

Uma Proposta de Rastreamento de Objetos Utilizando Computação em Borda

Fernando L. Oliveira¹, Mateus S. Delatorre², Heberth P. Reinstein²

¹ Programa de Pós-Graduação em Computação – PPGC
Centro de Desenvolvimento Tecnológico – CDTec
Universidade Federal de Pelotas (UFPel)
Pelotas – RS – Brasil

² Grupo de pesquisa Lardev – Instituto Federal Farroupilha (IFFar)
São Borja – RS – Brasil

fernando.oliveira@inf.ufpel.edu.br

{mateus.2020005222,heberth.2017007299}@aluno.iffar.edu.br

Abstract. *The science and technology field has been advancing over time, and people surround themselves with computer systems, which help them in their routines. This paper presents the results of developing an approach to detect and track people at edge computing. The system runs on a Raspberry Pi 3B board and, with the support of a camera, monitors the environment. An interface was built that analyses the flow of people using a particular staircase. Information such as calories expended, effort, or activity time are offered to users in return for using the stairs.*

Resumo. *Muito se tem avançado nos campos da ciência e tecnologia, as pessoas cercam-se por sistemas computacionais, os quais as auxiliam em suas rotinas. O presente artigo, apresenta os resultados do desenvolvimento de uma abordagem para detectar e rastrear pessoas na borda. O sistema é executado em uma placa Raspberry Pi 3B e, com apoio de uma câmera monitora o ambiente. Construiu-se uma interface que avalia o fluxo de pessoas que utilizam uma determinada escada de alvenaria. Informações como: calorias gastas, esforço ou tempo de atividade são oferecidos aos usuários em contrapartida pelo uso das escadas.*

1. Introdução

Os sistemas computacionais fazem parte do cotidiano da sociedade contemporânea. Ajudam as pessoas em suas atividades diárias, seja através de um computador, celular ou dispositivos eletrônicos com capacidade de processamento. Tais dispositivos quando combinados com outros equipamentos ou como parte de um sistema maior e mais complexo, podem ser entendidos como sistemas embarcados [Brisolara and Matos 2009]. Os sistemas embarcados são elementos com propósitos específicos, dentro de um sistema maior, podem atender as mais diversas finalidades, como por exemplo, um sistema de frenagem de um veículo (ABS) até o controle de altitude de um avião [Wolf 2012].

Geralmente, os sistemas embarcados precisam atender a inúmeras demandas e, ao mesmo tempo, serem eficientes em suas tarefas, isto é, um *smartphone* além de realizar

ligações, navegar na internet e tirar fotos, deve manter uma boa relação de custo benefício entre as funcionalidades executadas e o consumo de bateria [Brisolara and Matos 2009]. Por si só, eles enfrentam inúmeras restrições, associados ao contexto no qual estão inseridos, pelas limitações físicas de *hardware* que influenciam, inclusive, na parte lógica das soluções, na programação, processamento e armazenagem de dados [Heath 2002].

Neste sentido, o desenvolvimento e projeto de soluções embarcadas se revela, naturalmente, um desafio que envolve não somente aspectos físicos, mas sobretudo, aspectos relacionados ao contexto. As soluções cada vez mais, caminham para a sensibilidade e adaptação ao ambiente no qual estão inseridas, com intuito de dar suporte proativo ao usuário final [Knappmeyer et al. 2013]. Nessa prática, destacam-se os sistemas baseados em visão computacional.

A visão computacional compreende a técnica de reproduzir, através de software e hardware, aspectos relacionados a visão humana. De forma simplificada, essa abordagem possibilita, dentre outras coisas, processamento de imagens, reconhecimento de objetos, analisar ambientes, determinar tamanho e velocidade de objetos, otimizar processos na indústria que exigem algum controle visual, identificar e rastrear pessoas entre outros [Forsyth and Ponce 2002].

Neste contexto, com base nos princípios da computação pervasiva, desenvolveu-se um sistema utilizando computação em borda, o qual é capaz de detectar e rastrear pessoas. O sistema proposto, atua de forma transparente ao usuário, isto é, não necessita de nenhum dispositivo sobressalente para funcionar. O objetivo geral é coletar informações referentes ao deslocamento de pessoas ao subirem ou descerem escadas em prédios. Tal mecanismo, é parte de um projeto de pesquisa, ainda em desenvolvimento, de um dispositivo para gerar energia através do uso de escadas (seção 3).

