

# Aplicação de Aprendizado de Máquina para o Controle de Estoque de Produtos em Processos Industriais

1<sup>st</sup> Diego Gabriel Soares Pivoto  
Centro de Referência em Radiocomunicações (CRR)  
Instituto Nacional de Telecomunicações  
Santa Rita do Sapucaí, Brasil  
diegopivoto@gea.inatel.br

2<sup>nd</sup> Felipe Gustavo de Freitas Rocha  
Instituto Nacional de Telecomunicações  
Santa Rita do Sapucaí, Brasil  
feliperocha@mtel.inatel.br

**Abstract**—A evolução da indústria, impulsionada pela automação, ressalta a importância crucial do controle de estoque nos processos produtivos. Dentro desse contexto, a aplicação de técnicas avançadas, como a classificação de produtos por meio de *Machine Learning* (ML) integrado à *Internet of Things* (IoT) representa uma significativa inovação à comunidade científica, viabilizando algoritmos em *hardwares* de baixo custo. Essa acessibilidade expandida permite a inclusão tecnológica à uma variedade de indústrias e promove a disseminação de sistemas avançados de controle de estoque, independentemente das limitações financeiras. Além disso, a implementação de ML integrado à IoT para o controle de estoque em processos industriais não se limita a uma otimização da eficiência operacional, mas também representa uma resposta prática às demandas contemporâneas por práticas sustentáveis na indústria. Sendo assim, este artigo propõe o desenvolvimento de um algoritmo de ML para classificação de produtos em um processo industrial, com o foco na flexibilidade do modelo treinado para ser introduzido em dispositivos de baixo custo e capacidade computacional. Para tal, um caso de uso foi considerado de forma a classificar diferentes produtos em tempo real durante um processo de transferência dos mesmos por uma esteira, com o objetivo de se aproximar do cenário industrial. Os resultados obtidos mostram que esta técnica pode capacitar a automatização da produção de forma eficiente e reduzir os custos de dispositivos e equipamentos industriais necessários para o seu controle, fortalecendo o comprometimento das empresas com a responsabilidade ambiental, sendo uma resposta integrada aos desafios modernos, alinhando-se aos princípios de democratização, eficiência e sustentabilidade.

**Index Terms**—*Machine Learning, Internet of Things, controle de estoque, processos industriais, classificador.*

## I. INTRODUÇÃO

Desde as primeiras revoluções industriais, testemunhamos uma extraordinária transformação no panorama da produção, impulsionada por avanços tecnológicos significativos. O que começou com processos manuais evoluiu para a automação, redefinindo fundamentalmente a maneira como as indústrias operam. Em meio a esse progresso, destaca-se o papel crucial do controle de estoque, um componente vital em processos industriais, especialmente na esteira da revolução industrial atual, caracterizada pela integração de tecnologias avançadas [1].

No cerne dos processos industriais modernos, o controle de estoque desempenha um papel central na eficiência operacional. Uma ilustração marcante desse avanço é a capacidade dos sistemas automatizados em classificar produtos em uma esteira industrial. Esse processo envolve um classificador inteligente capaz de identificar não apenas a presença de um produto, mas também sua tipologia, permitindo uma contabilização precisa do estoque em tempo real. Esse método revolucionário não apenas simplifica a gestão de inventário, mas também proporciona insights valiosos sobre a produção diária [2].

Ao aprofundar a importância desse controle de estoque avançado, torna-se evidente que a classificação eficiente não se limita apenas à contagem de unidades. Essa abordagem permite a obtenção de informações cruciais, como a taxa de produção de diferentes tipos de produtos. Imagine, por exemplo, um centro de distribuição onde os produtos são automaticamente classificados, facilitando a rápida identificação de itens em alta demanda. Em ambientes de produção mais complexos, como fábricas de produtos eletrônicos, o controle de estoque por meio de classificadores pode ser aplicado para rastrear componentes específicos, agilizando a montagem, reduzindo o tempo de produção e custos [3].

