

ESTUDO E PROPOSTA DE UM MODELO GERAL DE UM ALGORITMO PARA AUXÍLIO NO CONTROLE DE DISTANCIAMENTO SOCIAL

Hemili B.A. TRINDADE¹; João P. do N. SILVA²; Diego SAQUI³; Rodrigo C. EVANGELISTA⁴

RESUMO

Atualmente, o mundo vivencia a Covid-19 e como há a falta de medicamentos para o tratamento, tem-se a necessidade de medidas que dificultem a sua proliferação como o distanciamento social. Conseguir esse distanciamento é desafiador e tecnologias de Visão Computacional tornam-se aliadas. Porém, uma das dificuldades na sua utilização está na complexidade de desenvolvimento de algoritmos. Portanto, estratégias adequadas devem ser aplicadas para colaborar nesse processo. Um algoritmo desenvolvido com tecnologia de Visão Computacional para calcular a distância entre as pessoas é estudado neste projeto. O objetivo é entender e extrair um fluxograma para compreender como o código deve se comportar e quais as tecnologias utilizadas, propondo posteriormente um modelo geral que será utilizado na criação de novos algoritmos. Nos resultados iniciais, além da elaboração do fluxograma foram encontradas possíveis melhorias, tais como, a troca da tecnologia utilizada para diminuir o custo computacional e a generalização da distância de detecção.

Palavras-chave: Isolamento; Pandemias; Aprendizado de Máquina.

1. INTRODUÇÃO

Uma eficaz ação para evitar o contágio até que vacinas sejam criadas é pelo distanciamento social (DS). Com o surgimento do Covid-19, tal ação se destacou, pois a principal causa de sua difusão, é pelo contato humano. Com isso, aconselha-se às pessoas a ficarem isoladas, em especial aquelas com o estado de saúde frágil (BENVENUTO, et al. 2019). A preocupação está em manter o DS e monitorar o seu cumprimento, situação dificultada pela falta de recursos. De acordo com Nguyen, et. al (2020), há a possibilidade do uso da Visão Computacional (VC), pois a mesma pode transformar câmeras comuns em “inteligentes” e trazendo para o atual cenário, estas ficam aptas a detectar se há o DS. Para isso é necessário um algoritmo que quando identifica alguma violação no DS, aciona agentes competentes (GHORAI; GAWDE; KALBANDE, 2020).

Recentemente algoritmos de monitoramento de DS têm sido disponibilizados na comunidade científica para melhoramento e implementação. Um exemplo é o algoritmo *OpenCV Social Distancing Detector* proposto por Rosebrock (2020). Algoritmo que faz uso da biblioteca *You Only Look Once (YOLO)* um classificador pré-treinado que trabalha com o aprendizado profundo, passando pelo *frame* em execução uma rede neural e retornando a partir dos seus registros a identificação de objetos. Conforme visto em Nguyen, et. al (2020) o *YOLO* é capaz de detectar um objeto em qualquer distância, visto que mapeia todos os pixels da imagem e reconhece o seu padrão. O aprendizado profundo utilizado no *YOLO* é um ramo do aprendizado de máquina

¹Bolsista PIBIC/CNPQ, IFSULDEMINAS - Campus Muzambinho. E-mail: hemilibetriz@gmail.com

²Bolsista NIPE, IFSULDEMINAS - Campus Muzambinho. E-mail: 12161004203@muz.ifsuldeminas.edu.br

³Orientador, IFSULDEMINAS - Campus Muzambinho. E-mail: diego.saqui@muz.ifsuldeminas.edu.br

⁴Coorientador, IFSULDEMINAS - Campus Muzambinho. E-mail: rodrigo.evangelista@muz.ifsuldeminas.edu.br

baseado em redes neurais artificiais onde suas conexões são feitas como se fossem em camadas com o *input*, o processamento da informação e o *output* que será a entrada da outra camada até o final da rede (DSA, 2019).

O algoritmo de Rosebrock (2020) foi proposto por conta da Covid-19, porém, um entrave é que exige um desempenho elevado do *hardware* o que inviabiliza muitas aplicações. Para contornar esse entrave, adequações nos algoritmos devem ser realizadas para que possam ser utilizados. Uma dificuldade é devido a tais algoritmos utilizarem conceitos e bibliotecas avançadas como aprendizado profundo e a *YOLO*. Sabe-se que uma maneira de estruturar um algoritmo é por fluxogramas que são descritos de forma gráfica e compostos por figuras geométricas que representam diferentes ações. Fluxogramas são um recurso intermediário para compreensão de algoritmos que depois de criados permitem a elaboração de códigos escritos em programação (OBERLEITNER, 2020).

