和 L3 互连网络实现。NPU 单元可以计量指令执行时间, 将运算状态与执行时间写入到状态寄存器中, 随后 HPS 可以通过 Light Weight HPS-to-FPGA(H2F-LW)接口获取 NPU 运行状态信息。

### 2.3 流水线 LK 光流计算模块设计

本文设计的流水线 LK 光流计算框架如图 8 所示。

每当 MT9D111 完成一帧图像传输(VSYNC 下降),或者一行传输完成(HSYNC 下降),都会启动从 DDR 中读取一行的视频数据缓存到 FIFO 内,生成  $I(x, y, t-1)$ ;而后,MT9D111 输出的像素点分别进入长度为 1 行像素点数量和长度为 1 的移位寄存器,得到  $I(x, y-1, t)$  和  $I(x-1,$

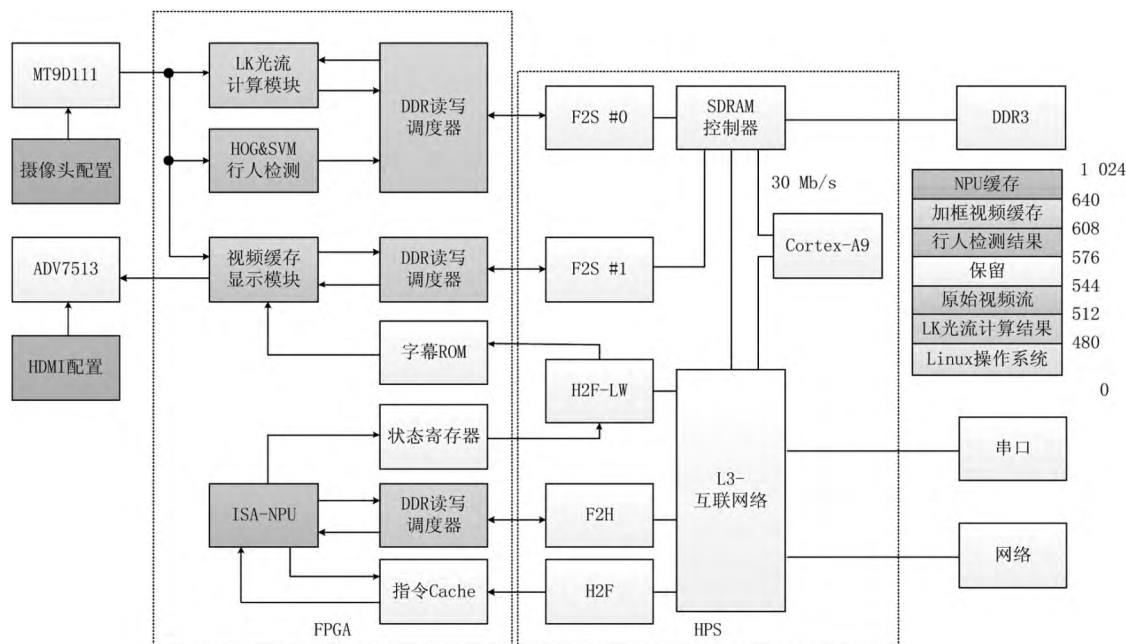


图 7 动作识别硬件系统框架示意图

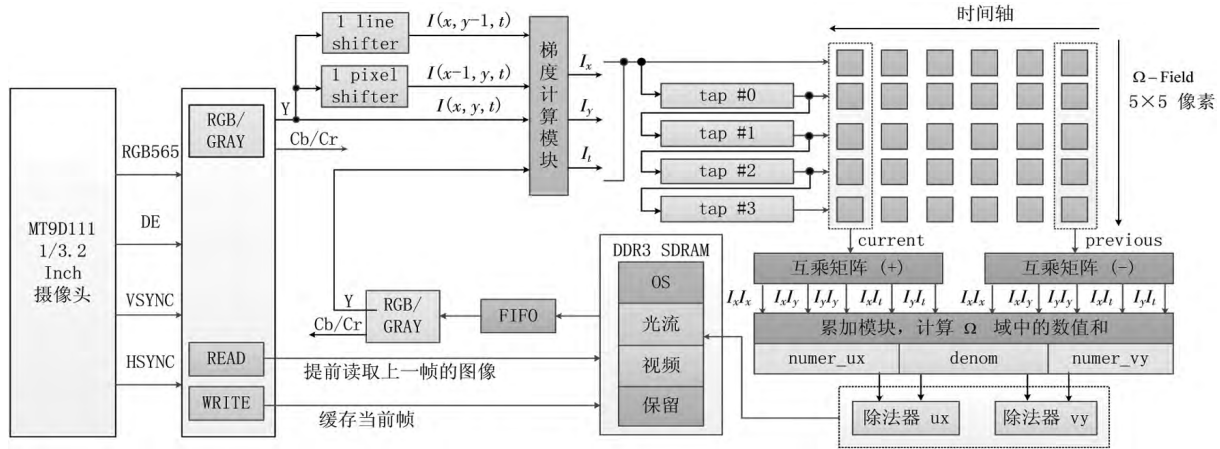


图 8 流水线 LK 光流计算模块的内部框图

$y, t$ ); 将  $I(x, y, t), I(x-1, y, t), I(x, y-1, t)$  和  $I(x, y, t-1)$  输入到梯度计算模块, 得到在  $x, y$  和  $t$  方向上的一阶差分分为  $I_x, I_y$  和  $I_t$ ; 使用 4 个长度一致为一帧图像一行像素点数量的移位寄存器, 以及  $5 \times 6$  个寄存器阵列构造 LK 光流法计算区域, 即  $\Omega$  域。