Para o desenvolvimento do protótipo, adotou-se um computador de placa única (*Single Board Computer* - SBC), em particular, foi utilizado uma placa Raspberry Pi 3B. Aponta-se como contribuições deste artigo: (i) Demonstração de uma técnica combinatória de algoritmos para detectar e rastrear pessoas; (ii) Apresentação de uma abordagem para contabilizar pessoas dentro de uma região específica e, (iii) uma proposta de cálculo de gasto calórico associado a utilização de escadas.

O restante desse artigo segue organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta a visão geral do projeto UpStairs. Na Seção 4 são descritos os procedimentos metodológicos. A Seção 5 apresenta os resultados preliminares e, por fim, são elencadas as conclusões.

2. Trabalhos relacionados

Os autores Lee e colaboradores [Lee et al. 2019], elaboraram um mecanismo de detecção e rastreamento de pessoas dividido em 3 partes: O detector Brisk-YOLO que consiste em uma ramificação da rede R-CNN Tiny-YOLO, contendo menor número de *kernels* de convolução, uma segunda parte que trata da redução do tamanho característico dos maps e, por fim, uma nova estrutura chamada de *Blaze layer*. Para o rastreamento é utilizado o *Kernelized Correlation Filters*(KFC); caso o alvo seja perdido é usado a técnica da cor de histograma.

No trabalho [Østby 2018] foi feita a proposta de criar uma rede CNN base-

ada na meta-arquitetura R-CNN[Ren et al. 2015], especializada para detectar objetos específicos, com a característica de realizar um rápido pré-processamento em busca de áreas que podem conter o objeto. Essa técnica é conhecida como *region of interest(ROI)*. Após as regiões de interesse serem encontradas, o programa passa a classificar seus objetos. O rastreamento dos elementos é feito usando o IOU-Tracker otimizado com o *Hungarian algorithm*.

Uma outra visão de detecção e rastreamento de objetos foi proposta por Tepljakov[Tepljakov 2015], a qual consiste em utilizar o sistema de coloração HSV para realizar a detecção de objetos, e manter uma “struct”contendo os dados do objeto, para ser posteriormente comparada nas outras imagens sequenciais, tendo como resultado o rastreamento do objeto.

A presente proposta se diferencia das demais porque é executado em ambientes limitados (Computação em Borda). Utiliza-se uma estratégia diferenciada para detectar os objetos em movimento, a qual não utiliza redes neurais, apenas trabalha-se com a subtração do plano de fundo aplicando alguns filtros para otimizar os resultados.

3. Projeto UpStairs

As escadas são utilizadas para conectar diferentes pavimentos, sendo encontradas em diversos tipos de construções e fazem parte da rotina diária de muitas pessoas. O princípio básico de uma escada é permitir que as pessoas se desloquem para cima ou para baixo e, nesse movimento, o corpo humano consome energia, a qual é conhecida como força mecânica ou energia potencial do corpo [Riener et al. 2002]. De modo geral, essa energia gasta no movimento é desperdiçada, no entanto, ela poderia ser reaproveitada de alguma forma.

Partindo do pressuposto que, a Internet of Things (IoT) permite que objetos comuns possam se tornar inteligentes [Gubbi et al. 2013], o projeto UpStairs pretende empoderar uma tradicional escada de alvenaria, tipicamente encontrada em prédios e construções em geral. O objetivo geral é coletar informações de quem transita pela escada e, ao mesmo tempo, aproveitar o fluxo de pessoas para gerar energia, dando suporte a um sistema autossustentável.

Em linhas gerais, o projeto é segmentado em três áreas: (i) o mecanismo físico, o qual será responsável pela coleta, armazenagem e transmissão da energia; (ii) sistema de interação/identificação que representa a interface de comunicação com os indivíduos - Objeto de estudos deste artigo; e, (iii) sistema/aplicativo para gerenciar e consultar as informações do sistema.