Atualmente, diversas técnicas de classificação impulsionam esse progresso, como por exemplo a visão computacional, utilizando algoritmos de processamento de imagem. Sistemas baseados em regras lógicas e algoritmos de aprendizado tradicionais também são relevantes em determinados contextos, proporcionando flexibilidade na escolha da abordagem mais adequada às necessidades específicas de cada aplicação. Entretanto, ao considerar as diversas abordagens disponíveis, o uso de *Machine Learning* (ML) voltado à *Internet of Things* (IoT) emerge como uma escolha particularmente benéfica para a aplicação de controle de estoque [4].

A capacidade de adaptação dos modelos de ML a padrões complexos e variáveis, combinada com a capacidade de processamento descentralizado em dispositivos IoT, confere uma flexibilidade e escalabilidade notáveis. Além disso, este cenário oferece a capacidade de aprendizado contínuo,

ajustando-se dinamicamente a mudanças nas condições de produção, garantindo uma eficiência e precisão contínuas. Além disso, a integração dessas tecnologias em um *hardware* mais acessível proporciona uma abordagem eficiente e econômica. Sendo assim, a capacidade de implementar algoritmos de ML em dispositivos de menor custo oferece vantagens consideráveis em comparação com os métodos tradicionais, tornando a classificação de produtos mais acessível e eficaz [5].

Baseado nessas informações, este artigo tem como objetivo implementar um classificador de produtos através de técnicas de ML, de modo a possibilitar o seu uso para o controle de estoque em processos industriais. Para tal, um caso de uso foi desenvolvido para classificar três diferentes peças de *hardwares* para computadores, sendo: memória RAM, placa-mãe, e processador. Os testes foram realizados considerando que estes produtos são classificados em tempo-real enquanto são transferidos por uma esteira, de forma a se aproximar do cenário real industrial. Dentre as principais contribuições deste artigo para a comunidade científica, destacam-se:

- desenvolvimento de um algoritmo de ML capaz de otimizar a produção, permitindo uma gestão mais precisa e automatizada do estoque;
- possibilidade de implementação do algoritmo de ML em dispositivos de baixo custo, tornando a tecnologia mais acessível a uma ampla gama de indústrias;
- discussão sobre desafios e trabalhos futuros, de forma a evidenciar possíveis melhorias e ampliar os cenários onde esta aplicação pode ser usada.

Para o melhor entendimento do leitor, as seções a seguir estão organizadas da seguinte maneira: a seção II apresenta os trabalhos relacionados a este artigo. A seção III descreve a metodologia usada para o desenvolvimento deste trabalho. A seção IV detalha o modelo treinado, além de mostrar os resultados obtidos a partir do caso de uso adotado e realizar uma discussão à respeito de desafios, possíveis melhorias e trabalhos futuros. Por fim, a seção V apresenta as conclusões deste trabalho.

## II. TRABALHOS RELACIONADOS

Esta seção busca contextualizar o presente estudo dentro do panorama acadêmico existente, destacando alguns trabalhos que abordaram temáticas similares ou relacionadas ao escopo deste artigo. Um resumo desses estudos anteriores oferece um embasamento teórico e prático para compreender o estado atual da pesquisa e identificar lacunas que esta pesquisa procura preencher.

Em [6], os autores utilizaram uma técnica de visão computacional para detecção de qualidade de produto. O modelo classifica defeitos na superfície de papel do produto. Esses defeitos são dobras, manchas pretas, buracos ou rachaduras. Os resultados foram satisfatórios, já que foi obtida uma acurácia de 99,3%. Porém, não é levado em consideração o atendimento à dispositivos de baixo custo e com baixo consumo de energia.

Em [7], é utilizado o conceito de visão computacional para classificação de vários tipos de produtos em um estoque:

carregador USB, garra robótica, segurador robótico, bocal de uma impressora 3D, extrusora de uma impressora 3D e um copo de sucção robótico. O resultado do modelo foi bom, atingindo uma acurácia de 98,94%, porém um tempo de predição de 230 ms para cada imagem. Os testes foram realizados em um computador Intel CPU Xeon com GPU Tesla K80 e 12 GB de RAM. Dessa forma, não foi considerada a opção de uso de dispositivos de baixo custo e baixo poder computacional.

