



CRICTE 2017

XXVIII Congresso Regional de Iniciação Científica e Tecnológica em Engenharia



UTILIZAÇÃO DE VISÃO COMPUTACIONAL E IoT PARA O CONTROLE DE UM PLAYER WEB

Pablo José Pavan

Acadêmico do curso de Ciência da Computação UNIJUI

pablo.pavan@unijui.edu.br

Éder Paulo Pereira

Egresso do curso de Ciência da Computação UNIJUI

unijic@gmail.com

Jardel Renan Stephanini

Egresso do curso de Ciência da Computação UNIJUI

jardel.stephanini@unijui.edu.br

Edson Luiz Padoin

Professor do curso de Ciência da Computação UNIJUI

padoin@unijui.edu.br

Resumo. *A visão computacional busca emular a visão humana. Desta forma, imagens podem ser utilizadas para gerar fluxos de execuções em sistemas computacionais. Neste trabalho foi utilizada visão computacional para o controle de um sistema de player web. Para tanto, adotou-se os requisitos de IoT, como o protocolo MQTT, uma aplicação Python que captura a imagem e publica no broker e outra aplicação, em JavaScript, que assina o broker e controla o player.*

Palavras-chave: *SimpleCV. Reconhecimento de Facial. IoT*

1. INTRODUÇÃO

Um dos principais sentidos dos seres vivos é a visão. Na computação, a área de computação e visão computacional busca emular tal sentido nos dispositivos. Para isso utiliza de entrada imagens que, ao serem processadas, tem-se de saída uma interpretação de uma forma completa ou parcial das mesmas (Marengoni *et al.* [1]).

Existem diversas bibliotecas que auxiliam no processamento de imagens. Dentro delas destacam-se a biblioteca SimpleCV. Segundo Demaagd *et al.* [2] esta biblioteca simplifica o uso de visão computacional, ela utiliza a linguagem de programação Python, onde agrupa diversas bibliotecas de processamento de imagens para implementar de maneira simplificada uma solução utilizando visão computacional.

O objetivo deste trabalho é utilizar a visão computacional para encontrar padrões faciais, onde através das saídas seja possível controlar um player web para executar um vídeo de forma onde não exista nenhuma intervenção humana com o programa, no caso, somente a partir do reconhecimento de uma pessoa em frente a uma webcam.

O restante do trabalho está assim organizado. A Seção 2 apresenta os conceitos básicos das tecnologias utilizadas. A Seção 3 apresenta os trabalhos relacionados. A Seção 4 descreve a metodologia utilizada. Resultados são discutidos na Seção 5 seguidos das considerações finais.



2. TECNOLOGIAS

Um sistema de visão computacional é utilizado para analisar dados de uma fonte de imagem - normalmente estas imagens são coletadas através do uso de câmeras - são extraídos dados sobre essas imagens, realizando tarefas específicas com os resultados. Para realizar a extração de dados é utilizados técnicas de processamento de imagem.

Segundo Gonzalez *et al.* [3], conforme citado por Ref. [1] o espectro que vai do processamento de imagens até a visão computacional pode ser dividido em três níveis: baixo-nível, nível-médio e alto-nível.

Os processos de baixo-nível envolvem operações primitivas, tais como a redução de ruído ou melhoria no contraste de uma imagem. Os processos de nível-médio são operações do tipo segmentação (particionamento da imagem em regiões) ou classificação (reconhecimento dos objetos na imagem). Os processos de alto nível estão relacionados com as tarefas de cognição associadas com a visão humana.

Internet of things (IoT) atualmente é uma estrutura sólida para o desenvolvimento de aplicativos conectados utilizando computação em nuvem, para isso é disponibilizado pela comunidade diversos protocolos para realizar as comunicações. Estes protocolos buscam trazer maneiras leves e simplificadas para implementar os aplicativos em diversas plataformas com baixo processamento.

