

Detecção e classificação de objetos utilizando computador de baixo custo Raspberry PI

Kaique Rijkaard de Sousa Oliveira, Ricardo de Sousa Farias, Hoffmann da Silva Queifer, Jannayna Domingues Barros Filgueira e Geam Carlos de Araújo Filgueira

Resumo—Este artigo apresenta um estudo sobre o processo de detecção e classificação de objetos. Apresenta ainda um estudo sobre a plataforma Raspberry PI utilizada para processamento e captura de imagens. Para isto utilizou-se bibliotecas do OpenCV e o método de classificação Viola-Jones, mais conhecido como Haar Cascade. Inicialmente, os testes foram realizados com dois objetos, em seguida foi realizada a análise dos resultados obtidos.

Palavras-Chave—Processamento de imagens, Classificação de objetos, Raspberry PI, OpenCV.

Abstract—This paper presents a study about the process of detecting and classifying objects. It also presents a study on the Raspberry PI platform used for processing and capturing images. For this we used OpenCV libraries and the Viola-Jones classification method, better known as Haar Cascade. Initially, the tests were performed with two objects, after which the results obtained were analyzed.

Keywords—Image processing, Object classification, Raspberry PI, OpenCV.

I. INTRODUÇÃO

Segundo Gonzales [1], uma das primeiras aplicações das imagens digitais ocorreu na indústria dos jornais, onde as imagens eram enviadas por cabos submarinos entre as cidades de Londres e Nova York. O grande avanço no processamento digital de imagens ocorreu com o advento dos primeiros computadores digitais de grande porte, desde então a área de processamento digital de imagens vem crescendo expressivamente e suas aplicações se estendem para diversos ramos da atividade humana.

Define-se processamento digital de imagem como uma técnica para capturar, representar e transformar imagens com auxílio de sistemas computacionais [1]. A formação de uma imagem digital dar-se pela composição de um número de elementos finitos, com tamanho, localização e valores específicos. Esses elementos são chamados de *pixel*, sendo este o termo utilizado para representar uma imagem digital ou binária.

No campo da visão computacional, as técnicas e métodos de processamento de imagem são amplamente utilizados, como na análise de imagens, reconstrução de imagens, monitoramento de vídeos, reconhecimento de objetos, estimativa de movimentos, restauração de imagens e etc. A visão computacional é uma importante ferramenta para o reconhecimento de ambientes, pois as informações resultantes do processamento das imagens podem ser utilizadas para o

reconhecimento e a detecção de padrões [2]. No processo de captura e análise de imagens algumas informações são extraídas a fim de definir qual ação deverá ser tomada, baseado em padrões encontrados nas imagens capturadas.

Para realizar a análise das informações extraída das imagens, são utilizados um conjunto de algoritmos específicos para o processamento de imagens. Esses algoritmos são classificados de acordo com suas funcionalidades, como por exemplo algoritmos para filtragem, algoritmos detectores de bordas de objetos, algoritmos classificadores de texturas, etc.

Neste aspecto, este trabalho visa apresentar um estudo sobre visão computacional e processamento de imagens. Para isto realiza-se a detecção e classificação de objetos utilizando a plataforma Raspberry PI, que consiste em um computador de baixo custo. Para captura das imagens uma Câmera Raspberry PI foi utilizada. Em seguida, aplicou-se às imagens algoritmos de detecção e reconhecimento de objetos.

II. MATERIAIS E MÉTODOS

A. Raspberry PI

Em 2011, a Fundação Raspberry PI desenvolveu seu primeiro computador em placa única. O Raspberry PI consiste em um computador com o tamanho reduzido, semelhante ao tamanho de um cartão de crédito, com baixo custo e baixo consumo de energia [3]. O computador foi criado com o intuito de promover o ensino de programação para crianças e adultos. Hoje ele é utilizado de diversas formas, por exemplo, para programar, como vídeo game, mini servidor, automação residencial, robótica entre outros projetos [4].

Este dispositivo pode ser considerado um computador “completo”, podendo ser classificado como SBC (*Single-Board Computer*), pois integram em uma única placa processador, memória e interfaces de entrada e saída. O Raspberry PI se enquadra na categoria de equipamento denominada *Open-Source Hardware* (OSH), ou equipamento livre. Os OSHs podem ser definidos como dispositivos elétricos ou mecânicos cujas informações sobre o projeto são disponibilizadas para todos, de forma que o público possa fazer uso, modificar, produzir ou distribuir este material.

