



TECNOLOGIA DE VISÃO COMPUTACIONAL COMO BASE PARA O MONITORAMENTO NO DISTANCIAMENTO SOCIAL

Hemili B. A. TRINDADE¹; Diego SAQUI²; Rodrigo C. EVANGELISTA³

RESUMO

Atualmente, o mundo vivencia uma pandemia com a Covid-19, e com a escassez de medicamentos, torna-se necessário a adoção de medidas alternativas para combater a sua proliferação. A alternativa mais eficaz é através do distanciamento social, porém, conseguir esse feito tem sido muito difícil. Portanto, tecnologias baseadas em visão computacional tornam-se ferramentas importantes, pois seu custo é relativamente baixo e sua funcionalidade adaptável. Um algoritmo desenvolvido com tecnologia de visão computacional para calcular a distância entre as pessoas em tempo real utilizando o detector de objetos *YOLO*, desenvolvido no ambiente *online* do *Colab* é apresentado neste artigo. Os testes realizados comprovaram a sua boa funcionalidade mesmo com a interferência da qualidade da *internet*.

Palavras-chave: *YOLO*; Aprendizado de Máquina; Pandemia.

1. INTRODUÇÃO

Uma das ações mais eficazes para evitar o contágio de doenças é o distanciamento social (DS). Atualmente, com o Covid-19, tal ação se destacou, pois o principal meio para a difusão da doença é a proximidade entre humanos. Com isso, aconselha-se às pessoas a ficarem em casa o máximo de tempo e quando houver o dever de sair, manter o DS (BENVENUTO, et al. 2019). Nesse sentido, a preocupação está em manter o DS, que de acordo com o Ministério da Saúde é de 2 metros. Situação delicada, pois há a falta de consciência de uma parcela da população e recursos públicos para esse fim. Logo, há a necessidade de explorar opções que ajudem no DS.

De acordo com a revisão realizada por Nguyen, et. al (2020), há a chance do uso da Visão Computacional como um caminho. Pois esta transforma as câmeras comuns em “inteligentes” e trazendo para o atual cenário, estas se tornam aptas a detectar se há o cumprimento do DS. Para isso é necessário um algoritmo que seja executado com as entradas dessas câmeras, que ele processe as

¹Bolsista PIBIC/CNPq, IFSULDEMINAS – Campus Muzambinho. E-mail: hemilibeatriz@gmail.com

²Orientador, IFSULDEMINAS – Campus Muzambinho. E-mail: diego.saqui@muz.ifsuldeminas.edu.br

³Coorientador, IFSULDEMINAS – Campus Muzambinho. E-mail: rodrigo.evangelista@muz.ifsuldeminas.edu.br

informações e, quando identificar alguma violação, alertar sobre o fato. Dessa forma o algoritmo fica responsável pelo monitoramento, e auxilia no combate ao Covid-19 (GHORAI et al., 2020).

Baseado nesse contexto e considerando a necessidade de um método para o auxílio no monitoramento do DS, este estudo traz o detalhamento do algoritmo, assim como a realização de alguns experimentos que demonstram a sua funcionalidade.

2. MATERIAL E MÉTODOS

Na Figura 1 é apresentado o fluxograma do algoritmo. Em seguida, detalha-se a ação das principais tecnologias e métodos empregados em determinadas etapas do fluxograma.

