



Introdução de estudantes à área de inteligência artificial através do uso de visão computacional aplicada à identificação de comportamento de indivíduos em rebanhos

DOI: 10.37702/2175-957X.COBENGE.2022.4028

Diego Tefili - diego.tefili@ifpr.edu.br
INSTITUTO FEDERAL DO PARANÁ

Marcos Santos Hara - mshara@gmx.de
INSTITUTO FEDERAL DO PARANÁ

Beatriz dos Santos Pês - beatriz.santos@ifpr.edu.br
Instituto Federal do Paraná

Jayne Wiezbicki - jayanewiezb@gmail.com
Instituto Federal do Paraná

Nevton Cajé Lima Filho - neytonfilho1@gmail.com
INSTITUTO FEDERAL DO PARANÁ

Anderson Felipe Moreira Ribeiro - andersonfelipe1801@gmail.com
INSTITUTO FEDERAL DO PARANÁ

Resumo: *Este trabalho relata a execução de um projeto em que alunos de um curso técnico em automação industrial integrado ao ensino médio foram desafiados a utilizar programas de visão computacional em um Raspberry Pi para analisar vídeos de animais e identificar padrões de comportamento. Foi utilizada a classe de modelos MobileNets, que foi desenvolvida para aplicações em dispositivos móveis e embarcados, com as bibliotecas OpenCV e Tensorflow compiladas para C++. Esta é uma tentativa de introduzir conteúdos de inteligência artificial para estudantes da educação básica que pode orientar futuras alterações no currículo dos cursos técnicos. As ferramentas indicadas podem ser utilizadas para pesquisa ou extensão com potencial para reduzir a dificuldade inicial de estudantes com pouco conhecimento técnico na utilização de ferramentas de visão computacional.*

Palavras-chave: *inteligência artificial, visão computacional, educação básica, raspberry pi*



Introdução de estudantes à área de inteligência artificial através do uso de visão computacional aplicada à identificação de comportamento de indivíduos em rebanhos

1 INTRODUÇÃO

Reconhecimento de objetos em imagens é uma das tarefas mais típicas da área de aprendizagem de máquina (VON WANGENHEIM et al., 2021). Algoritmos de visão computacional têm sido utilizados para identificação facial, detecção de animais (SANTOSO et al., 2021), ervas, sementes (CHECHLIŃSKI, 2019), pessoas, (HARTAWAN et al., 2019), veículos (ZURAIMI et al., 2021), entre outras aplicações. Entre as plataformas mais utilizadas para aprendizagem de máquina estão Tensorflow, Keras, PyTorch and Caffe. A plataforma Tensorflow foi desenvolvida principalmente para visão computacional, é considerada bastante flexível e pode ser executada localmente em um computador, sem necessidade de enviar dados a um servidor com alto poder de processamento. O Tensorflow é capaz de utilizar vários núcleos do processador ao mesmo tempo, além de possuir vasta documentação online e muitos tutoriais, geralmente focados em aplicações industriais ou voltadas à pesquisa. Esta plataforma pode ser utilizada inclusive em dispositivos com recursos limitados como smartphones e placas Raspberry Pi (KUZMIC e RUDOLPH, 2021.). Em um estudo de comparação entre as quatro principais plataformas, o Tensorflow foi considerado como a plataforma dominante na área (DINGHOFER e HARTUNG, 2020).

Tópicos relacionados à inteligência artificial são comumente ensinados em cursos de computação e engenharia nos níveis de pós-graduação e, eventualmente, de graduação. No entanto, existe uma tendência, principalmente em países desenvolvidos, de introduzir habilidades de programação e inteligência computacional na educação básica (QUALLS e SHERREL, 2010; YU e CHEN, 2018). Existem inclusive relatos de experiências com ensino de aprendizagem de máquina em crianças de 7 a 13 anos (HITRON et al., 2019; KAHN e WINTERS, 2017). No contexto brasileiro, noções de inteligência artificial têm sido introduzidas em cursos de nível médio (FREITAS et al., 2021) e de extensão (SILVA, 2015). Nesta área, as ferramentas tendem a ocultar as etapas mais complexas para usuários iniciantes, ou possuem interface gráfica e o usuário apenas faz ajuste de parâmetros, de modo que resultados possam ser obtidos rapidamente, e os estudantes possam evoluir na aprendizagem antes de usar ferramentas mais avançadas. Em algumas ferramentas, apenas modificações essenciais são possíveis, e o processo de treinamento torna-se uma caixa preta (VON WANGENHEIM et al., 2021). Essa simplificação é criticada por autores que argumentam que isso pode limitar excessivamente a aprendizagem (HITRON et al., 2019).