使用互乘矩阵模块, 分别计算寄存器阵列中 previous 对应数据  $I_{x, prev}, I_{y, prev}$  和  $I_{t, prev}$  的乘法交叉项; 同理, 计算出 current 对应数据  $I_{x, cur}, I_{y, cur}$  和  $I_{t, cur}$  的乘法交叉项。 $\Omega$  域内乘法交叉项可以更新为:

$$\sum_{\Omega} I_m I_n := \sum_{\Omega} I_m I_n + I_{m, cur} I_{n, cur} - I_{m, prev} I_{n, prev} \quad (4)$$

其中,  $m, n \in \{x, y, t\}$ 。

结合光流场计算式(2), 可以通过流水线型除法器实现光流场的计算。最后将运算结果回写到 DDR 中的 480 MB~512 MB 空间。

#### 2.4 指令集架构 NPU 设计

为了便于实现、修改 CNN 架构, 本文设计了基于指令集架构的 NPU 处理单元, 能够执行矩阵运算(如 ADD、SUB、MULT、DOT 等)、立即数运算(ADDi、MULTi 等)、2-D 图像处理(如卷积 CONV、池化 POOL 等)、激活函数(如 SIGM、TANH、ReLU 等)。指令集架构 NPU 结构框图如图 9 所示。通过不同 NPU 指令组合, 能够实现不同 CNN 结构。NPU 指令设计参考 RISC 格式, 如表 1 所示, 每条指