Existem vários modelos de Raspberry PI, alguns são apropriados para projetos mais robustos e outros para projetos mais simples. Nesta pesquisa utilizamos Raspberry PI 3 modelo B. O modelo Raspberry PI 3 B possui uma CPU ARMv8 quad-core de 64 bits com frequência de 1.2 GHz, núcleo de gráficos

3D Vídeo Core IV, 1GB de memória RAM, 4 portas USB, LAN sem fio 802.11n, Bluetooth 4.1. Associada ao computador Raspberry utilizamos uma câmera, Raspberry Câmera PI 3 V2 que possui um sensor Imx219 de 8 megapixels da Sony, que pode ser usado para capturar vídeo de alta definição, fotos estáticas e captura contínua [3].

A Figura 1 ilustra o Raspberry PI 3 B e a Raspberry Câmera utilizados nesta pesquisa.



Fig. 1. Raspberry PI 3 associado a Raspberry Câmera compondo o Setup utilizado. Fonte: autor.

B. OpenCV

Como base para as rotinas computacionais a serem usadas, foi escolhido a biblioteca OpenCV. Para instalação e uso, foi realizado o download da versão mais recente do sistema Raspbian na Raspberry PI 3 B e em seguida o mesmo foi compilado e instalado para uso [5]. Criado pela Intel, a OpenCV é uma biblioteca multiplataforma totalmente livre para uso acadêmico e comercial desenvolvido na linguagem C/C++ com suporte para outras linguagens como Python e Java. O OpenCV fornece funções bem estruturadas que recebe o classificador de características e a partir deste classificador consegue detectar objetos [6].

C. Haar Cascade

O Algoritmo escolhido para o processo de classificação dos objetos foi o Haar Cascade, também chamado de método Viola-Jones. Este método faz parte das bibliotecas do OpenCV.

Para fazer a interpretação dos objetos os sistemas de visão computacional precisam de um conhecimento prévio sobre suas formas. Isso pode ser conseguido através de duas maneiras distintas. A primeira é ter uma base já existente de formas, onde os padrões já estão previamente armazenados e a segunda trata-se do reconhecimento a partir de aprendizado [7].

Este método utiliza no processo de aprendizagem imagens positivas e imagens negativas. Classifica-se como imagens positivas aquelas que possuem o objeto a ser classificado e imagens negativas aquelas que não possuem o objeto que se deseja classificar. O algoritmo realiza o aprendizado de máquina onde os classificadores são treinados a fim de obter e armazenar características em cascata dos objetos desejados.

Este algoritmo é amplamente utilizado nos processos de detecção facial, podendo também ser aplicado a outros objetos.

O Haar Cascade extrai as características do objeto desejado a partir do treinamento de aprendizagem de máquina chamado AdaBoost. Neste processo uma grande quantidade de imagens positivas e negativas são aplicadas ao algoritmo gerando uma cascata de características do objeto desejado no formato XML. A técnica utilizada no classificador consiste em uma análise de imagens em tons de cinza, em que as diferenças de intensidade de regiões distintas são comparadas a características armazenadas em uma árvore de decisões, que é desenvolvida através de um algoritmo de treinamento e um conjunto de imagens das quais são extraídas características específicas do objeto a ser rastreado [8]. Na execução do algoritmo uma série de características são analisadas em sequência até que um padrão seja encontrado

Este classificador foi escolhido por oferecer uma grande liberdade de escolha de alvo de acordo com a aplicação, porque dispõe de técnicas de treinamento inteligente e por oferecer algoritmos rápidos e de baixo custo computacional para a detecção em tempo real do alvo. Além disso, o Haar Cascade oferece uma grande robustez ao detector de alvo, pois o mesmo pode ser treinado para detectar alvos em condições mais adversas como: baixa luminosidade e oclusão parcial do alvo [9].

III. RESULTADOS E DISCUSSÕES

As informações seguintes descrevem os processos de captura e classificação dos objetos desejados. Nesta pesquisa os objetos a serem classificados foram obtidos de imagens e vídeos. Portanto a análise dos resultados foi dividida nesses dois grupos: imagens e vídeos.

