Kleber Dobrowolski

busca de pessoas em tempo real utilizando OPENCV em placa raspberry pi

Disciplina IOT018 - Metodologia Científica

Docente: Prof. Dr. Carlos Alberto Ynoguti

são paulo

# Introdução

Nos grandes centros urbanos a ocorrência de pessoas perdidas (sejam crianças longe dos pais ou idosos senis que se afastaram do acompanhante) em estabelecimentos de grande área tais como metrô, shopping centers e museus não é incomum (buscar ref). Porém, mesmo com a existência de grande quantidade de equipamentos de vigilância, tais como câmeras, isto não auxilia na busca prática destas pessoas, já que estas câmeras no máximo são utilizadas para busca visual manual por operadores humanos. Para efetuar estas buscas, normalmente são utilizadas características físicas simples, tais como: se o sujeito da busca é criança ou adulto; cor da roupa; uso de óculos ou laço no cabelo; características físicas como calvice ou cabelos ruivos.

A abordagem de utilização de hardware de baixo custo para reconhecimento facial já foi apresentada anteriormente por *Chen at al.* [1], portanto o problema de busca pessoas utilizando processo similar parece ser uma tarefa factível, dado que a complexidade do problema é menor.

Este projeto propõe a criação de sistema de reconhecimento de pessoas utilizando abordagem de IoT (*Internet of Things*): hardware de baixo custo, comunicação remota em tempo real, aproveitamento de processamento em nuvem. O sistema será composto por uma placa Raspberry Pi 3, acoplado à sua sua câmera modular (sensor OV5647, de 5 MP de resolução), utilizando comunicação por rede WiFi e protocolo MQTT para controle a atualização dinâmica do algoritmo de busca (modelo). O processamento de imagens será realizado utilizando a biblioteca de domínio público *OpenCV* (*Open Source Computer Vision Library*).

Convém notar que o foco do projeto é a integração do harware com conceitos de IoT, e não a criação dos modelos de reconhecimento de imagem em si. Porém, será proposto que a criação destes modelos, frequentemente através de técnicas de *machine learning*, poderá ser realizada utilizando o poder computacional de processamento em nuvem, sendo que o resultado obtido (modelo) será injetado em tempo real nas câmeras, que passarão a buscar o novo sujeito do modelo, enviando os resultados das buscas para um controlador central.



Figura : resultado de uma busca por criança vestida de cor-de-rosa

## Detalhes de Implementação

A programação foi realizada utilizando linguagem Java, bem como biblioteca PaHo MQtt para comunicação, e biblioteca OpenCV para o processamento de imagens e controle das imagens da câmera. O hardware utilizado foi a placa RaspberryPI 3, embora a escolha da linguagem Java permita a utilização dos mais diversos hardwares, seja laptop, ou outros dispotivos IoT similares, tais como placa SnapDragon da Qualcomm. A câmera acoplada também permite flexibilidade: desde câmeras USB genéricas, até mesmo módulos especificos para hardware IoT, como a Câmera Raspberry Pi ou mesmo câmeras com visão noturna.

Para o controle em real-time dos módulos Argos, alguns comandos foram implementados. Estes comandos são recebidos através do ‘topic’ do MQtt.

* ‘exit’ e ‘restart’: desligam e reiniciam, respectivamente, o programa Java.
* ‘read’ e ‘write’: permitem ler e escrever parâmetros do arquivo de configuração (argos.properties). Exemplos de parâmetros: endereço IP do broker MQtt, iD individual da unidade Argos, nome da classe Java OpenCV utilizada na detecção. Alguns comandos demandam reinício do programa (tal como mudança do endereço IP do broker), através do comando ‘restart’, acima. O novo conteúdo é recebido dentro do ‘payload’ do protocolo MQtt.
* ‘snapshot’: força a unidade Argos enviar uma foto atual, com upload através de protocolo MQtt, dentro do campo payload (observação: pode existir limitação de bytes, dependendo do beoker utilizado).
* ‘class’: recebe um novo arquivo OpenCV. Este arquivo será compilado dentro da placa Raspberry Pi, e carregado imediatamente, através da técnica de Java Reflexion.

A utilização do protocolo MQtt permite grande flexibilidade no envio de comandos: podemos enviar comandos para uma certa unidade Argos específica (/cam\_id37/command/restart), ou para grupos de câmeras, ou mesmo todas as câmeras acopladas ao sistema (/command/snapshot), bastando para isto escolher o tópico MQtt desejado.

Uma interface GUI rudimentar foi criada, apenas para possibilitar a visualização de fotos de maneira rápida. Um sistema profissional

## Um possível cenário de operação

O projeto é flexível o bastante para atender muitos cenários operacionais (monitoração de metrô, sistemas distribuídos em ônibus, etc). Para um exemplo mais palpável, descreveremos uma operação em um shopping center hipotético.

* Diversos módulos Argos estão distribuídos pelos departamentos do shopping, sejam acoplados a câmeras de segurança já existentes ou substituindo as mesmas. Comunicam-se por MQtt, usando a já existente rede TCP/IP do shopping. No Centro de Controle (CO), um harware mais poderoso, para o controle centralizado dos módulos e, opcionalmente, com ferramentas mais poderosas de Machini Learning, para criar novos perfis baseados em fotos, em poucos minutos.
* o CO recebe uma demanda: uma criança de aproximadamente 4 anos, vestida de vermelho, perdeu-se da mãe. Uma foto da criança é repassada ao Centro de Controle, através de um dos seguranças do shopping. CO decide qual a melhor abordagem: criar um modelo ‘haarcascade’ novo (classe java+xml), personalizado com a foto da criança, ou apenas utilizar um modelo padrão, porém ajustado (‘criança’ + ‘roupa vermelha’).
* Quase imediatamente, as câmeras Argos recebem o novo modelo por MQtt, compilam, e passam a executá-lo. As câmeras que detectarem crianças que se enquadrem nas novas características, começarão a enviar imagens periodicamente (arquivos .JPG) com a localização do setor e a hora, para o CO.
* Como, provavelmente, existirão algumas detecções falso-positivo, um operador do CO fará a inspeção visual, para confirmar se a criança foi realmente encontrada. Se sim, enviará a foto da detecção para o segurança mais próximo do local, que poderá abordar a criança e trazê-la ao CO (ou acompanha-la diretamente ao encontro da mãe).

Desenvolvimentos futuros

[1] CHEN, Yong-Ping at al. **Low-Cost Face Recognition System Based on Extended Local Binary Pattern**. Em International Conference on Automatic Control Conference, p. 13–18, novembro. 2016.

[2] NGUYEN. Huu-Quoc at al. **Low Cost Real-Time System Monitoring Using Raspberry Pi**. Em 2015 Seventh International Conference on Ubiquitous and Future Networks, 7-10 July 2015