

Monitoramento do fluxo de pessoas por meio de Técnicas de Reconhecimento de Padrões

Monitoring the flow of people using Pattern Recognition Techniques

RESUMO

Este trabalho relata o desenvolvimento de um projeto com objetivo de realizar o monitoramento do fluxo de pessoas em um sistema de computação de borda, utilizando o minicomputador *Raspberry Pi* e o *Stick* de computação Neural da Intel. Com isso foi realizada uma pesquisa para decidir quais seriam os melhores métodos e ferramentas a serem aplicadas, com isso foi desenvolvido um sistema implementado na linguagem de programação *python* com a biblioteca para visão computacional *OpenCV*, utilizando e comparando métodos de classificação SVM e KNN para o reconhecimento de rostos. Os primeiros resultados foram promissores, mas quando há o aumento de faces a serem reconhecidas treinadas, a acurácia diminui consideravelmente, o que mostra a necessidade de alguns aprimoramentos no sistema.

PALAVRAS-CHAVE: Reconhecimento Facial, *Raspberry*, *Opencv*.

ABSTRACT

This paper reports the development of a project with the objective of monitoring the flow of people in an edge computing system, using the *Raspberry Pi* minicomputer and the Intel Neural computing stick. With this, a research was carried out to decide which would be the best methods and tools to be applied, with this a system implemented in the python programming language with the *OpenCV* computer vision library was developed, using and comparing SVM and KNN classification methods for the face recognition. The first results were promising, but when there is an increase in faces to be recognized trained, the accuracy decreases considerably, which shows the need for some improvements in the system.

KEYWORDS: Face Recognition, *Raspberry*, *Opencv*.

Gustavo de Almeida Duarte

gduarte@alunos.utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Cornélio Procopio, PR, Brasil

Rodrigo Henrique Cunha Palácios

rodrigopalacios@utfpr.edu.br

Universidade Tecnológica Federal do Paraná, Cornélio Procopio, PR, Brasil

Recebido: 04 ago. 2020.

Aprovado: 01 out. 2020.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

A inteligência artificial (IA) (Russell e Norvig, 2013) é uma área que está crescendo bastante e o avanço do desempenho computacional tem favorecido isso, um de seus focos é sobre como fazer uma máquina aprender (*machine learning*). A aprendizagem de máquina é dividida em 4 abordagens, que são: Aprendizado supervisionado, onde todas as amostras são rotuladas e há um algoritmo que faz o mapeamento da entrada para a saída; Aprendizado não supervisionado, o qual é feito o agrupamento de amostragem similares; Aprendizado Semi supervisionado, apenas uma parte das amostragens são rotuladas e a partir dessas são feitas uma propagação de rótulos para as outras amostras não rotuladas; e Aprendizado Ativo, forma de aprendizagem que são feitas algumas poucas interações com o especialista que irá rotular poucas amostras, essas que iram contribuir para uma curva maior de aprendizagem.

O processamento de imagens (PDI) (BEZDEK; KELLER; KRISNAPURAM e PAL N, 2005) é uma área de estudo de algoritmos ou modelos para a manipulação de imagens realizadas por um computador. É formado por um conjunto de técnicas que possui uma imagem como entrada, é realizada uma série de processamentos sobre essa imagem, tendo como resultado uma imagem de saída com algumas propriedades alteradas em relação a imagem original, que podem ser utilizadas para facilitar análises de acordo o objetivo pretendido.

Visão computacional (PEREIRA, 2016) é inspirada na visão biológica e, em termos não técnicos, seria a forma como um computador vê. Ela pode ser entendida como a interseção da área de IA com a de Processamento de imagens, ou seja, a partir de imagens processadas e aquisição dos dados, é feita interpretação do que foi obtido e ao final tem-se o retorno do resultado com o objetivo de classificar, efetuar a detecção e/ou reconhecimento em imagens e vídeos. Ela possui aplicação em várias outras áreas, como segurança, saúde, monitoramento, automatização de atividades na indústria e na agricultura. Apesar da visão computacional estar em alta, há várias técnicas e métodos sendo desenvolvidos e estudos para a obtenção de uma melhor desempenho e escalabilidade da aplicação desenvolvida, ou seja, não há uma definição de qual conjunto de técnicas, de métodos e tecnologias são mais adequados para cenários específicos. A partir desse gap, foi realizado um estudo na literatura para a obtenção de quais técnicas tinham um melhor custo-benefício em termos de desempenho, custo de desenvolvimento e se é ou não de fácil mobilidade para realizar a tarefa de monitoramento de fluxo de pessoas.

MATERIAIS E MÉTODOS

O projeto foi desenvolvido com o objetivo de ser de fácil mobilidade de instalação em qualquer ambiente, portanto foi desenvolvido para execução em um *Raspberry Pi*, um computador pequeno e de baixo custo, sendo utilizado o modelo 3B+. Para um melhor desempenho do sistema também foi utilizado o *stick* de computação Intel, uma VPU - *Vision Processing Unit* (Unidade de Processamento Visual), unidade que possui um altíssimo desempenho em aplicação na área de Visão Computacional.