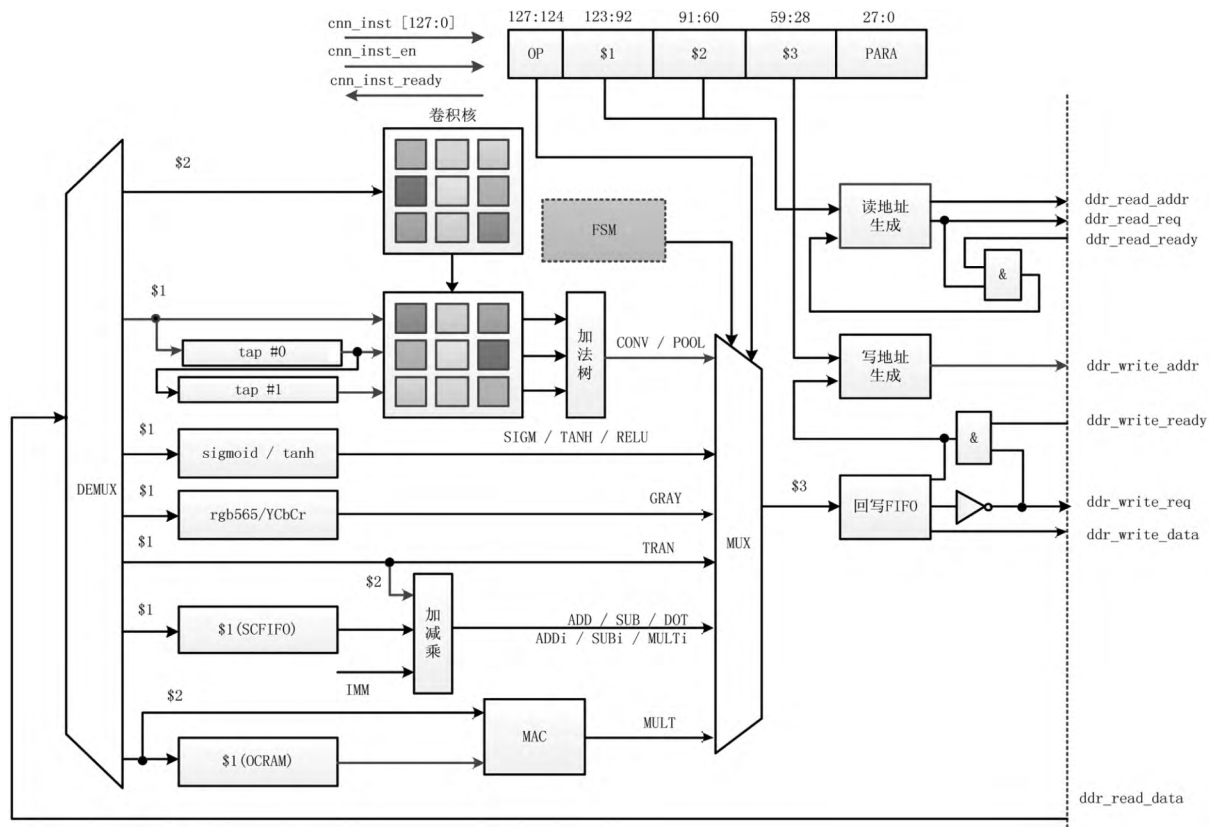


图 9 指令集架构 NPU 结构框图

表 1 NPU 指令格式

127:124	123:92	91:60	59:28	27:0
指令	参数\$1	参数\$2/	输出\$3	运算
类型	首地址	立即数	首地址	参数

令是 128 bit 长度, 其中高 4 bit 区分指令类型,[123:92]表示参数\$1 内存首地址,[91:60]表示参数\$2 的内存首地址或者立即数 IMM,[59:28]表示运算结果\$3 的内存首地址,[27:0]用于表示运算参数,如函数输入矩阵尺寸、卷积核尺寸等。

比如全连接层的 NPU 指令可以表述如下。将存储在首地址 0x0F000000 的输入矩阵\$1(尺寸为 1×112)与存储在 0x0A6B0000 的权值矩阵\$2(尺寸为 112×32)相乘,结果缓存在 0x0C000000;而后加上存储在 0x0A6C0000 的偏置(尺寸为 1×32),缓存到 0x0C010000;最后通过 SIGM 指令实现非线性映射,结果保存到 0x0E000000。

```
Layer 6: fully_connection
MULT, @0F000000, @0A6B0000, @0C000000, M=1, N=112,
P=32
inst=40F0000000A6B00000C0000000170200
ADD, @0C000000, @0A6C0000, @0C010000, M=1, N=32
inst=00C0000000A6C00000C0100000120000
SIGM, @0C010000, xx, @0E000000, M=1, N=32
inst=90C010000000000000E0000000120000
```

整个 NPU 划分成 FSM 控制逻辑、NPU 指令解析逻辑、NPU 指令运算逻辑和 DDR 读写接口逻辑四个部分。

如果 NPU 就绪, 那么会给出 cnn\_inst\_ready 信号, 表示可以接收运算指令; cnn\_inst 信号则是外部控制逻辑给出的运算指令; cnn\_inst\_en 信号则表示 cnn\_inst 指令有效使能。