A. Plano de Teste

O plano de teste tem como objetivo organizar a sequência de passos necessários para apanhar os resultados do algoritmo de detecção de objetos e analisar manualmente o mesmo. Para isso uma análise foi feita com base nos relatórios gerado pelo programa e nas imagens/vídeos que foram processados e salvos. Utilizou-se dois grupos de objetos para serem classificados: faces e relógios.

Para a detecção das faces nas imagens foram utilizadas as bases de dados Faces 1999 disponível em [10] e a Helen Dataset disponível em [11] com imagens positivas. A [10] dispõe uma quantidade de 450 imagens, onde cada imagem possui apenas uma face frontal. A [11] dispõe uma quantidade de 2.330 imagens, cada imagem possui diversas faces em diferentes posições. Para a detecção dos relógios foi criado uma base de dados com 100 objetos, sendo que a quantidade de relógios varia por cada imagem.

Foi utilizado a base de dados de [12] para as imagens negativas que dispõe um total de 7.983 imagens. Este banco de dados foi utilizado para testar o classificador da face e do relógio.

Para a detecção de faces e relógios em vídeo foi utilizado a câmera Raspberry PI, onde os vídeos foram gravados com uma resolução de 320x280 pixel. No processo de gravação, o algoritmo realiza o armazenamento de uma imagem todas as vezes que identifica o objeto desejado. O tempo de gravação variou entre 2 a 3 minutos, de modo que foi possível obter uma grande quantidade de imagens para classificação.

B. Resultados das imagens

Atendendo aos requisitos do algoritmo utilizado, aplicou-se o plano de testes. Executamos as simulações com as duas bases de dados de imagens positivas. Os seguintes parâmetros foram configurados para obter os resultados na detecção de cada objeto:

- Faces: o tamanho mínimo de (120, 120), o fator de escala que consiste na distância dos objetos na imagem de 1.3 e o parâmetro da quantidade mínima de vizinhos de 4, que é a quantidade de retângulos vizinhos que o retângulo candidato deve ter;
- Relógio: o tamanho mínimo de (30, 30), um fator de escala de 1.05 e a quantidade mínima de vizinhos de 3.

Na Figura 2 o algoritmo classificou corretamente todas as Faces positivas, porém identificou uma Face inexistentes. Na Figura 3 o algoritmo identificou a presença de uma Face em uma imagem negativa.

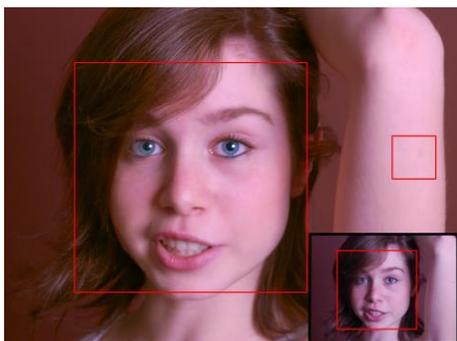


Fig. 2. Detecção de faces em imagem positiva com identificação de faces inexistentes [11].



Fig. 3. Detecção de faces em imagem negativa [12].

Na Figura 4 o algoritmo classificou corretamente todas as faces presentes na imagem positiva.

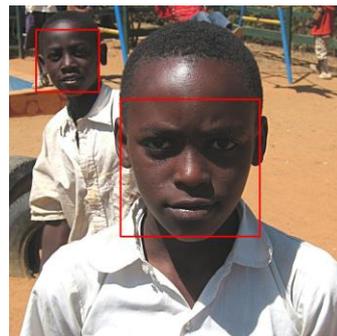


Fig. 4. Detecção exata das faces na imagem positiva [11].

Nos testes realizados com o objeto Relógio o algoritmo se mostrou satisfatório. Na Figura 5 o algoritmo classificou corretamente todos objetos presentes na imagem positiva. Porém, houve casos em que o objeto foi identificado em imagens negativas, como ilustrado na Figura 6.



Fig. 5. Detecção exata dos relógios na imagem positiva [13].



Fig. 6. Detecção de relógio na imagem negativa [12].