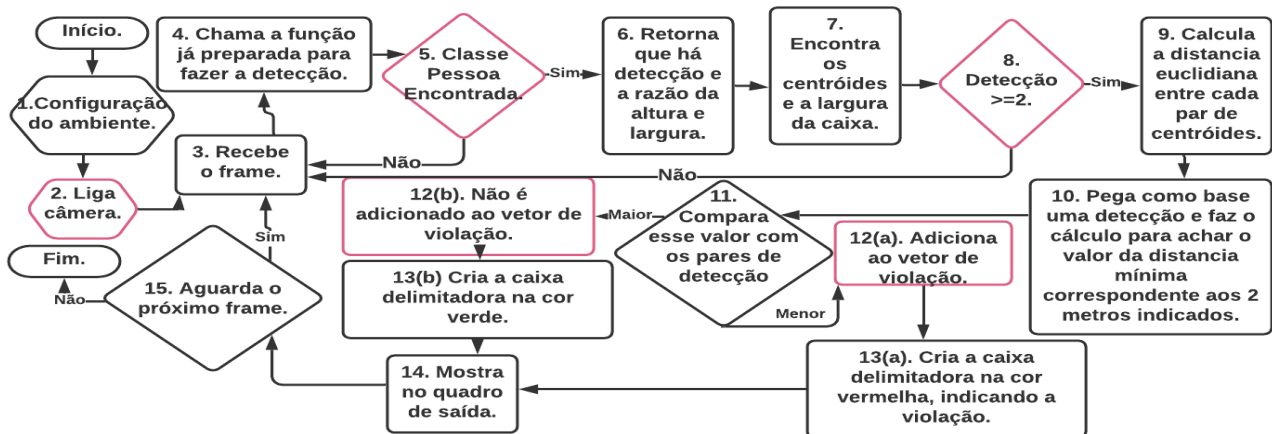


Figura 1 - Fluxograma do algoritmo.

Etapas 1: A linguagem utilizada no projeto foi *Python* devido às bibliotecas avançadas que facilitam esse tipo de aplicação. A tecnologia base foi o detector de objetos em tempo real *YOLOv4*. Este, utiliza uma arquitetura de rede neural convolucional chamada *Darknet*. A base de dados padrão do *YOLO* e utilizada no projeto é a *COCO*, que possui a classe pessoa (de interesse). *YOLO* é implementado com o *framework* para trabalhos com *deep learning* também chamado *Darknet* (BOCHKOVSKIY et al., 2020). Na implementação, optou-se pela utilização do *Google Collaboratory (Colab)*, pelo fato de que nele retira-se a exigência de um *hardware* potente para a execução do código em tempo real. Visto que o *Google* disponibiliza a execução em um servidor que permite a configuração de uma máquina virtual que possua GPUs ou TPU. A infraestrutura do *Colab* está hospedada na plataforma *Google Cloud*, assim o usuário precisa possuir uma conta no *Google* (CARNEIRO et al., 2018).

Etapas 2 até 3: Com o uso de um sistema *online* para implementação, houve a necessidade de configurar o ambiente para poder acessar um *hardware* local (câmera). O próprio *Colab* fornece uma função que faz esse acesso, através de uma API *JavaScript*.

Etapas 4 até 7: Após o algoritmo identificar as pessoas com o *YOLOv4* e retornar as coordenadas da posição das identificações no *frame* em execução, ele coloca em um vetor todas as larguras das caixas

delimitadoras e em outro vetor a posição dos centróides das mesmas.

Etapa 8 até 10: Para calcular a distância mínima segura (DMS) de 2 metros, precisa encontrar a quantidade de pixels equivalente a essa distância de acordo com cada câmera, pois cada uma possui uma resolução diferente. Para isso, padroniza-se o cálculo: utilizando a largura da caixa delimitadora de uma detecção e dividindo pelo valor da distância entre os cotovelos (51 cm). Optou-se pela distância entre os cotovelos pois é uma parte do corpo visível no *frame* e possui o valor padrão médio de 45,9 cm (desvio padrão de 5,1 cm). Utilizou-se a soma dos valores para minimizar os erros (SCHOENARDIE et al. 2010). Assim, calcula-se a DMS: valor encontrado vezes 200 (2 m em cm). No cálculo em si utiliza-se a métrica euclidiana para calcular as distâncias entre as posições dos centróides, depois adiciona essa informação a um vetor.

Etapa 11 até 15: Em seguida, ele toma a primeira identificação e compara a distância entre seu centróide com os outros centróides usando a DMS. Caso essa distância seja menor ele gera a caixa delimitadora vermelha, caso oposto a caixa é verde. Em seguida, passa para o próximo *frame*.