Um destes protocolos é o *Message Queue Telemetry Transport* (MQTT¹) que utiliza um sistema de *Publish-subscribe* (Hunkeler *et al.* [8]), onde as trocas de comunicação acontecem entre os nodes que

publicam conteúdos em um *broker* que disponibiliza para os nodes assinantes.

3. TRABALHOS RELACIONADOS

No trabalho de Menezes *et al.* [4], é utilizado a biblioteca SimpleCV e a MPSoC Raspberry Pi para detectar objetos em movimento na área de captura de uma webcam, transmitindo as imagens através MPJG *Streamer*, com o objetivo de criar sistemas de segurança. Outro trabalho que utiliza SimpleCV e OpenCV é de Tripathy *et al.* [5], onde utilizando Raspberry Pi BCM2835, processa imagens em tempo real para encontrar padrões de face humana e de postura que a cabeça se encontra.

Outras pesquisas utilizam o MQTT para realizar comunicações, principalmente na área de *IoT*. O trabalho de Barata *et al.* [6] é utilizado o protocolo MQTT para receber dados de um aplicativo Android que envia informação sobre Oximetria de pulso dos pacientes com problema cardiovasculares para uma *broker*, isso permite que médico possa acessar os dados sobre o paciente sem estar fisicamente presente.

No trabalho de Upadhyay *et al.* [7] foi utilizado a Raspberry Pi B+ e MQTT para criar um sistema de segurança para casas. Utilizando de diversos sensores conectados na Raspberry Pi, é realizado o controle de segurança, os dados dos sensores são enviados para a nuvem através do MQTT e acessados em uma página web.

4. METODOLOGIA

Para desenvolver o sistema de visão computacional, foi utilizado a linguagem de programação Python juntamente com a biblioteca SimpleCV, este sistema tem como objetivo encontrar padrões nas imagens de

¹ <http://mqtt.org/>



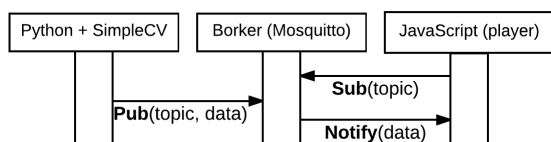
entrada, buscando o reconhecimento de uma face humana, olhando para frente.

Para encontrar estes padrões, é utilizado como entrada de imagens uma webcam em tempo real. O primeiro passo é realizar um filtro onde busca-se a parte superior acima do peito do corpo humano, em seguida realiza-se um corte na imagem, onde está posicionado esta parte. Depois deste processo, dentro desta imagem cortada, é realizado uma nova busca, esta busca filtra a imagem em rostos encontrados, realizando um novo corte.

Neste novo corte, realiza-se novamente um filtro por busca de olhos, se encontrado tem-se a garantia que esta pessoa está olhando para frente, assim a saída final do processo é verdadeira, dando entrada em um novo fluxo de instruções, caso de negativo, inicia-se o processo desde o primeiro passo.

Levando em consideração uma saída verdadeira, o novo fluxo de instrução realiza a comunicação com o player web. Dado que o player web é desenvolvido utilizando HTML e JavaScript, é necessário realizar uma integração entre duas linguagens de programação distintas: Python e JavaScript.

Para realizar esta integração é utilizado o MQTT onde o Python realiza as publicações e o JavaScript implementa o assinante. Para implementar o protocolo é utilizado o *broker* Mosquitto², que é um software livre de fácil configuração. A Figura 1 apresenta o fluxo de troca de mensagens através do tempo do modelo MQTT.



² <http://mosquitto.org>

Figura 1. Fluxo de troca de mensagens do MQTT através do tempo

Primeiramente o JavaScript assina o tópico “topic” no *broker*, em sequência o Python pública no tópico “topic” e envia junto os dados para o *broker*, por fim o *broker* envia os dados para o JavaScript.