Este artigo relata a execução de um projeto em que alunos de um curso técnico em automação industrial integrado ao ensino médio foram desafiados a utilizar programas de visão computacional em um Raspberry Pi para analisar vídeos de animais e identificar padrões de comportamento.

2 MATERIAIS E MÉTODOS

Antes de proceder para os *softwares* de inteligência artificial, foi feito um teste de *benchmark* em duas placas: Raspberry Pi 4 com 2GB de memória RAM e processador 1,5 GHz e Raspberry Pi 2B com 1 GB de memória RAM e processador de 900 MHz. O teste foi realizado com o sistema operacional recomendado pela desenvolvedora, o Raspberry OS, com a placa instalada em um *case* com *cooler*, para evitar a redução da

frequência do processador que pode ocorrer durante um superaquecimento. Para a realização deste procedimento, foi utilizado o *software* Geekbench 5 que, após instalado na máquina, realiza simulações de diversas tarefas e tarefas simultâneas como: compressão de arquivos, testes de reconhecimento facial, etc. No fim do processo, é gerado o registro dos testes específicos e uma pontuação geral para a *performance* usando apenas um dos núcleos e para múltiplos núcleos (UserBenchmark, 2022). A Tabela 1 apresenta os resultados de *benchmarks* para dois sistemas operacionais: Raspberry OS, um sistema baseado em Debian, recomendado pela desenvolvedora da placa, e Lubuntu, uma versão do Ubuntu que tem a proposta de ser leve, rápida e de fácil utilização para o usuário. Nos testes realizados, as melhores pontuações da placa foram obtidas com o Raspberry OS, sendo estas: 184 pontos com um núcleo e 567 com os quatro núcleos. *Benchmarks* de memória feitos por GAMESS e HERNANDEZ (2022) apontam o Raspberry Pi 4 como a melhor versão da sua linha, tendo sua pontuação superado as das edições anteriores em quase 100%.

O mesmo *software* de *benchmark* foi utilizado nos outros dispositivos de que os estudantes dispunham, a fim de comparar o poder de processamento da placa com os outros aparelhos. A comparação foi feita com um tablet e dois celulares. Convém observar que o teste foi feito em um Raspberry com o sistema operacional recém instalado, enquanto que os dispositivos de uso diário dos estudantes possivelmente continham outros *softwares* instalados que podem interferir negativamente no resultado do teste.

Tabela 1 - Resultados do Benchmark

Dispositivo	Pontuação (um núcleo)	Pontuação (vários núcleos)
Lenovo IdeaPad 330 81FD 2018	483	1011
Samsung A12	140	814
Galaxy J7 Metal 2016	122	454
Raspberry Pi 2B 1GB RAM Raspbian	49	96
Raspberry Pi 2B 1GB RAM Lubuntu	48	91
Raspberry Pi 4 2GB RAM Raspbian	184	567
Raspberry Pi 4 2GB RAM Lubuntu	182	553

Os estudantes foram introduzidos aos conceitos iniciais de visão computacional a nível de usuário, e com breves noções de desenvolvimento, visto que trata-se de estudantes do ensino médio. Uma placa Raspberry Pi 4 com 4GB de memória RAM com refrigeração ativa foi fornecida aos alunos, juntamente com instruções de instalação e primeiros passos de configuração do sistema operacional padrão para acesso local ou remoto via SSH. Também foi providenciada uma câmera USB com resolução 1080p e referências para as plataformas de visão computacional.

Foi utilizada a classe de modelos MobileNets, que foi desenvolvida para aplicações em dispositivos móveis e embarcados. MobileNets é baseada em uma arquitetura simplificada que constrói redes neurais profundas com pouca exigência de processamento. A construção de modelos é baseada em dois parâmetros globais simples que definem se a prioridade do modelo será latência ou precisão do reconhecimento de

objetos. Isso possibilita que se escolha o tamanho certo dos modelos para cada aplicação. Em comparação com outros modelos populares, essa classe apresenta um bom equilíbrio entre precisão e uso do processador (HOWARD et al., 2017). Inicialmente, foi utilizado um pacote de modelos treinado e fornecido pelo desenvolvedor, capaz de identificar 80 objetos diferentes, como mesas, cadeiras, pessoas e animais. Estes modelos foram utilizados com o conjunto de *softwares* recomendado por Q-engineering (2022): a biblioteca OpenCV, para captura e tratamento de imagens, Tensorflow, para reconhecimento de objetos, e a interface de desenvolvimento CodeBlocks. O desenvolvedor fornece ainda vários códigos-fonte de exemplos de visão computacional, como reconhecimento de objetos, da posição de seres humanos, detecção e reconhecimento facial, entre outros.