Na próxima seção são demonstrados alguns resultados dos experimentos. Estes, foram feitos em um local isolado, com pessoas selecionadas, de forma que permitiu analisar o comportamento do algoritmo, principalmente, se era capaz de identificar o distanciamento ou proximidade entre pessoas.

3. RESULTADOS E DISCUSSÕES

As Figs. 2a e 2b demonstram o funcionamento do algoritmo em tempo real. Os testes realizados tomaram como base um ambiente que possui aproximadamente 2 metros (2,06 m) que está sendo demarcado com a posição das tomadas elétricas. Realizou-se testes simulando a execução do algoritmo durante o dia e durante a noite (com iluminação inadequada).



Figura 2- a-) Distanciamento respeitado. b-) Com violação no DS.

Apesar do *YOLO* ser considerado o melhor detector para aplicações em tempo real, tomando como base os testes feitos com as métricas do *dataset COCO* (utilizado também para avaliar os sistemas de detecção, BOCHKOVSKIY et al. 2020), nos experimentos, há um atraso na detecção

(visto na Figura 2b pela posição do centróide). Este atraso se deve ao fato do processamento do próprio *YOLO* e da conexão com a *internet* (quanto melhor, mais rápido). Porém, este fato não compromete o funcionamento da aplicação, pois é corrigido em sequência.

4 CONCLUSÕES

Com base nos experimentos, verificou-se que o algoritmo está funcionando adequadamente dadas as configurações disponíveis. Embora a qualidade da câmera usada esteja abaixo do padrão recomendado, o *YOLO* consegue identificar as pessoas e o algoritmo faz o processamento sobre as distâncias e retorna as classificações. Com isso, conclui-se que a aplicação deste projeto pode ser utilizada para auxiliar no monitoramento do DS de maneira eficiente, independentemente do ambiente.

REFERÊNCIAS

- BENVENUTO,D.;GIOVANETTI M.;CICCOZZI A.;SPOTO S.;ANGELETTI S.;CICCOZZI M. The 2019-new coronavirus epidemic: Evidence for virus evolution. J Med Virol. 2020;1-5. doi: 10.1002/jmv.25688.
- BOCHKOVSKIY,A.;WANG, C.;LIAO,H. M.;”YOLOv4:Optimal Speed and Accuracy of Object Detection”.Cornell University.Computer Science > Computer Vision and Pattern Recognition. 2020. Disponível em URL: <https://arxiv.org/abs/2004.10934> . Acesso em 21.06.2021.
- BRASIL.Ministério da Saúde. Como se proteger? Disponível em: <https://www.gov.br/saude/pt-br/coronavirus/como-se-proteger>. Brasília, DF, 2020. Acesso em: 03/08/2020.
- CARNEIRO, T.;NÓBREGA, R. V. D.;NEPOMUCENO, T.;BIAN, G.;ALBUQUERQUE, V. H. C.;FILHO, P. P. R..”Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications”. IEEE, 2018. Disponível em URL: <https://ieeexplore.ieee.org/abstract/document/8485684>. Acesso em 15.06.2021
- GHORAI, A.;GAWDE, S.;KALBANDE, D..”Digital Solution for Enforcing Social Distancing”, 2020. Disponível em URL: SSRN: <https://papers.ssrn.com/sol3/papers.cfm?abs>. Acesso em 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.
- SCHOENARDIE,R.P.;TEIXEIRA,C.S.;MERINO,G.S.A.D.;MERINO,E.A.D.;GONTIJO,L.A.;
“Antropometria: necessidade de constantes investigações para a efetiva contribuição na área da Ergonomia”. EfdPortes, 2010. Disponível em URL: <https://www.efdeportes.com/efd149/antropometria-contribuicao-na-area-da-ergonomia.htm>. Acesso em 12.05.2021.