Baseado no contexto apresentado e considerando a necessidade do entendimento, adequação e elaboração de novos algoritmos para DS, neste projeto é estudado o algoritmo *OpenCV Social Distancing Detector* proposto por Rosebrock (2020) e a partir do mesmo proposto um modelo geral de algoritmo para controle de DS. O modelo desse algoritmo é representado por fluxograma, que é considerado como resultado inicial e parcial desta pesquisa, tem o propósito de auxiliar no entendimento de suas principais etapas. Por meio desse modelo, em trabalhos futuros poderão ser realizadas adequações e implementações de novos algoritmos para o controle do DS.

2. MATERIAL E MÉTODOS

Considerando como parte do objetivo deste trabalho o estudo do algoritmo de Rosebrock¹, a primeira etapa foi a implementação do mesmo em computador local para compreensão. O algoritmo está disponibilizado em linguagem de programação *Python* (comum em implementações com aprendizado de máquina devido as suas bibliotecas) e utiliza a biblioteca *YOLO-v3* (versão 3) para fazer a classificação. O estudo foi feito através da plataforma Anaconda, plataforma que já vem com o *Python* instalado e as suas principais bibliotecas, as principais utilizadas foram a *Numpy*, pacote básico que permite trabalhar com arranjos, vetores e matrizes; a *Imutils*, fornece funções básicas de processamento de imagem com a biblioteca *OpenCV* e a própria *OpenCV(cv2)* responsável pelo processamento de imagens para desenvolvimento de aplicativos na área de VC.

Durante o estudo foi realizada a execução do código no computador com vídeos diversos para visualização de sua aplicabilidade. O código foi analisado linha a linha para a extração de cada um dos elementos do modelo geral do algoritmo para controle de DS, que é representado na seção seguinte na forma de um fluxograma onde as etapas também são discutidas.

3. RESULTADOS E DISCUSSÕES

Na Figura 1 é apresentado o modelo geral do algoritmo para controle de DS na forma de fluxograma. Considera-se esse modelo como um resultado parcial da pesquisa em desenvolvimento.

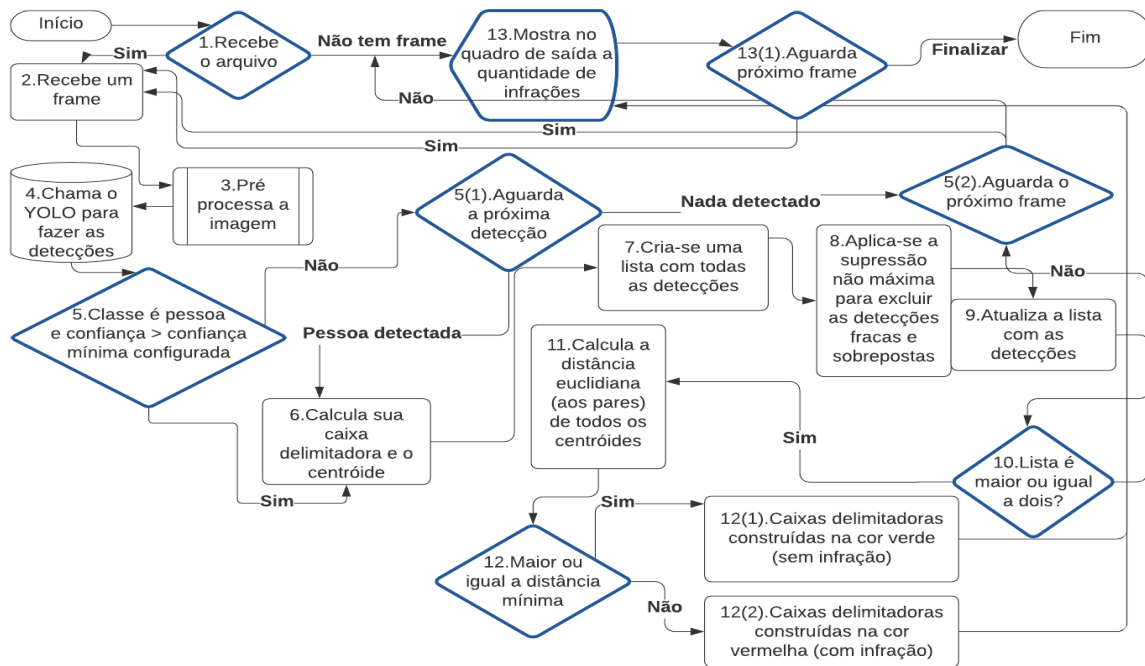


Figura 1. Fluxograma do código para controle de distanciamento social por VC.