Em [8], os autores utilizam um braço robótico com visão computacional com o objetivo de realizar inspeção de qualidade em latas de refrigerante. A parte de processamento de imagens foi realizada utilizando os softwares LabVIEW e MATLAB, fazendo uma interface entre uma webcam e um computador. O resultado dos testes obteve uma acurácia de 96%.

Em [9], conceitos de IoT e Visão Computacional são utilizados para monitoramento da área de um produto em tempo real em uma esteira. Os resultados foram observados através da análise de tomates, aqueles com um tamanho inferior a um valor previamente configurado foram classificados como rejeitados. O tempo de inferência do modelo é de 83 ms. Durante os experimentos, o modelo obteve um erro de até 12%.

Em [10], utilizou-se visão computacional para processamento de imagens com o objetivo de identificação de movimentos. As técnicas utilizadas foram algoritmo You Only Look Once (YOLO) e Convolutional Neural Network (CNN). Foram utilizados 3 cenários para testes: uma esteira vazia, uma rodovia com carros e uma esteira com objetos em cima. A eficiência do modelo variou entre 80% e 100%, de acordo com os cenários.

A presente pesquisa buscou explorar uma abordagem inovadora ao considerar o uso de técnicas de ML para a aplicação de tecnologias em dispositivos de processamento limitado e de custo mais acessível. Ao comparar nosso trabalho com estudos anteriores que frequentemente usavam técnicas que se concentravam em dispositivos robustos e de alto desempenho, emergem diferenças e implicações significativas. Nosso trabalho, ao focar em dispositivos mais acessíveis, abre portas para a implementação prática e escalável em cenários onde a disponibilidade de recursos é uma preocupação.

## III. METODOLOGIA

Esta seção tem o intuito de apresentar a metodologia usada para a aplicação de ML ao controle de estoque de produtos em processos industriais, considerando o cenário de IoT, onde o modelo treinado pode ser implementado em dispositivos de baixo custo e capacidade computacional reduzida. Para tal, a Figura 1 ilustra a metodologia aplicada neste trabalho.

Baseado na Figura 1, o primeiro passo da metodologia consiste em determinar o cenário-alvo para a aplicação da técnica de ML considerando o cenário de IoT. Para este trabalho, foi adotado o cenário industrial e, consequentemente, através do passo 2, podemos verificar a escolha da aplicação, que por sua vez refere-se ao controle de estoque em processos

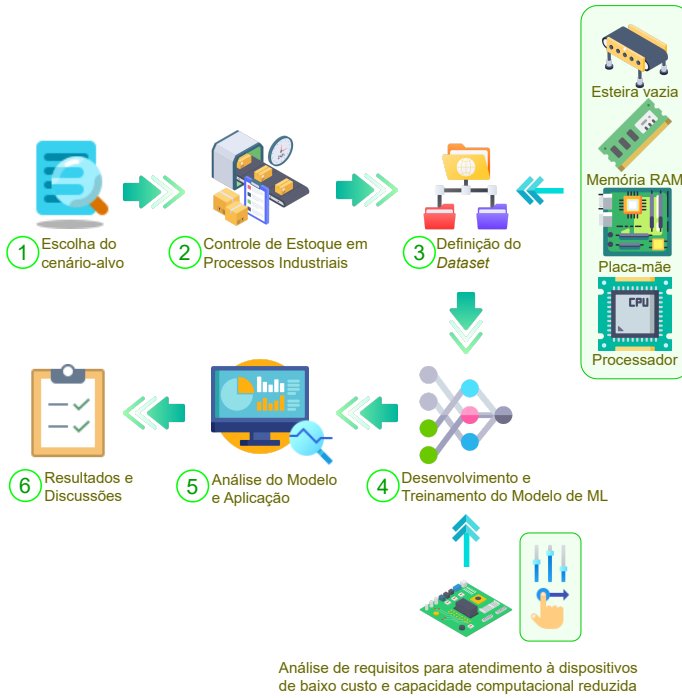


Fig. 1: Metodologia do trabalho proposto para a aplicação de um modelo de ML integrado à IoT para controle de estoque em processos industriais.

industriais. Após a definição do cenário e aplicação, temos a definição do *dataset* apresentada no passo 3. O conjunto de dados escolhido para a realização dos testes foi determinado considerando como caso de uso diferentes produtos de *hardwares* de computadores em estoque se movimentando em tempo-real em uma esteira industrial. Para tal, tem-se para o *dataset* 4 classes distintas: memória *Random Access Memory* (RAM), placa-mãe, processador, e esteira vazia, de forma que o modelo a ser treinado possa detectar quando não há nenhum produto presente no processo industrial de movimento da esteira.