As operações de manipulação em vídeos costumam demandar muito esforço do processador, e frequentemente utilizam processamento paralelo. Placas gráficas dedicadas podem chegar a ter mais de mil núcleos. Uma alternativa para aumentar o poder computacional de uma placa como o Raspberry é utilizar um chip gráfico dedicado, como o acelerador gráfico USB Google Coral, que é diretamente compatível com plataformas de visão computacional (GHOSH, 2020). Outra maneira de melhorar o desempenho das operações é escolher adequadamente a linguagem de programação para implementação. As bibliotecas possuem versões em Python e C++. Python é uma linguagem de mais fácil aprendizagem, o que seria conveniente para alunos de ensino médio com pouca ou nenhuma experiência com programação. No entanto, por se tratar de uma linguagem baseada em interpretação de comandos, o desempenho costuma ser menor do que o de programas em C++, por esta última ser uma linguagem baseada em compilação (BARGIACCHI et al., 2020). Optou-se, portanto, por utilizar as versões das bibliotecas compiladas para C++, a fim de se obter maior velocidade de processamento.

3 RESULTADOS

Com as ferramentas instaladas, os estudantes puderam iniciar as atividades. Inicialmente, foi necessário apresentar aos alunos as ferramentas necessárias, através de uma introdução ao sistema linux, manipulação de arquivos e instalação de programas e bibliotecas. Posteriormente, foi feita a apresentação da interface de desenvolvimento para que os alunos pudessem prosseguir com a interpretação dos códigos fornecidos para se familiarizar com os comandos das bibliotecas. Os estudantes foram orientados e estudar os exemplos, pesquisar na documentação sobre cada comando e inserir comentários sobre a função e sintaxe das principais linhas de código.

Atualmente, dois meses após o início do projeto, os estudantes são capazes de executar o reconhecimento de objetos em vídeos gravados ou a partir da câmera, fazer alterações para restringir quais objetos devem ser procurados nas imagens ou vídeos, a quantidade máxima de detecções em um quadro, além de alterar o limiar de confiabilidade para que uma detecção seja considerada válida, e outros ajustes de parâmetros. O *software* utilizado, originário de (Q-engineering, 2022) apresenta bom desempenho de detecção de objetos, com uma taxa de atualização de imagem de cerca de 20 quadros por segundo com os ajustes utilizados atualmente em um Raspberry Pi 4 com 2 GB de memória RAM. Para comparação, um sistema similar em Python (EdjeElectronics, 2022) sendo executado em um Raspberry Pi 2B com 1GB de memória RAM tem uma taxa de atualização próxima de 1 quadro por segundo, utilizando apenas 1 núcleo do processador. Este mesmo programa em Python executado em um Raspberry Pi 4 com 2 GB de memória RAM atinge 5 quadros por segundo. O *software* de Q-

engineering (2022) apresenta uma velocidade maior por ser compilado em C++ e por utilizar os quatro núcleos do processador.

Nos testes iniciais, o sistema foi capaz de reconhecer vários indivíduos numa mesma imagem, mesmo em algumas situações em que um indivíduo se sobrepõe a outro parcialmente. No entanto, o algoritmo atual é um sistema sem memória, pois apenas identifica os objetos em um quadro, mas não aproveita esse resultado para a detecção do quadro seguinte. A próxima etapa do projeto consiste em diferenciar e rastrear indivíduos da mesma espécie em um rebanho. Esta etapa pode exigir que os estudantes executem a etapa de treinamento de um modelo próprio e que encontrem ou desenvolvam uma estratégia para encontrar os mesmos indivíduos em uma sequência de imagens, de modo a possibilitar a geração de um histórico de movimentação de cada animal. Espera-se que até o final do projeto, os estudantes sejam capazes de gerar alertas e relatórios de detecção, e, com base nestes, encontrar padrões de comportamento em vídeos de rebanhos de animais.