4. Metodologia

Essa seção apresenta as etapas metodológicas para concepção do mecanismo de comunicação, bem como, apresenta os detalhes técnicos da implantação.

4.1. Visão Geral da Solução

O algoritmo tem por objetivo detectar quando alguma pessoa utiliza a escada, seja no movimento de subida ou descida. Nessa análise, se captura o tempo de deslocamento, quantidade de pessoas e uma estimativa de altura de cada indivíduo. Tais informações,

são utilizadas para fins de estímulo aos usuários e como efeito de contrapartida pelo uso das escadas, fornecendo-lhes informações sobre os benefícios a sua saúde, tais como: calorias gastas, velocidade e intensidade do movimento, esforço ou até mesmo a energia gerada. A Figura 1 apresenta uma visão geral da proposta.

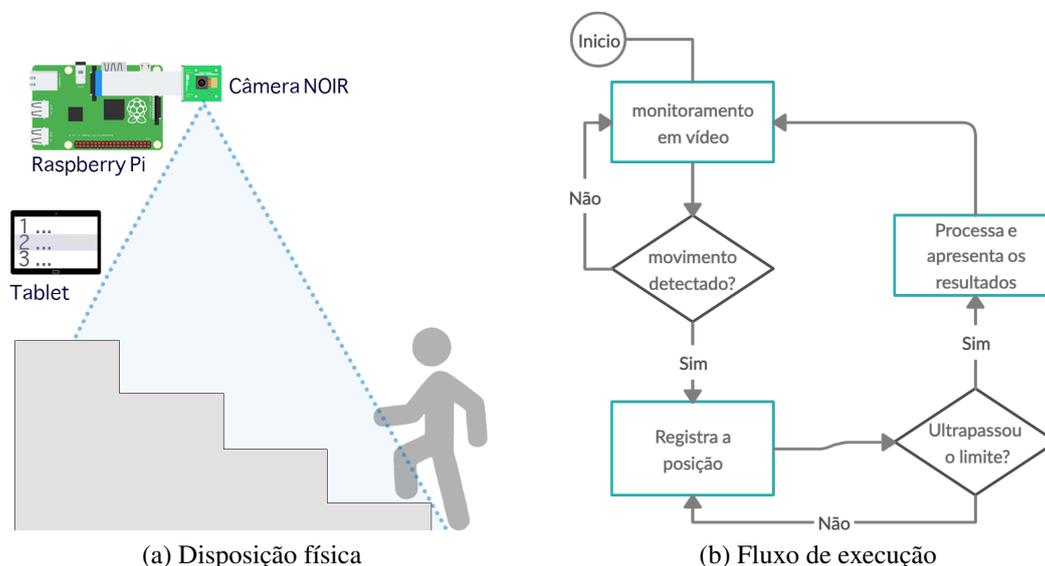


Figura 1. Visão geral da solução

Nessa proposição, no entanto, destaca-se que os dados, no que tange as informações pessoais, são valores aproximados, pois cálculos mais precisos requerem mais informações como a idade, gênero, altura e peso exatos da pessoa.

Com intuito de apresentar as informações referentes ao deslocamento dos usuários, um tablet de 7 polegadas foi utilizado como saída de vídeo. Em termos práticos, o dispositivo é posicionado no campo de visão das pessoas e fica exibindo uma página de internet. Tal página, contém os dados das últimas seis pessoas que trafegaram pela escada. O conteúdo é acessado pela rede WiFi, sendo fornecido por um computador que atua como servidor.

O servidor roda uma aplicação web desenvolvida com a linguagem Java. A aplicação é servida como um WebSocket, desta forma, a cada novo registro processado, o cliente (tablet) pode ser notificado e, por consequência, permite a atualização do conteúdo em tempo real [Pimentel and Nickerson 2012].

4.2. Ambiente

O processamento em borda é realizado na placa Raspberry Pi 3B. Ela é considerada um computador de placa única (*single board computers - SBC*), equipada com processador ARM quad-core de 1.2GHz (64 bits), 1 GB de memória RAM, WiFi integrado, Bluetooth, 4 portas USB, 40 pinos GPIO, porta Ethernet e entrada microSD [ras].