Com a definição do conjunto de dados, o próximo passo definido através trata-se do desenvolvimento e treinamento do modelo de ML. Conforme ilustrado na Figura 1, esta etapa leva em consideração uma análise dos requisitos necessários para atender à dispositivos de baixo custo e capacidade computacional reduzida. Assim sendo, a arquitetura da rede neural, bem como as técnicas utilizadas para redução de complexidade do modelo foram ajustadas para atender a parâmetros específicos de memória RAM e *flash* de um dispositivo de borda de forma a manter um bom desempenho. Uma vez treinado, o modelo é analisado e aplicado ao caso de uso para testes no passo 5. Finalmente, a última etapa da metodologia destaca os resultados obtidos e realiza uma discussão de possíveis melhorias, desafios e trabalhos futuros.

A apresentação e detalhes do modelo escolhido considerando os requisitos de *hardware* a serem atendidos para possibilitar que o mesmo possa ser implementado em dispositivos menos robustos, além dos resultados e discussões, serão

apresentados na seção a seguir.

## IV. APRESENTAÇÃO E RESULTADOS

Para melhor organização do texto, esta seção está organizada da seguinte maneira: a subseção IV-A apresenta o modelo de ML a ser treinado, bem como as características da arquitetura da rede neural usada, considerando como pré-requisitos parâmetros de memória RAM, *flash*, e tempo de inferência para a aplicação. Já a subseção IV-B realiza uma análise dos resultados obtidos através dos testes, pontuando métricas de acurácia, perda e desempenho geral do modelo. Por fim, a subseção IV-C aborda uma discussão à partir dos resultados obtidos, como por exemplo possíveis melhorias, desafios e trabalhos futuros.

### A. Apresentação do Modelo

Para a apresentação do modelo presente neste artigo, primeiramente, foi definida a escolha do cenário-alvo e da aplicação. Conforme previamente detalhado, foi proposta uma aplicação no cenário industrial, especificamente para o controle de estoque em processos industriais. Para o caso de uso adotado, foram definidas 4 classes para compor um *dataset* para o modelo referente à *hardwares* de computadores, sendo: (i) esteira vazia, considerando o fato de que não há nenhum produto em andamento na esteira; (ii) pente de memória RAM; (iii) placa-mãe; e (iv) processador.

A construção do *dataset* foi realizada através da coleta inicial de 75 amostras de cada uma das classes, sendo a fonte fornecida por meio de: capturas de imagens realizadas pela câmera de um celular; imagens disponíveis na Internet; e imagens coletadas em lojas *online* que vendem produtos de *hardwares* de computadores. Além disso, foi usada a ferramenta "Roboflow" [11] para aumentar a quantidade de amostras através de processos de *data augmentation*, como rotação de imagens, além de alterações de brilho e saturação. Após os processos previamente detalhados, o *dataset* foi incrementado, consistindo de 225 amostras de de cada uma das 4 classes, totalizando 900 imagens.

Com a ajuda da ferramenta "Edge Impuse" [12], esse conjunto de dados foi dividido em uma proporção de 80%/20% para treinamento e teste, respectivamente. Desta forma, a distribuição final do conjunto de dados usado para treinar a rede neural foi composta por: 720 amostras totais de treinamento (180 amostras para cada classe); e 180 amostras totais de teste (45 amostras para cada classe). A Figura 2 ilustra um resumo do processo de construção do *dataset* previamente descrito.