4 DISCUSSÃO

Os estudantes foram capazes de instalar e executar os programas com exemplos de aplicações de visão computacional. Por ser o primeiro contato dos estudantes com um computador de placa única, com sistema Linux e pela falta de experiência com programação, foi necessário acompanhamento dos professores e introdução à estrutura da plataforma, instalação dos componentes, navegação na interface de programação, identificação das funções no código e consulta à documentação. Após esses passos iniciais, os estudantes foram capazes de fazer alterações no código. Graças ao grande ganho de velocidade conseguido pelas bibliotecas obtidas de Q-Engineering (2022b), é possível realizar a identificação de objetos com monitoramento em tempo real e rastreamento de movimentação em um dispositivo com poder computacional relativamente baixo, como o Raspberry Pi.

FREITAS et al., (2021) relatam a condução de atividades de ensino remoto relacionado à inteligência artificial na educação básica no sentido de levar os estudantes a entenderem o que é inteligência artificial e quais são os tipos existentes de tecnologias que se enquadram nesta categoria, porém sem implementação e programação. De forma semelhante, Silva (2015) relata uma ação de extensão direcionada a estudantes de ensino médio na qual são discutidas as tecnologias e aplicações da inteligência artificial, sem, no entanto, chegar à etapa de desenvolvimento. No presente trabalho, sugerimos uma plataforma para atividades de ensino, pesquisa ou extensão com potencial para reduzir a dificuldade inicial de estudantes com pouco conhecimento técnico na utilização de ferramentas de visão computacional. As bibliotecas e *softwares* usados são gratuitos e de código aberto e podem ser encontrados nas referências (Q-Engineering, 2022B), e podem ser facilmente instaladas e replicadas. Alternativamente, o mesmo sistema pode ser utilizado em computadores de mesa, portáteis ou dispositivos móveis, devendo-se, para isso, utilizar as versões compatíveis. Em computadores intermediários, a versão Python das bibliotecas é capaz de atingir altas taxas de quadros, não sendo necessária a compilação das mesmas para C++, o que é mais dispendioso. Neste caso, é possível utilizar os exemplos de EdjeElectronics (2022), alterando-se as versões das bibliotecas para a arquitetura e sistema operacional desejados.

5 CONSIDERAÇÕES FINAIS

Este documento relata um projeto de pesquisa no qual estudantes de um curso técnico de automação industrial integrado ao ensino médio com pouca ou nenhuma experiência em programação utilizam ferramentas de visão computacional para reconhecer objetos e padrões de movimentação. Os alunos são capazes de utilizar as funcionalidades básicas das plataformas OpenCV e Tensorflow em um Raspberry Pi modelo 4. O projeto segue em desenvolvimento, na expectativa de que os estudantes sejam capazes de identificar padrões de comportamento em animais a partir de vídeos gravados, e, a partir disso, gerar alertas e relatórios sobre a saúde dos indivíduos. Esses estudantes darão prosseguimento ao trabalho por mais 7 meses, quando devem concluir o ensino técnico. As próximas fases do projeto demandam refinamento na programação e capacidade de desenvolvimento de soluções compatíveis com estudantes que estão concluindo o ensino técnico integrado ao médio. Desta forma, pode-se dizer que a complexidade do projeto cresce conforme os estudantes avançam no curso. Esta é uma tentativa de introduzir conteúdos de inteligência artificial para estudantes da educação básica que pode orientar futuras alterações no currículo dos cursos técnicos.

REFERÊNCIAS

BARGIACCHI, Eugenio; ROIJERS, Diederik M.; NOWÉ, Ann. AI-Toolbox: A C++ library for Reinforcement Learning and Planning (with Python Bindings). **J. Mach. Learn. Res.**, v. 21, p. 102:1-102:12, 2020.

CHECHLIŃSKI, Łukasz; SIEMIĄTKOWSKA, Barbara; MAJEWSKI, Michał. A system for weeds and crops identification—reaching over 10 FPS on raspberry pi with the usage of MobileNets, DenseNet and custom modifications. **Sensors**, v. 19, n. 17, p. 3787, 2019.

DINGHOFER, Kai; HARTUNG, Frank. Analysis of criteria for the selection of machine learning frameworks. **2020 International Conference on Computing, Networking and Communications (ICNC)**. IEEE, 2020. p. 373-377.

EdjeElectronics. **TensorFlow Lite Object Detection on Android and Raspberry Pi**. Disponível em <https://github.com/EdjeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi/blob/master/Raspberry_Pi_Guide.md> Acesso em 15 de Maio de 2022.