A Raspberry permite agregar novas funcionalidades através de módulo externos, dessa forma, utilizou-se um módulo de câmera NOIR (sem filtro infravermelho). Tal câmera, é capaz de capturar imagens e vídeos com resoluções a partir de 640x480 pixel. A falta do filtro de infravermelho, deixa as imagens/vídeos obtidos sob luz natural com

tons avermelhados, contudo, ela permite a captura de materiais em ambientes com pouco luz ou escuros [cam].

Quanto ao aspecto de software, adotou-se como sistema operacional o Raspbian (kernel 4.19), para processamento de imagens a biblioteca OpenCV (4.0.0) e, por fim, a linguagem Python para implementação dos algoritmos.

O ambiente de teste foi o Campus do Instituto Federal Farroupilha, São Borja. A Instituição atende alunos do ensino médio a pós-graduação, funcionando diuturnamente. O dispositivo foi posicionado na escada de acesso ao auditório do campus, localizado no quarto andar. Trata-se de um ambiente bem iluminado e com fluxo de pessoas recorrente.

4.3. Processamento em Borda

Um dos maiores desafios da visão computacional está relacionado ao reconhecimento de imagens, acentuando-se quando aplicado na borda através de SBCs. Tão complexo quanto amplo são as variações de aplicações para a tecnologia, dentre as quais, se destaca o reconhecimento e rastreo de objetos [Minichino and Howse 2020].

Embora aparente ser uma tarefa simples, acompanhar o movimento de um objeto ou pessoa, se revela, em sua natureza sutil, um sofisticado e abstruso mecanismo. Tipicamente, compreende duas fases bem definidas: identificação do objeto e o efetivo rastreo.

4.3.1. Identificação de Objetos

O reconhecimento de um objeto em uma imagem ou em um fragmento de vídeo (*frame*) é uma tarefa cara, isto é, exige bastante processamento computacional, logo, tende a ser uma operação lenta. Muitas são as alternativas para realizar tal ação, como por exemplo, os algoritmos *Haar cascades*, *HOG*, *Linear SVM*, além dos sistemas baseado em *deep-learning* tais como: SSD MobileNet Light, R-CNNs, YOLO, Tensorflow entre outros [Minichino and Howse 2020]

Realizaram-se testes com duas abordagens, SSD MobileNet Light e YOLOv3-tiny com modelos pré-treinados. Em ambos os casos, a taxa de atualização/processamento (quadros por segundo) foi baixa, em torno de 0.8 FPS (*Frames Per Second*), tal desempenho inviabilizaria a dinâmica da proposta de rastreo, bem como a execução dos cálculos calóricos.

Neste sentido, se optou por não caracterizar o objeto em movimento, ou seja, dado o contexto da aplicação proposta, compreende-se que seria um processamento desnecessário, pois é sabido que somente pessoas trafegam pelo ambiente avaliado. Assim sendo, o foco se deu no processo de rastreamento.

4.3.2. Rastreamento de Objetos

O rastreamento do indivíduo é necessário para saber a direção, velocidade e intensidade do movimento. Em termos práticos, define-se uma área de monitoramento e, se estabelece linhas de referência delimitando as fronteiras da região monitorada. A partir disso, se verifica se as pessoas estão saindo ou entrando dessa região.

O registro do deslocamento é dado quando o usuário cruza uma das linhas de referência. Para detectar tal ação, é necessário obter uma cena (*frame*) base, isto é, uma imagem do cenário sem movimento, a medida em que o ambiente é monitorado cenas são comparadas e, ao detectar algo diferente do cenário base, entende-se que existe movimento, portanto inicia-se uma sequência de ações para extrair os dados pertinentes.