A análise dos resultados foi baseada em cinco aspectos: acertos em imagens positivas (AP), erros nas imagens positivas (EP), erro crítico em imagens positivas (ECP), acertos nas imagens negativas (AN) e erro crítico nas imagens negativas (ECN). Nos acertos positivos o algoritmo deve identificar apenas os objetos existentes nas imagens positivas. Nos erros positivos o algoritmo consegue detectar o objeto desejado na imagem positiva, porém também pode detectar objetos inexistentes, como foi o caso da Figura 2, e em outros casos deixar de detectar o restante dos objetos. Nos erros críticos positivos o algoritmo falha e não consegue detectar nenhum dos objetos na imagem e ainda detecta objetos que não existem. Nos

acertos negativos o algoritmo não pode identificar nenhum objeto nas imagens negativa, o que mostrou ser excepcional no classificador das faces. Nos erros críticos negativos o algoritmo falha identificando objetos inexistentes nas imagens negativas, como mostra a Figura 3.

A Tabela I apresenta os resultados obtidos.

TABELA I. RESULTADOS PERCENTUAIS DA ANÁLISE DAS IMAGENS

BANCOS DE DADOS	AP	EP	ECP	AN	ECN
FACES 1999	97,11 %	0,89%	2,00%	99,79 %	0,21%
HELEN DATASET	63,57 %	14,76 %	21,67 %		
RELÓGIOS	33,00 %	42,00 %	25,00 %	69,41 %	30,59 %

Para a base de dados [10] o algoritmo apresentou um resultado satisfatório com 97,11% de acerto positivo. Para a base de dados disponível em [11] o resultado também foi satisfatório, com 63,57% de acertos positivos. Na execução no banco de dados das imagens negativas para o classificador das faces os resultados foram satisfatórios, apresentando uma elevada taxa de acerto.

Para a base de dados dos relógios os resultados foram bem variados. O algoritmo conseguiu ter bons resultados nas imagens positivas, conseguiu detectar a maior parte dos relógios existentes nas imagens, porém, ele teve uma facilidade em detectar relógios inexistente, produzindo assim uma porcentagem maior dos erros positivos. Nas imagens negativas houve uma redução e comparada com o classificador das faces, mas o seu resultado foi satisfatório considerando que este classificador possui uma facilidade de detecção como foi citado acima.

C. Resultados dos vídeos

Para a simulação do vídeo os parâmetros originais foram mantidos.

Na Figura 7 o algoritmo classificou corretamente todas as Faces presentes no vídeo. O mesmo não ocorreu na Figura 8, onde duas Faces não foram detectadas, neste caso a distância do objeto à câmera e o movimento no momento da gravação dificultou o processo de detecção do objeto desejado.

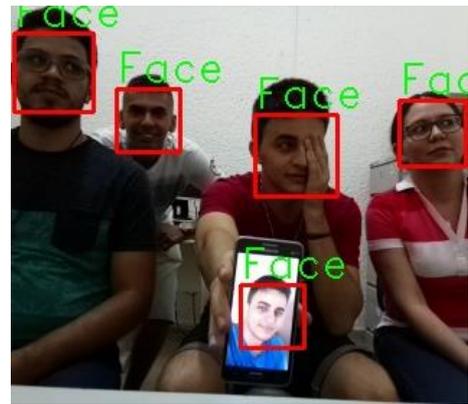


Fig. 7. Detecção de faces em vídeo. Fonte: autor.

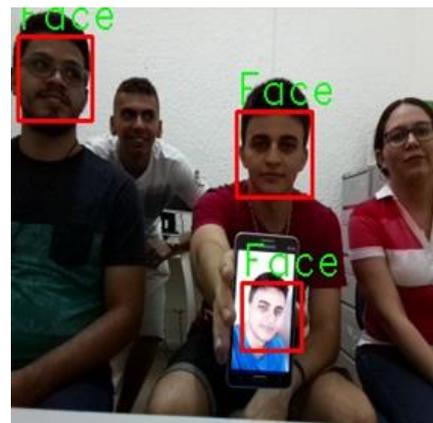


Fig. 8. Erros de detecção de faces em vídeo. Fonte: autor.

É importante ressaltar que na gravação dos vídeos alguns parâmetros influenciam na classificação correta do objeto, como por exemplo, o objeto deve estar numa distância mínima da câmera.