Além da biblioteca cv2 que atua em todo o código, destacam-se as seguintes etapas:

Nas **etapas 1, 2 e 3**: é recebido o arquivo *frame a frame*, com o *frame* é feito o pré processamento da imagem com o uso da biblioteca *imutils*; Nas **etapas 4, 5**: uso do código já treinado *YOLO-v3* que faz o reconhecimento dos objetos na imagem, depois filtra a classe de interesse nesse estudo que nesse estudo é a pessoa. Observa-se que o algoritmo identificou com precisão as pessoas em vídeos de câmera. Nessas etapas, aponta-se uma possível melhoria no reconhecimento em qualquer distância e é indispensável que o computador possua GPU e SSD para execução em tempo real, sugerindo possível dificuldade devido ao custo. Nas **etapas 6, 7, 8, 9, 10**: é calculado as dimensões das caixas delimitadoras e os centroides para cada pessoa com o uso da biblioteca *numpy*, gera-se uma lista com os resultados, aplica o comando de supressão não máxima para excluir identificações sobrepostas e/ou fracas e atualiza a lista; Nas **etapas 11, 12, 13**: caso a lista seja maior ou igual a dois é feito o cálculo da distância euclidiana (aos pares) de todos os centroides com o auxílio da biblioteca *numpy*, caso a distância seja menor que a distância mínima é identificado a infração e criada as caixas delimitadoras na cor vermelha para cada pessoa, caso não haja infração ela criada na cor verde, por fim é feito o quadro de saída com as classificações, depois é esperado o próximo frame ou o encerramento do programa.

4. CONCLUSÕES

Neste trabalho foi realizado um estudo sobre um algoritmo para controle de DS por meio de

VC e a partir dele elaborado um modelo geral para essa aplicação. Essa realização visa auxiliar na compreensão de algoritmos de VC para controle de DS que são baseados em estratégias e bibliotecas avançadas. Durante a elaboração do fluxograma, já foram identificadas possíveis alterações:

- Custo computacional: o código estudado apresenta desempenho satisfatório, no entanto, destaca-se que o uso em tempo real do mesmo requer uma disponibilidade *hardware* mais potente e o uso do *YOLO-Lite* é uma alternativa, pois é uma adaptação para computadores sem GPU;

- Identificação de pessoas: pois hoje, o código identifica com precisão apenas em *frames* que as pessoas estão longe da câmera, como as câmeras de segurança, pode-se, portanto, fazer uma configuração para o algoritmo conseguir identificar uma pessoa a qualquer distância.

Em trabalhos futuros, as alterações mencionadas acima serão de importante valor pois quanto maior sua agilidade e precisão, mais adequado o algoritmo será para contribuir no controle do DS em momentos de e pós pandemia. Como o estudo desse tipo de aplicação é recente, outras melhorias poderão ser apontadas, além de novas tecnologias. Posteriormente, o algoritmo também será implementado e testado em situações reais, permitindo obter resultados do seu funcionamento.

REFERÊNCIAS

-BENVENUTO, D., GIOVANETTI M., CICCOCCHI A., SPOTO S., ANGELETTI S., CICCOCCHI M. The 2019-new coronavirus epidemic: Evidence for virus evolution. J Med Virol. 2020;1-5.

-DSA-Data Science Academy. Deep Learning Book, 2019. Disponível em: <<http://www.deeplearningbook.com.br/>>. Acesso: Julho de 2020.

-GHORAI, A.; GAWDE, S.; KALBANDE, D. Digital Solution for Enforcing Social Distancing, 2020. Disponível em: SSRN: <https://papers.ssrn.com/sol3/papers.cfm?abs>. Acesso: Junho de 2020.

-NGUYEN, C. T.; HUYNH, N.; DINH THAI, H.; SAPUTRA, Y.; NGUYEN, T.; NGUYEN, D.; VU, T.; DUTKIEWIC, E.; CHATZINOTAS, S.. “Enabling and Emerging Technologies for Social Distancing: A Comprehensive Survey”. Cornell University. Physics and Society. 2020.

-OBERLEITNER, A. Algoritmos e programação I. Série Universitária. Ed. Senac São Paulo. 2020.

-ROSEBROCK, A. OpenCV Social Distancing Detector, 2020. Acesso: Junho de 2020. Disponível em: <https://www.pyimagesearch.com/2020/06/01/opencv-social-distancing-detector/>.