DDR 读写接口由 ddr\_write\_\* 和 ddr\_read\_\* 两组信号构成, 且都符合 Avalon-MM 接口协议。由 \*\_addr 给出读写地址, \*\_req 给出读写请求, 对于 DDR 写入使用 ddr\_write\_data 表示写入数据。读写过程中信号都要保持稳定, 直到 DDR 控制器给出 \*\_ready 读写请求完成信号。对于 DDR 读取, ddr\_read\_data 表示读取的数据, 而 ddr\_read\_data\_valid 则表示 DDR 读取数据有效。

## 2.5 软件设计架构

HPS 中的软件程序框架图如图 10 所示。其中分别实现了下述功能模块。

(1)初始化接口,使用 mmap()函数将 HPS-to-FPGA 接口和 DDR 物理内存映射到 Linux 用户空间,方便其他模块访问、读写 FPGA 和 DDR 内的数据;

(2)按键响应模块,一旦用户按下键盘上的按键,程序能够及时响应;

(3)目标窗口生成模块,加载光流数据和静态图像行人检测结果,生成动态窗口和静态窗口,并通过窗口融合算法对其进行融合,确定目标所在的位置;



图 10 HPS 端 C 语言软件设计框图

(4)数据采集模块,从 DDR 中采集视频、光流等数据,并保存到相应的文件中,用于 CNN 的离线学习使用;

(5)NPU 指令传输和状态监控模块,能够通过 HPS-to-FPGA 接口传输 NPU 指令到 FPGA 端的指令 cache 中;并且能够启动 NPU 的运算、监控 NPU 的指令执行情况;

(6)绘图模块,能够将窗口融合结果保存到 DDR 中的内存区间,便于 HDMI 输出显示。

## 3 动作识别系统测试

### 3.1 CNN 参数训练

为了训练本文的 CNN 模型,对于“无人”、“挥手”、“下蹲”、“站立”和“行走”这 5 种情况分别采集 600 个训练样本,共计 3 000 个样本。采用深度学习框架 Tensor-Flow 对 CNN 模型参数进行训练<sup>[21]</sup>。整个训练过程在 Nvidia GeForce 940M 显卡进行,大约消耗 3 小时,识别精度可以在训练集达到 90%,在测试集达到 88%。对于不同的动作,训练结果如表 2 所示。本文采用的动作识别方法能够较好识别“挥手”、“站立”和“行走”这三种姿势;而对于“无人”、“下蹲”这两个情况检出率较低,但是误报率也相对较低。

表 2 CNN 训练效果

		无人	挥手	下蹲	站立	行走
训练	TP	0.73	0.97	0.72	1.00	0.98
	FP	0.00	0.02	0.01	0.04	0.06
测试	TP	0.74	0.93	0.76	0.97	0.90
	FP	0.00	0.05	0.01	0.03	0.05

### 3.2 硬件资源消耗与运行性能

由于本文设计的 LK 光流模块、基于 HOG 和 SVM 的行人检测模块、指令集架构 NPU 单元都是采用流水线处理,数据吞吐量极大。各模块的资源消耗与理论最大性能如表 3 所示。

选择 DE10-Nano 开发板作为软硬件运行平台,在 Quartus II 14.0 环境下对动作识别系统进行综合、布局布线,工程消耗 FPGA 资源如表 4 所示。

布局布线后各模块分布情况如图 11 所示。

### 3.3 实际运行测试

对本文提出的动作识别系统进行运行测试。将开发板和摄像头 MT9D111、HDMI 显示器进行连接,如图 12



表 3 各模块的资源消耗与理论最大性能

	性能	LK 光流	HOG+SVM	NPU
速度	Fmax	118.6 MHz	76.8 MHz	79.6 MHz
	Period	800×600	800×600	814,177
	FPS(max)	247	160	97.7
资源	ALMs	7 799	14 105	3 530
	Memory Bits	188 202	1 167 512	73 984
	DSPs	41	16	43

表 4 系统总资源消耗

	ALMs	Memory Bits	DSPs
资源使用	37 933(91%)	2 513 876(44%)	106(95%)