O segundo teste com vídeo foi realizado com o objeto Relógio. Para este objeto o algoritmo teve um bom desempenho. A Figura 9 ilustra a correta detecção do objeto desejado. Em alguns vídeos o algoritmo não identificou o objeto mesmo ele estando presente na imagem capturada, como ilustrado na Figura 10.

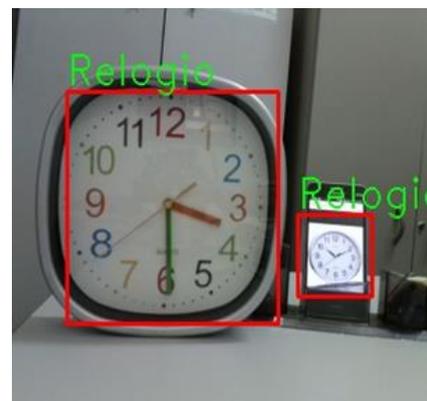


Fig. 9. Detecção de Relógio em vídeo. Fonte: autor.



Fig. 10. Erros de detecção de Relógio em vídeo. Fonte: autor.

Na análise dos resultados das imagens obtidas a partir dos vídeos, identificamos que o algoritmo teve um desempenho equilibrado para ambos os objetos.

A Tabela II apresenta os resultados obtidos.

TABELA II. RESULTADOS PERCENTUAIS DA ANÁLISE DAS IMAGENS OBTIDAS A PARTIR DOS VÍDEOS

IMAGENS	IMAGENS COM ACERTOS	IMAGENS COM ERROS
FACES	42,86%	57,14%
RELÓGIO	44,25%	55,75%

IV. CONCLUSÕES

Este trabalho apresentou um estudo sobre visão computacional e processamento de imagens. Para isto realizou-se a detecção e classificação de objetos utilizando a plataforma Raspberry PI e uma Câmera Raspberry. Para classificação das imagens o algoritmo Haar Cascade, também chamado de método Viola-Jones, foi aplicado. Este algoritmo se mostrou eficaz às aplicações, porém essa pesquisa está em andamento. Ao fim deste processo espera-se desenvolver um dispositivo robótico móvel capaz de detectar e classificar objetos fixos ou em movimento. Para isto, outros algoritmos serão testados a fim de comprovar e validar a eficácia do método aplicado.

AGRADECIMENTOS

A Universidade Estadual da Paraíba, Campus VII e ao PIBIC.

REFERÊNCIAS

- [1] GONZALEZ, R. C.; WOODS, R. E. Processamento Digital de Imagens. [S.l.: s.n.], 2009. 623 p.
- [2] GERONIMO, D. et al. Trac sign recognition for computer vision project-based learning. Education, IEEE Transactions on, v. 56, n. 3, p. 364{371, 2013. ISSN 0018-9359.
- [3] RASPBERRY PI PRODUCTS, 2017. Disponível em <<https://www.raspberrypi.org/products/>>. Acessado em 11 de Abril de 2017.
- [4] MONK, Simon. Programando o Raspberry PI: Primeiros Passos com Python. 1ª Edição. São Paulo: Editora Novatec, 2013.
- [5] Raspbian. Raspbian [Online]. Disponível em: <https://www.raspbian.org/>.
- [6] OpenCV. OpenCV [Online]. Disponível em: <http://opencv.org/>.
- [7] ROSA, A. L. B. Sistema de tracking de objetos a partir de várias câmeras, 2010.
- [8] PAVAN, Adilson Roberto; MODESTO, Fábio Alexandre Caravieri. Reconhecimento de Gestos com Segmentação de Imagens Dinâmicas Aplicadas a Libras.
- [9] PEREIRA, Rafael Cardoso. Técnicas de rastreamento e perseguição de alvo utilizando o algoritmo haar cascade aplicada a robôs terrestres com restrições de movimento, 2017.
- [10] Caltech. Caltech [online]. Disponível em <http://www.vision.caltech.edu/html-files/archive.html>. Acesso em: 05 de Julho de 2018.
- [11] Helen Dataset. Helen Dataset [online]. Disponível em: <http://www.ifp.illinois.edu/~vuongle2/helen/>. Acesso em: 05 de Julho de 2018.
- [12] L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004.
- [13] PXHERE, 2018. Disponível em <<https://pxhere.com>>. Acesso em :25 de Janeiro de 2018.