FREITAS, Leandro; DO NASCIMENTO, Joelma Aparecida; SANTOS, Michelle Mendes. Inteligência artificial no ensino técnico: estudo comparativo entre duas metodologias no ensino remoto. **Revista Brasileira da Educação Profissional e Tecnológica**, v. 2, n. 21, p. 11456, 2021.

GAMESS, Eric; HERNANDEZ, Sergio. Performance Evaluation of Different Raspberry Pi Models for a Broad Spectrum of Interests. **International Journal of Advanced Computer Science and Applications**, v. 13, n. 2, 2022.

GHOSH, Amit et al. Assistive technology for visually impaired using tensor flow object detection in raspberry Pi and coral USB accelerator. **2020 IEEE Region 10 Symposium (TENSYP)**. IEEE, 2020. p. 186-189.

GRESSE VON WANGENHEIM, Christiane et al. Visual tools for teaching machine learning in K-12: A ten-year systematic mapping. **Education and Information Technologies**, v. 26, n. 5, p. 5733-5778, 2021.

HARTAWAN, Dean Rizki; PURBOYO, Tito Waluyo; SETIANINGSIH, Casi. Disaster victims detection system using convolutional neural network (CNN) method. 2019 IEEE **International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)**. IEEE, 2019. p. 105-111.

HITRON, Tom et al. Can children understand machine learning concepts? The effect of uncovering black boxes. **Proceedings of the 2019 CHI conference on human factors in computing systems**. 2019. p. 1-11.

HOWARD, Andrew G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

KAHN, Ken; WINTERS, Niall. Child-friendly programming interfaces to AI cloud services. **European Conference on Technology Enhanced Learning**. Springer, Cham, 2017. p. 566-570.

KUZMIC, Jurij; RUDOLPH, Günter. Object Detection with TensorFlow on Hardware with Limited Resources for Low-power IoT Devices. **Proceedings of the 13th International Joint Conference on Computational Intelligence**. 2021.

Q-Engineering. **Deep learning examples on Raspberry 32/64 OS**. Disponível em: <<https://qengineering.eu/deep-learning-examples-on-raspberry-32-64-os.html>>. Acesso em 23 de fevereiro de 2022.

_____. **Install TensorFlow 2.4 on Raspberry 64 OS**. Disponível em: <<https://qengineering.eu/install-tensorflow-2.4.0-on-raspberry-64-os.html>>. Acesso em 23 de fevereiro de 2022.

QUALLS, Jake A.; SHERRELL, Linda B. Why computational thinking should be integrated into the curriculum. **Journal of Computing Sciences in Colleges**, v. 25, n. 5, p. 66-71, 2010.

SANTOSO, S. A.; JAYA, I.; IQBAL, M. Design and Implementation of Chaetodontidae Fish Identification Algorithms with Deep Learning Methods. **2021 IEEE Ocean Engineering Technology and Innovation Conference: Ocean Observation, Technology and Innovation in Support of Ocean Decade of Science (OETIC)**. IEEE, 2021. p. 39-44.

SILVA, Sabrina Maria Rodrigues Feliciano da; MASCARENHAS, Suely AN. **Avaliação do rendimento acadêmico em inteligência artificial de estudantes de extensão no IFRO**. 2015.

YU, Yanfang; CHEN, Yuan. Design and development of high school artificial intelligence textbook based on computational thinking. **Open Access Library Journal**, v. 5, n. 9, p. 1-15, 2018.

ZURAIMI, Muhammad Azhad Bin; ZAMAN, Fadhlán Hafizhelmi Kamaru. Vehicle Detection and Tracking using YOLO and DeepSORT. **2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)**. IEEE, 2021. p. 23-29.

Abstract: *This document reports the execution of a project in which students from a vocational highschool course were challenged to use computer vision programs on a Raspberry Pi to analyze videos of animals and identify patterns of behavior. MobileNets model class was used, which was developed for applications on mobile and embedded devices, with OpenCV and Tensorflow libraries compiled for C++. This is an attempt to introduce artificial intelligence to basic education students which may lead to future changes in the curriculae of vocational courses. The suggested tools can be used for teaching, research or extension in order to reduce the initial difficulty of students who may have little technical knowledge in the use of computer vision tools.*

Keywords: *artificial intelligence, computer vision, vocational education, raspberry pi*