Por possuir objetivo similar ao ser atingindo por esse trabalho e por possuir resultados promissores, o projeto desenvolvido por (ROSEBROCK, 2020) foi utilizado como base.

Para a parte de programação foi utilizado a linguagem de programação *python* e o *OpenCV*, API - *Application Programming Interface* (Interface de Programação de Aplicações) desenvolvida pela Intel para o desenvolvimento de aplicações relacionadas a visão computacional.

Uma das primeiras etapas para realização desse projeto é a construção do *dataset* com as imagens requeridas e em uma aplicação de reconhecimento facial essa etapa é de suma importância, pois ela que representa os dados que serão oferecidos para a rede neural a ser treinada, assim influenciando no resultado do treinamento, algo que será melhor abordado no tópico de resultados e discussões.

Para a construção do *dataset*, o primeiro passo é a aquisição das imagens dos rostos, que são obtidos de duas formas, utilizando a API de busca de imagens do Bing, para o caso de testes de imagens com figuras públicas ou tirando foto pela *webcam*. Após essa etapa é realizada o processo de alinhamento das faces que são salvos em outra pasta. Esse processo de alinhamento possui os seguintes passos:

- a) Primeiro é realizada a detecção da face;
- b) Assim que detectada, é realizada o recorte da imagem apenas na região de interesse, que é o rosto;
- c) Posteriormente é realizada a detecção das marcas do rosto (*face landmarks*).
- d) Por último é utilizado a posição dos olhos para realizar a rotação da imagem e centralizar a face.

Esse processamento foi utilizado para melhorar a acurácia final do treinamento.

Após a montagem do *dataset*, com as imagens já devidamente tratadas, é realizada o processo de extração de *embeddings*, vetores de baixa dimensionalidade que possuem as informações mais relevantes a serem comparadas no processo de classificação de cada face.

Posteriormente é realizada a fase de treinamento, onde os dados são tratados e passam pelo processo de classificação utilizando os métodos SVM (*Support Vector Machine*) ou KNN (*K-Nearest Neighbors*), os dois são métodos de classificação a partir do dados de entrada recebidos, entretanto utilizam técnicas diferentes para a realização da classificação em si e os dois estão disponíveis na biblioteca do *scikit learn* (PEDREGOSA, 2011). Nos dois métodos a escolha apropriada dos parâmetros é muito importante para o melhor desempenho do classificador e por isso foi utilizada a função *GridSearchSVC()*, função que encontra os melhores parâmetros a serem utilizados no modelo a ser treinado.

Ao final do treinamento são gerados dois arquivos que serão utilizados na parte de reconhecimento, com isso, toda vez que o script de reconhecimento for utilizado, não é necessário passar por um novo processo de treinamento. O processo de treinamento só será realizado quando o *dataset* de imagens for utilizado.

RESULTADOS E DISCUSSÕES

Para a realização dos testes foi pensado vários cenários diferentes, para poder ser analisado a eficácia do projeto em si, os cenários foram:

- Cenário 01: Treinar apenas o reconhecimento dos rostos que aparecem no vídeo, no caso, 4 pessoas.
- Cenário 02: Reunir imagem de mais 50 pessoas e analisar os resultados.
- Cenário 03: Reunir imagem de mais 500 pessoas e analisar os resultados

No Cenário 02 e 03, o qual são acrescentadas imagens de 50 e 500 pessoas, respectivamente, é utilizado o *dataset* LFW (*Labeled Faces in the Wild*), disponibilizado por (ANAND, 2019). As faces rotuladas foram escolhidas de forma aleatória e cada face rotulada possui um número diferente de imagens.

Para realizar os testes, foi obtido um vídeo de uma entrevista do ator norte americano Hugh Jackman, com três influenciadores digitais brasileiros (Alexandre Ottoni ou Jovem Nerd, Atila Iamarino e Cauê Moura). A partir disso foi realizado o *download* das imagens utilizando a API de busca de imagens do Bing, em seguida foi realizado o processo de alinhamento das faces, realizado o treinamento e por fim é executado o reconhecimento. Lembrando que o tempo de treinamento leva em consideração o tempo para encontrar os melhores parâmetros de cada método de classificação, sendo assim, os resultados nos testes foram os obtidos na tabela 1.