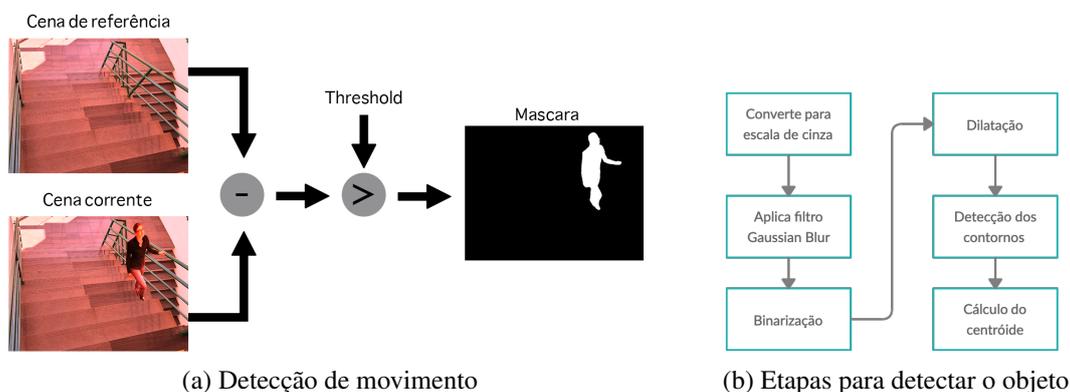


Figura 2. Estratégia para detecção de objetos

A Figura 2 (a), apresenta uma visão geral do processo de destaque dos objetos por meio da técnica de subtração de fundo (*background*). Nessa abordagem, os elementos que não são relevantes na imagem são removidos, deixando em evidência apenas os itens de interesse. A Figura 2 (b) detalha as fases para destacar o objeto.

Para otimizar os resultados do algoritmo, as imagens são convertidas em tons de cinza, facilitando as operações pixel-a-pixel. Na sequência, conforme ilustrado na Figura 2 (b), o filtro Gaussian Blur é aplicado. Esse filtro tem por objetivo deixar as figuras mais uniformes, ou seja, suaviza-se os contornos dos objetos, desta forma, a subtração do plano de fundo se torna mais eficiente, o resultado é a porção que variou entre as cenas.

Por conseguinte, realiza-se a binarização do resultado, isso garante que exista apenas duas tonalidades de cores, ou melhor, preto e branco (zero e um), por consequência, é possível processar os dados de forma mais eficiente. Entretanto, por questões de ruído ou iluminação, pode acontecer do objeto possuir falhas (buracos). Tais imperfeições, podem prejudicar a avaliação das imagens ou ainda gerar algum falso positivo. Neste sentido, para tornar o item mais consistente um algoritmo de dilatação é aplicado.

Por fim, são identificadas as coordenadas ou medidas que, em formato de retângulo, circulam o objeto. A partir dessa área, calcula-se o centro da imagem. O centro é conhecido como centróide, ele é utilizado como referência na coleta da posição do objeto na cena.

4.3.3. Cálculos Calóricos

Para se calcular o gasto calórico é necessário alguns dados como o peso do indivíduo, as dimensões da escada e o tempo de deslocamento. Entretanto, nesta proposta, utiliza-se valores aproximados para se obter o consumo calórico, pois estimativas mais precisas exigiriam um poder computacional significativo.

Por outro lado, existem alternativas que permitem calcular o custo calórico sem algumas variáveis, como por exemplo, sem o peso do usuário. A marcha Humana compreende um ciclo repetitivo, desta forma, os eventos (passadas) são uniformes independente do tempo [Rose and GAMBLE 1998].

Neste contexto, é possível aplicar métodos empíricos para estimar a quantidade de energia em kilocaloria gasto por um indivíduo em determinada atividade. Para tanto, considera-se um valor médio para uma pessoa manter seu estado de metabolismo basal por dia, isto é, 1600 Kcal/dia [Nelson and Cox 2018]. Posto isso, avalia-se a quantidade de energia em kilocaloria gasta em uma determinada atividade, isto é, uma estimativa média de acordo com a intensidade do exercício, que no caso em questão, se refere à subir ou descer escadas.

Em suma, utiliza-se um valor de referência como gasto calórico por hora, como por exemplo, subir escadas equivale a 300 kcal/h [Remizov 1991]. A partir disso, encontra-se o valor kilocaloria por segundo, ou seja, 0.102 kcal/s e, por fim, multiplica-se pelo tempo de deslocamento da pessoa, obtendo assim, o gasto calórico estimado, conforme ilustrado na Equação 1.

$$ec = \frac{1600}{day} + \frac{300}{hour} \Rightarrow \frac{1600}{86400} + \frac{300}{3600} \Rightarrow 0.0185185 + 0.083333 \Rightarrow 0.102 \quad (1)$$

$$erc = \sum_0^n (up + dc) \setminus n \quad (2)$$

$$pcc = time * ec * erc \quad (3)$$

Onde:

ec: custo efetivo de deslocamento.

erc: Taxa de correção.

up: Custo para subir (calorias).

dc: Custo para descer (calorias).

time: Duração do movimento em segundos.

pcc: Custo calórico proposto.