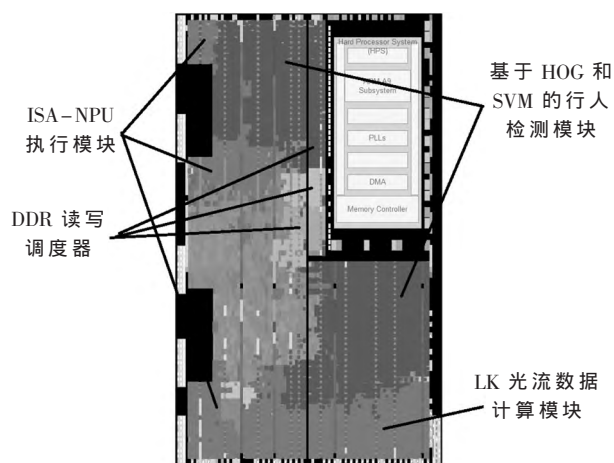


图 11 布局布线后各模块分布情况

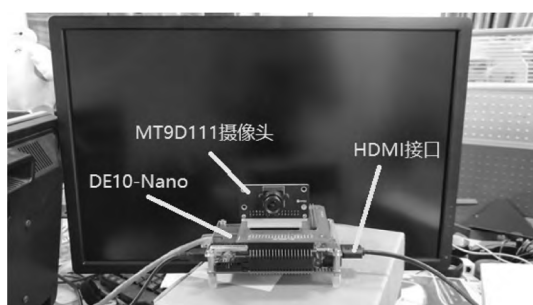


图 12 布局布线后各模块分布情况

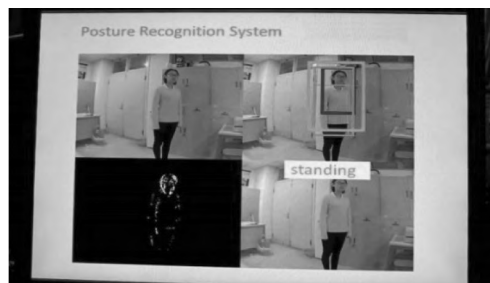
所示。

针对“站立”、“行走”、“下蹲”和“挥手”等不同的人物动作进行识别,结果输出到显示器。其中,左上角为原始视频;左下角为光流计算结果;右上角为行人加框结果;右下角为动作识别结果。系统运行测试情况如图 13 所示。

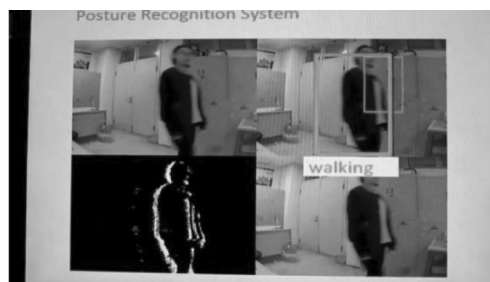
#### 4 结论

本文提出并实现了一种基于 SoC FPGA 和 CNN 模型的动作识别系统。该系统具有流水线型运算结构,能快速地执行光流计算和静态的行人检测;同时,指令集架构 NPU 的设计可以很方便地适应于多种 CNN 模型结构;该姿势识别系统在训练和实际识别测试中都表现出较高的准确率。