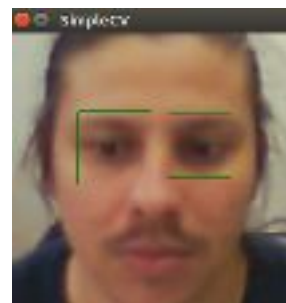
Enfim, caso a saída do Python seja verdadeira, tem-se uma pessoa olhando para a frente - olhando na direção da webcam - o player web inicia a apresentação de um vídeo.

5. RESULTADOS

Um dos resultados alcançados são referentes a integração utilizando MQTT, neste sentido a utilização do mesmo apresenta tempo de resposta entre as aplicações < 100 milissegundos, garantindo um sistema em tempo real.

Outro resultado refere-se a utilização do SimpleCV, este garante um desenvolvimento bastante simples, porém como deixa abstrato para o programador, o uso de diferentes bibliotecas sobrecarregam as interações, isso resulta em aumento da necessidade de capacidade computacional, onde pode ser notado na captura de imagem da webcam que somente pode ocorrer a cada 1 segundo.

A Figura 2 demonstra o resultados dos filtros utilizados, mostrando o rosto da pessoa e identificando seus olhos.





CRICTE 2017

XXVIII Congresso Regional de Iniciação Científica e Tecnológica em Engenharia



Figura 2. Janela final da implementação do SimpleCV reconhecendo os olhos

O sistema será utilizado durante a EXPOIJUÍ FENADI 2017, na casa da UNIJUÍ, onde será replicado em 6 equipamentos, para contar a história dos 60 anos da UNIJUÍ. O objetivo é que no momento que uma pessoa parar em frente de uma das webcams um vídeo iniciará no projetor corresponde a direção que a pessoa está olhando.

6. CONSIDERAÇÕES FINAIS

A utilização de visão computacional, pode ser utilizada para diversos casos, neste trabalho buscou-se realizar um controle de um player web, utilizando a mesma.

Ao utilizar visão computacional, é necessário realizar diversos filtros e tratamentos de imagem. Com isso, percebe-se que este processo demanda uma grande capacidade computacional.

Como trabalho futuros pretende-se implementar melhorias no fluxo de filtros, deixando o processo de forma a não necessitar de uma grande capacidade computacional.

Agradecimentos

Trabalho realizado no Curso de Ciência da Computação da UNIJUÍ, financiado parcialmente pelo PIBEX/UNIJUÍ.

7. REFERÊNCIAS

[1] Marengoni M, Stringhini S. Tutorial: Introdução à visão computacional usando opencv. Revista de Informática Teórica e Aplicada. 2009;16(1):125-60.

[2] Demaagd K, Oliver A, Oostendorp N, Scott K. Practical computer vision with

SimpleCV: The simple way to make technology see. " O'Reilly Media, Inc."; 2012 Jul 26.

[3] Gonzalez RC, Eddins SL, Woods RE. Digital Image Publishing Using MATLAB. Prentice Hall; 2004.

[4] Menezes V, Patchava V, Gupta MS. Surveillance and monitoring system using Raspberry Pi and SimpleCV. InGreen Computing and Internet of Things (ICGCIoT), 2015 International Conference on 2015 Oct 8 (pp. 1276-1278). IEEE.

[5] Tripathy R, Daschoudhury R. Real-time Face Detection and Tracking Using Haar Classifier on SoC. International Journal of Electronics and Computer Science Engineering. 2014;3:175-84.

[6] Barata D, Louzada G, Carreiro A, Damasceno A. System of acquisition, transmission, storage and visualization of pulse oximeter and ECG data using android and MQTT. Procedia Technology. 2013 Jan 1;9:1265-72.

[7] Upadhyay Y, Borole A, Dileepan D. MQTT based secured home automation system. InColossal Data Analysis and Networking (CDAN), Symposium on 2016 Mar 18 (pp. 1-4). IEEE.

[8] Hunkeler U, Truong HL, Stanford-Clark A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. InCommunication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on 2008 Jan 6 (pp. 791-798). IEEE.