O custo calórico é diferente ao subir ou descer as escadas. Dessa forma, é necessário encontrar uma taxa de correção, pois ao descer as escadas o esforço é menor, por consequência, o consumo de calorias também é reduzido. Nesse sentido, para se obter uma taxa de correção precisa, foi realizado um cálculo com base na variação entre o consumo calórico dado ao subir e descer as escadas (Equação 2). Tais dados foram capturados e calculados manualmente, utilizando informações de um grupo de 6 pessoas, com biotipos diferentes.

Finalmente, o custo calórico é apresentado na Equação 3, na qual o tempo de deslocamento é multiplicado pelo custo da atividade, aplicando a taxa de correção.

5. Resultados

A Figura 3 reproduz os componentes físicos da solução. A face (a) evidencia a Raspberry Pi e o módulo de câmera, enquanto a face (b) demonstra os resultados dos indivíduos que

trafegaram pela escada. O lado (b) ainda apresenta um ícone que representa a direção do movimento, seta verde equivale à subida, enquanto que, seta vermelha indica descida.



(a) Raspberry Pi e câmera



(b) Saída de informações

Figura 3. Protótipos

A Tabela 1 demonstra as medições realizadas pelo algoritmo, comparando como os resultados de cálculos realizados manualmente, os quais consideram o peso real dos indivíduos e a velocidade dos deslocamentos (metros/segundo).

Tabela 1. Cálculo de gasto calórico

Peso	Tempo Seg.	Kcal estimada (Algoritmo)	kcal real (Manual)	Variação%
62	5.453	0.555398148	0.554409224	0.18
99	5.597	0.570064815	0.596546593	4.65
59	3.214	0.327351852	0.325488736	0.57
81	3.589	0.365546296	0.373949666	2.30
65	6.125	0.623842593	0.62517159	0.21
87	3.325	0.338657407	0.349091558	3.08

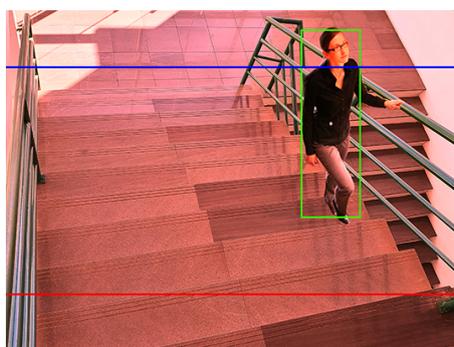


Figura 4. Pessoa em deslocamento

Destaca-se que, na Tabela 1 é possível observar que existe uma pequena variação entre o dado calculado pelo algoritmo quando comparado ao valor calculado manualmente. Na prática, o sistema considera um valor fixo de Kcal/dia (1600), enquanto o segundo cálculo, encontra o consumo real de acordo com o peso da pessoa. Como resultado se obtém uma variação de até 5%, o desvio padrão foi de 3,33%.

A Figura 4 apresenta um exemplo de avaliação do ambiente. Percebe-se a existência de linhas cruzando a imagem na horizontal. Essas linhas delimitam a região de interesse, ou seja, a área monitorada. Quando o centroide do objeto ultrapassa algumas dessas fronteiras, registra-se o horário e a posição da pessoa, bem como a direção do movimento. Quando o limite subjacente é alcançado, calcula-se o tempo e realiza-se o computo do gasto calórico.

6. Conclusão

Os sistemas embarcados são essenciais para tarefas rotineiras e repetitivas, geram novas oportunidades de negócio e impulsionam o desenvolvimento tecnológico. Com preços acessíveis, eles possuem capacidade de processamento e armazeno que dão suporte a novas aplicações, permitem avaliar contextos e agir de forma transparente, rompendo de forma singela, a linha tênue entre o físico e o digital.