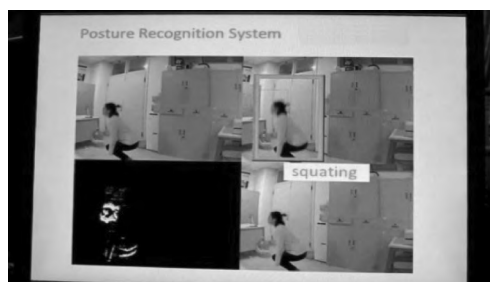
《电子技术应用》2019年 第45卷 第2期



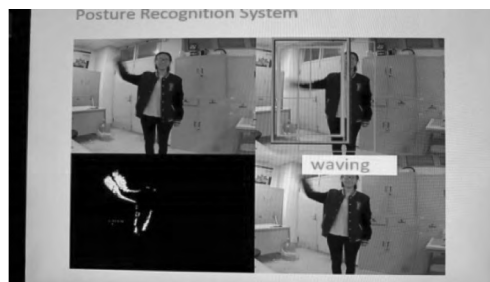
(a)站立姿势



(b)行走姿势



(c)下蹲姿势



(d)挥手姿势

图 13 动作系统实际运行测试情况

#### 参考文献

- [1] 范嘉义.基于骨架信息的人体动作识别[D].合肥:中国科学技术大学,2017.
- [2] WANG H, KLÄSER A, SCHMID C, et al. Action recognition by dense trajectories[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2011: 3169-3176.
- [3] TURAGA P, CHELLAPPA R, SUBRAHMANYAN V S, et al. Machine recognition of human activities: a survey[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2008, 18(11): 1473-1488.
- [4] 王鑫, 沃波海, 管秋, 等. 基于流形学习的人体动作识别[J].

中国图象图形学报, 2014, 19(6): 914-923.

- [5] 战青卓, 王大东. 基于深度神经网络的人体动作识别研究[J]. 智能计算机与应用, 2018, 8(2): 151-154.
- [6] VAROL G, LAPTEV I, SCHMID C. Long-term temporal convolutions for action recognition[J]. IEEE Trans on Pattern Anal & Mach Intell, 2015, PP(99): 1-1.
- [7] SILVA V D O, VIDAL F D B, ROMARIZ A R S. Human action recognition based on a two-stream convolutional network classifier[C]. IEEE International Conference on Machine Learning and Applications. IEEE, 2017: 774-778.
- [8] 王亮亮. 基于视觉的人体动作识别研究[D]. 哈尔滨: 哈尔滨工业大学, 2017.
- [9] SONG S, LAN C, XING J, et al. An end-to-end spatio-temporal attention model for human action recognition from skeleton data[J]. arXiv: 1611.06067, 2016.
- [10] 周凯龙. 基于深度学习的图像识别应用研究[D]. 北京: 北京工业大学, 2016.
- [11] GIRSHICK R. Fast R-CNN[J]. Computer Science, 2015.
- [12] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal

networks[C]. International Conference on Neural Information Processing Systems. MIT Press, 2015: 91-99.

- [13] HAN S, KANG J, MAO H, et al. ESE: efficient speech recognition engine with sparse LSTM on FPGA[J]. arXiv: 1612.00694, 2017.
- [14] 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.
- [15] ABADI M, AGARWAL A, BARHAM P, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems[J]. arXiv: 1603.04467, 2016.
- [16] LIU S, DU Z, TAO J, et al. Cambricon: an instruction set architecture for neural networks[C]. International Symposium on Computer Architecture. IEEE Press, 2016: 393-405.
- [17] 王思阳. 基于 FPGA 的卷积神经网络加速器设计[D]. 成都: 电子科技大学, 2017.
- [18] GUO K, SUI L, QIU J, et al. Angel-Eye: a complete design flow for mapping CNN onto embedded FPGA[J].

(下转第 108 页)

(上接第 96 页)