Em relação às especificações de *hardware* definidas como alvo para serem capazes de suportar o modelo a ser treinado, deve-se ter um balanço em relação aos requisitos de memória RAM e *flash*, e o tempo de inferência. Esta avaliação deve ser realizada devido ao fato de que, dado que o objetivo do modelo é ser usado em tempo-real considerando que os objetos a serem classificados estão em movimento em uma esteira, o tempo de inferência deve ser relativamente baixo, a ponto da câmera a ser usada captar a imagem e reconhecer o produto antes do

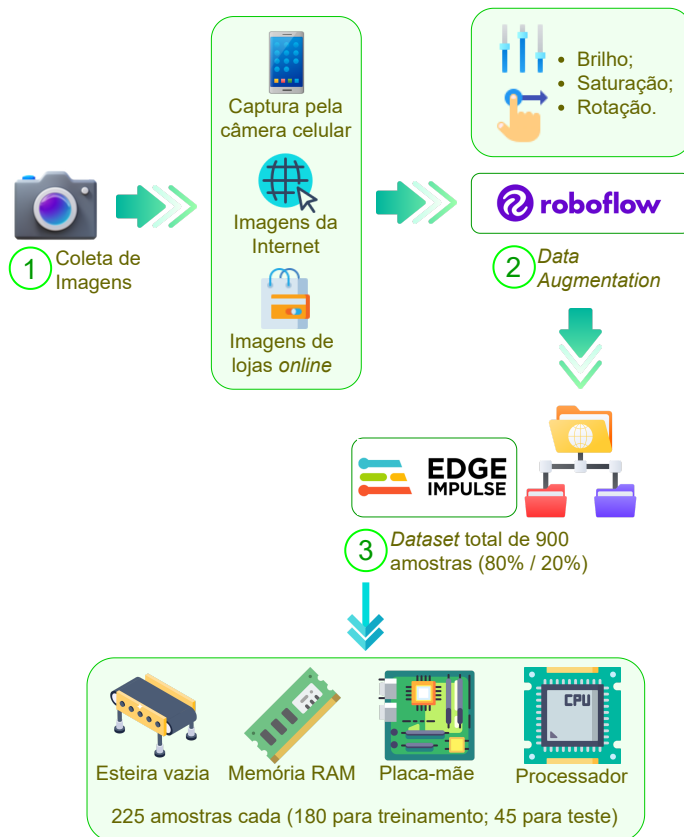


Fig. 2: Construção do *dataset* usado para o desenvolvimento do modelo de ML proposto.

mesmo sair do quadro ideal de visão. Como exemplo, a Figura 3 traz os resultados de desempenho de um modelo treinado considerando apenas 3 classes do *dataset*, onde a acurácia e perda são excelentes. Além disso, podemos observar ainda que o uso de memória RAM no seu pico e de *flash* foram baixos, com apenas 182,6 KB e 59 KB, respectivamente. Porém, o tempo de inferência é de aproximadamente 1397 ms, sendo um tempo consideravelmente alto para poder classificar objetos em movimento, mesmo que a velocidade da esteira não seja relativamente alta.



Fig. 3: Exemplo de modelo com baixo uso de memórias RAM e *flash*, mas alto tempo de inferência.

Baseada nessa premissa, foi definido como principal pré-requisito para atender à aplicação de controle de estoque em processos industriais o tempo de inferência de aproximadamente 15 ms. Esta estimativa prevê uma rápida predição de imagens, podendo ser usada para o caso de uso proposto. Com a ajuda da ferramenta "Edge Impulse", foi utilizado um bloco de processamento de imagem para pré-processamento e normalização de dados, tendo como *dataset* de entrada o conjunto de amostras ilustrado na Figura 2 com ajustes no comprimento e largura para 96x96.

Em relação às imagens, foi adotado o *Red, Green, Blue* (RGB) como parâmetro para extrair as características principais das amostras considerando a escala de cores. A Figura 4 ilustra um resumo das características geradas.

## Feature explorer



Fig. 4: Características extraídas das imagens usando o parâmetro RGB como escala.