As transformações trazidas por estes mecanismos possibilitam que processamento computacional seja feita na borda – e junto ao ambiente no qual estão inseridos – especialmente quando se considera processamento de imagens. Por meio de técnicas sofisticadas e elaboradas, a visão computacional se estabelece como um ascendente e promissor campo de pesquisa.

Combinando sistemas embarcados e visão computacional, se construiu um sistema que monitora e coleta dados sobre o tráfego de pessoas. Sua aplicação não impacta na rotina dos usuários, contudo, de forma sutil são ofertados dados quanto ao seu uso, cabendo ao indivíduo decidir pelo seu consumo.

Esse tipo de solução, quando aplicado em locais com alto fluxo de pessoas, podem gerar dados que podem ser utilizadas para controle de movimentações de pessoas, identificação dos horários de maior circulação, dados estes essenciais para a tomada de decisão quanto à segurança, manutenção e aos investimentos.

Como limitações, aponta-se o cálculo aproximado das informações pessoais. A busca por dados mais precisos exigiria um elevado nível de processamento, inviabilizando sua execução em borda. Na contramão, a não identificação do indivíduo possibilita resultados rápidos e atende ao requisito de tempo real da aplicação.

Os dados referentes ao esforço gasto para se deslocar e a potência dissipada no movimento, não puderam ser calculados pois, não foi possível coletar a altura e peso da pessoa. Para se obter tais informações seria necessária uma complexa e profunda análise sobre cada cena, considerando diferentes aspectos da anatomia humana, condições do ambiente como luz, ruído e angulação. Nessa perspectiva, se revela incompatível com o hardware proposto.

Por fim, a proposta ora apresentada, constitui-se como uma parte inicial de um projeto de pesquisa, o qual se reúne o conceito de computação pervasiva e visão computacional. Trata-se de um sofisticado e complexo mecanismo para desbloquear o valor ocioso que reside na utilização das tradicionais escadas, colaborando com as políticas de sustentabilidade e meio ambiente.

Referências

Pi NoIR Camera v2. <https://www.raspberrypi.org/products/>

- pi-noir-camera-v2/. Acesso: 20/03/2020.
- Raspberry Pi model description. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso: 20/03/2020.
- Brisolara, L. and Matos, J. (2009). *Desafios no Projeto de Sistemas Embarcados*, pages 153–175.
- Forsyth, D. A. and Ponce, J. (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.
- Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.
- Heath, S. (2002). *Embedded systems design*. Elsevier.
- Knappmeyer, M., Kiani, S. L., Reetz, E. S., Baker, N., and Tonjes, R. (2013). Survey of context provisioning middleware. *IEEE Communications Surveys & Tutorials*, 15(3):1492–1519.
- Lee, C.-C., Wang, C.-Y., Kao, C.-W., and Fan, K.-C. (2019). Real-time embedded system for human detection and tracking. In *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, pages 147–148. The Steering Committee of The World Congress in Computer Science, Computer
- Minichino, J. and Howse, J. (2020). *Learning OpenCV 4 Computer Vision with Python 3: Get to grips with tools, techniques, and algorithms for computer vision and machine learning*. Packt Publishing Ltd.
- Nelson, D. L. and Cox, M. M. (2018). *Princípios de Bioquímica de Lehninger-7*. Artmed Editora.
- Østby, T. G. (2018). Object detection and tracking on a raspberry pi using background subtraction and convolutional neural networks. Master’s thesis, Universitetet i Sørøst-Norge.
- Pimentel, V. and Nickerson, B. G. (2012). Communicating and displaying real-time data with websocket. *IEEE Internet Computing*, 16(4):45–53.
- Remizov, A. (1991). *Física médica y biológica*. Mir.
- Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.
- Riener, R., Rabuffetti, M., and Frigo, C. (2002). Stair ascent and descent at different inclinations. *Gait & posture*, 15(1):32–44.
- Rose, J. and GAMBLE, J. G. (1998). *Marcha humana*. São Paulo: Premier.
- Tepljakov, A. (2015). *Raspberry Pi based System for Visual Object Detection and Tracking*. PhD thesis, Bachelor’s Thesis.
- Wolf, M. (2012). *Computers as components: principles of embedded computing system design*. Elsevier.