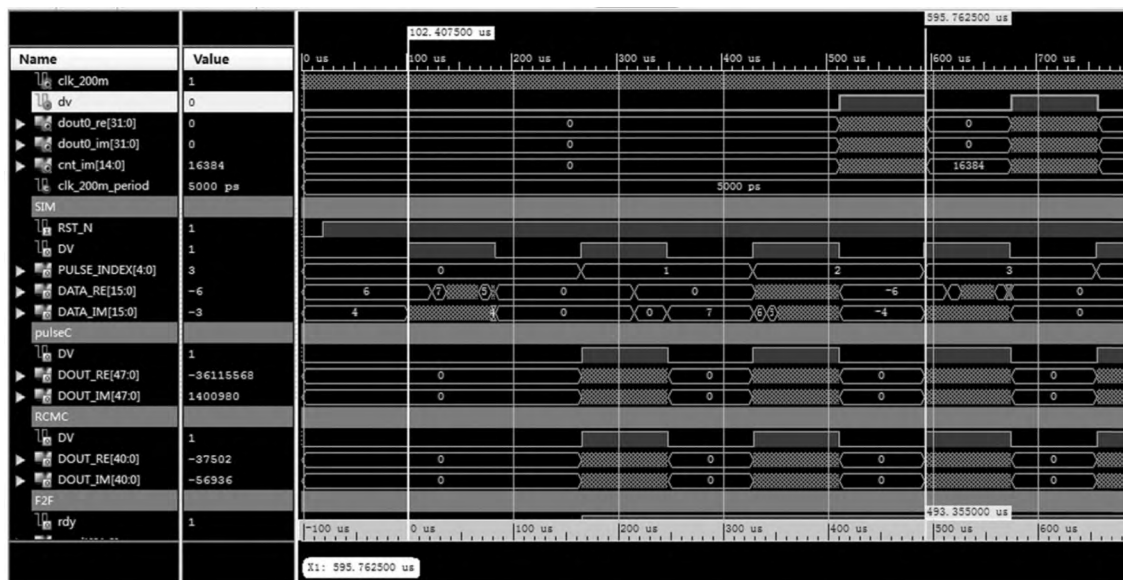


图 8 MODELSIM 仿真波形

法具有实时性高、处理速度快、精度高等特点, 为雷达导引头提供了很好的软硬件支持, 提高了雷达导引头的探测性能。

## 参考文献

- [1] 尚炜. 某相参末制导雷达信号处理机的理论分析及实现[D]. 西安: 西安电子科技大学, 2009.
- [2] 宋万均, 张安堂. 双基地雷达目标速度计算的 FPGA 实现[J]. 电子技术应用, 2014(1): 47-49.
- [3] 王龙, 杨承志, 吴宏超, 等. 基于 FPGA 的数字基带多模雷达信号源设计[J]. 电子技术应用, 2016(8): 87-90.

- [4] CUMMING I G, WONG F H. 合成孔径雷达成像算法与实现[M]. 洪文, 等, 译. 北京: 电子工业出版社, 2012.

(收稿日期: 2018-08-23)

## 作者简介:

侯凯强 (1991-), 男, 硕士研究生, 助理工程师, 主要研究方向: 雷达信号处理。

施君南 (1988-), 男, 博士研究生, 工程师, 主要研究方向: 雷达信号处理。

许彦章 (1983-), 男, 硕士研究生, 工程师, 主要研究方向: 雷达信号处理。



现场实验结果的 6 组数据如表 1 所示, 制动距离  $L_2$  和制动距离  $L_1$  具有较好的一致性, 制动距离  $L_1$  较制动距离  $L_2$  略微偏大, 这主要是由于防坠安全器开始动作时离心制动块与锥毂进行刚性契合, 契合过程中升降机会产生间隙性短暂空滑, 造成积分运算得到的  $L_1$  相对于真实制动距离必然偏大, 应该说角度传感器测得的制动距离  $L_2$  更加准确, 更能反映防坠安全器的真实制动性能。

表 1 实验结果数据表

防坠安全器 编号	次 数	制动 距离 $L_1/m$	制动 距离 $L_2/m$	绝对 误差/m	相对 误差/%	动作 速度/ (m/s)
YT401709080006	1	0.588	0.584	-0.004	-0.68	1.008
YT401709080006	2	0.681	0.673	-0.008	-1.17	1.032
H5112	3	0.355	0.349	-0.006	-1.69	1.056
H5112	4	0.510	0.505	-0.005	-0.98	1.078
B220302	5	0.654	0.644	-0.010	-1.53	1.061
B220302	6	0.609	0.600	-0.009	-1.48	1.037

注: 制动距离  $L_1$  为采用速度计算得到的防坠安全器制动距离, 制动距离  $L_2$  为采用角度计算得到的防坠安全器制动距离, 绝对误差  $= L_2 - L_1$ , 相对误差  $= (L_2 - L_1) / L_1$ 。

## 5 结论

施工升降机防坠安全器嵌入式检测系统是以传感器器件作为检测基础的嵌入式检测设备, 基于 Android 平台及 Java 编程语言设计了上位机分析软件, 以串口通信的方式实现传感器和上位机的数据传输。本检测系统对于制动距离的理论测量精度约为  $\pm 1$  mm, 完全可以满足施工升降机防坠安全器的检测要求。

## 参考文献

- [1] 冯功斌, 段红莉. 施工升降机防坠试验方式探讨[J]. 建筑机械, 2016(4): 85-86.
- [2] 卢立东, 马俊. 便携式施工升降机防坠安全器检测装置[J]. 建筑机械, 2013(21): 113-114.
- [3] 马溢坚, 黄胜红, 陈巍. 防坠安全器现场检测装置信号处理方法[J]. 起重运输机械, 2017(8): 37-41.
- [4] 杨义, 丁涛. 两种典型防坠安全器检测方法的比较研究[J]. 湖北工程学院学报, 2016, 36(3): 87-89.