Tabela 1 – Tempo e acurácia dos treinamentos

Cenários	KNN		SVM	
	Tempo de treino (s)	Acurácia (%)	Tempo de treino (s)	Acurácia (%)
Cenário 01	59	84	1.5	87
Cenário 02	142	56	8	61
Cenário 03	3376	43	497	45

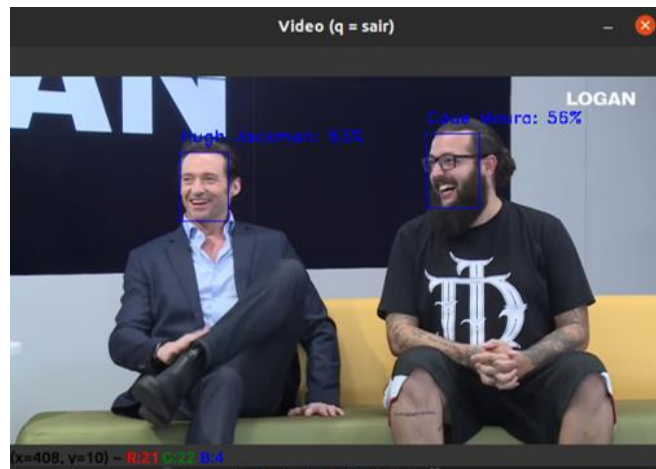
Fonte: Autoria própria (2020).

Olhando a tabela 1, nota-se que a acurácia dos dois métodos de aprendizagem, SVM e KNN, são bem próximos, entretanto o tempo de treinamento é bem diferente.

É bem notável também que quanto mais pessoas são inseridas nos dados o sistema tem uma piora na acurácia, ou seja, em um possível sistema de monitoramento, precisaria de meios para refinar ainda mais a acurácia para o sistema cometer menos erros.

Analisando esse resultado em vídeo, temos os resultados visto nas figuras 1 e 2 que representam o cenário 01 e 03, respectivamente.

Figura 1 – Cenário 01: Hugh Jackman (lado esquerdo); Cauê Moura (lado Direito).



Fonte: Autoria própria (2020).

Figura 3 – Cenário 03: Atila Iamarino (lado esquerdo); Alexandre Ottoni (lado Direito).



Fonte: Autoria própria (2020).

Como vista do resultado de todos os cenários, nota-se que a adição de faces de novas pessoas torna o processo de aprendizado mais difícil, tendo em vista que muitas pessoas possuem características similares em suas faces.

Alguns fatores que influenciam muito no resultado é a utilização de um *dataset* (o LFW) que apesar de ser utilizado para o estudo no reconhecimento de faces, havia muitas faces rotuladas que possui apenas um exemplo e isso influencia em um sistema com poucos dados de algumas classes. Portanto, nota-se a necessidade de que o *dataset* montado tenha bastante exemplos de face de cada pessoa.

Ao final da execução do trabalho e da aquisição dos resultados, foi encontrado o trabalho “*Real-time surveillance through face recognition using HOG and feedforward neural networks*” desenvolvido por (AWAIS, 2019), que possui acurácia acima de 90% em um sistema desenvolvido para o reconhecimento em tempo real e de baixo custo. O trabalho encontrado utiliza a técnica de características HoG (Histograma de gradientes) para a redução da dimensionalidade dos vetores e uma rede neural *feedforward* para a classificação.

CONCLUSÃO

Os resultados iniciais utilizando *deep neural network* para a detecção e reconhecimento facial são promissores, mas é notável que o sistema necessita de melhorias na construção do *dataset*, assim como um melhor refinamento na utilização das técnicas de aprendizagem.

Devido a pandemia não foi possível o teste da aplicação com o *Movidius*, portanto não foi possível analisar o desempenho final no sistema pretendido, então como trabalho futuro, pretende-se realizar o teste em tempo real com a câmera no *Raspberry Pi* e também realizar os testes no mesmo cenários utilizando o método desenvolvido por (AWAIS, 2019).

AGRADECIMENTOS

Agradeço à Universidade Tecnológica Federal do Paraná por fomentar e incentivar a pesquisa, e a todos os seus docentes por proporcionarem um ensino de qualidade.

REFERÊNCIAS

AWAIS, Muhammad et al. **Real-time surveillance through face recognition using HOG and feedforward neural networks**. IEEE Access, v. 7, p. 121236-121244, 2019.

ANAND, A. **LFW - People (Face Recognition): The Labeled Faces in the Wild face recognition dataset**. 2019. Disponível em:
<https://www.kaggle.com/atulanandjha/lfwpeople>. Acesso em: 29 Jul, 2020.

BEZDEK, J. C.; KELLER, J.; KRISNAPURAM, R.; PAL, N. R. **Fuzzy models and algorithms for pattern recognition and image processing**. Springer. vol. 4. 2005.

PEREIRA, C. A. **Detecção de pessoas em imagens, implementando técnicas de visão computacional em um raspberry pi**. 2016. Disponível em:
<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/7474>. Acesso em: 29 Jul, 2020

ROSEBROCK, Adrian. **Raspberry Pi and Movidius NCS Face Recognition**. 2020. data de acesso: 29 de Julho, 2020. [Online]. Disponível em:
<https://www.pyimagesearch.com/2020/01/06/raspberry-pi-and-movidius-ncs-face-recognition/>

RUSSEL, S. J.; NORVIG P. **Inteligência Artificial**. CAMPUS, 2013.

PEDREGOSA, F. **Scikit-learn: Machine Learning in Python**. 2011. vol 12. Journal of Machine Learning Research. 2825–2830 p.