(上接第 104 页)

- IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017, PP(99): 1-1.
- [19] BARRON J L, FLEET D J, BEAUCHEMIN S S, et al. Performance of optical flow techniques[J]. International Journal of Computer Vision, 1994, 12(1): 43-77.
- [20] 谢红, 原博, 解武. LK 光流法和三帧差分法的运动目标检测算法[J]. 应用科技, 2016, 43(3): 23-27, 33.
- [21] LI J, YIN Y, LIU X, et al. 12,000-fps Multi-object detection using HOG descriptor and SVM classifier[C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2017: 5928-5933.

- [5] 李良, 朱善安. 基于 Java 的串口通信[J]. 电子器件, 2007 (2): 714-716, 720.
- [6] 陈小贝, 汪志成, 叶宝安. 基于嵌入式系统的 PLC 触摸屏控制系统设计[J]. 电子测量技术, 2015, 38(3): 54-56.
- [7] 秦天柱, 张伟刚, 瞿少成. 基于 Modbus 协议的多路数据采集器[J]. 电子测量技术, 2017, 40(11): 175-178.
- [8] 单星, 林晓焕, 郭丹蕊, 等. 基于 ModBus 协议的负载模拟系统设计[J]. 国外电子测量技术, 2015, 34(7): 72-74.
- [9] 杜颖财, 王希军, 王树洁, 等. 增量式编码器自动检测系统[J]. 电子测量与仪器学报, 2012, 26(11): 993-998.
- [10] 方天恩, 漆晶, 马金辉, 等. 基于 Android 的智能蓝牙温度计的研究与设计[J]. 电子技术应用, 2017, 43(3): 73-76.
- [11] 吕骏, 王仁波, 汤彬, 等. 基于 USB 总线多点数据采集系统的设计[J]. 电子测量技术, 2015, 38(11): 88-91.
- [12] 方勇军, 赵红旗, 骆星九, 等. 基于 Android 平台的心电数据采集与传输系统设计[J]. 仪表技术, 2014(4): 36-37, 42.
- [13] Zou Xiaojia, You Xiangdong, Pan Hao, et al. User interface design and implementation for electricity operation information system based on Android[C]. Applied Mechanics and Materials, 2014, 3365(602).
- [14] Wei Zhiqiang, Wang Xiaowen, Jia Dongning, et al. An information collection and transmission strategy of vehicle state-aware system based on OBD technology and Android mobile terminals[C]. Applied Mechanics and Materials, 2015, 3749(719).

(收稿日期: 2018-08-06)

## 作者简介:

孙操 (1990-), 男, 硕士研究生, 主要研究方向: 可编程器件与系统设计。

刘士兴 (1969-), 通信作者, 男, 博士, 副教授, 主要研究方向: 光电探测技术、可编程器件与系统设计, E-mail: liusx@hfut.edu.cn。

宋亚杰 (1992-), 男, 硕士研究生, 主要研究方向: 可编程器件与系统设计。

- [22] ZHANG M, XU H, WANG X, et al. Application of Google TensorFlow machine learning framework[J]. Microcomputer & Its Applications, 2017.

(收稿日期: 2018-08-25)

## 作者简介:

陈旭东 (1994-), 男, 硕士研究生, 主要研究方向: 微电子学与固体电子学、雷达信号处理、芯片设计和系统集成。

周昱琪 (1993-), 女, 硕士研究生, 主要研究方向: 锂电池设计、图像识别、数字信号处理。

陈章进 (1968-), 通信作者, 男, 教授, 硕士生导师, 主要研究方向: 微电子及集成电路设计、多媒体支持下高清显示, E-mail: zjchen@staff.shu.edu.cn。