Por fim, para o classificador, diversos modelos foram treinados com o objetivo de haver um equilíbrio entre baixo uso de memórias RAM e *flash* e atender a um tempo de inferência de aproximadamente 15ms, ao mesmo tempo em que se obtém um bom desempenho em termos de acurácia e perda. Para tal, foram alterados parâmetros referentes ao número de épocas, taxa de aprendizado, e quantidade de filtros nas camadas convolucionais. Após os testes, obteve-se como melhor modelo a arquitetura de rede neural definida conforme a Tabela I. Este projeto, bem como detalhes do modelo e resultados obtidos, encontra-se disponível através dos *links* do "GitHub" [13] (<https://www.github.com/diegogspivoto/TP557>) e "Edge Impulse" (<https://studio.edgeimpulse.com/public/312119/latest>).

TABLE I: Parâmetros e Arquitetura da Rede Neural

Características de Treinamento	Nº de épocas	14
	Taxa de Aprendizagem	0,0005
	Conjunto de Validação	20%
	Tamanho do <i>Batch</i>	32
Arquitetura da Rede Neural	Camada de Entrada	27648 características
	Camada 2D Conv / Pool	32 filtros, 3 <i>Kernels</i>
	Camada 2D Conv / Pool	16 filtros, 3 <i>Kernels</i>
	Camada Flatten	-
	Dropout	Taxa de 0,25
	Camada de Saída	4 classes

A subseção a seguir apresenta os resultados obtidos usando a arquitetura descrita na Tabela I.

## B. Resultados

Baseada nas configurações de treinamento do modelo da rede neural, bem como na arquitetura definida na Tabela I, a

Figura 5 apresenta os resultados obtidos.



Fig. 5: Resultados obtidos a partir do modelo de ML proposto.

Através da Figura 5, podemos perceber que o modelo treinado obteve um ótimo desempenho, com uma acurácia máxima e 0,02 de perda considerando a entropia cruzada categórica. Para o conjunto de validação, a matriz de confusão apresentada na Figura também mostra que as predições foram realizadas de forma correta.

Do ponto de vista de desempenho no dispositivo, o objetivo-alvo de tempo de inferência de aproximadamente 15 ms foi atendido. Para tal, o uso de memória RAM de pico foi de 363,3 KB, enquanto que o uso de memória *flash* foi de 70,9 KB. Estes valores, consequentemente, permitem que o modelo possa ser treinado em dispositivos de baixo custo e baixo consumo energético, como por exemplo, celulares, e modelos de microcontroladores ESP32 e Raspberry Pi com memória RAM de pelo menos 512 KB, possibilitando usar o espaço restante para desenvolver outras tarefas necessárias para atender determinada aplicação.

Para o caso de uso proposto neste trabalho, esta memória livre possibilitaria, por exemplo, realizar a lógica necessária para o acionamento da esteira, enviar os dados para um servidor via Message Queuing Telemetry Transport (MQTT) ou até mesmo para um *display* contador capaz de identificar a quantidade de produtos de cada classe disponíveis. No nosso exemplo, foi usado um aparelho celular devido à maior resolução, nitidez e clareza da câmera disponível. Desta forma, um motor Direct Current (DC) com alimentação de 3V foi usado de forma a simular uma esteira industrial, colocando imagens dos diferentes dispositivos de *hardware* definidos no



*dataset* (memória RAM, placa-mãe, e processador), além de considerar nenhum produto em andamento.

Sendo assim, as imagens são colocadas em movimento pela esteira executada pelo motor DC. Com o modelo treinado, o dispositivo celular é posicionado em um ângulo superior onde a câmera seja capaz de classificar os objetos que passam pela esteira. A Figura 6 ilustra o teste realizado para o caso de uso proposto em diferentes pontos de vista.

A partir dos testes realizados, podemos perceber que os resultados obtidos foram satisfatórios, visto que o dispositivo celular com o modelo treinado foi capaz de classificar as amostras de teste inseridas na esteira de forma eficiente.

### C. Desafios e Trabalhos Futuros

Baseado nos resultados apresentados na seção IV, diversas discussões relacionadas à desafios, melhorias, e consequentemente trabalhos futuros, puderam ser realizadas. Dentre essas observações mediante o trabalho proposto, podem-se destacar: (i) aprimoramento do *dataset*; (ii) maior redução de complexidade do modelo; (iii) possíveis aplicações adicionais para o algoritmo de ML proposto; e (iv) testes em cenários reais considerando diferentes dispositivos.

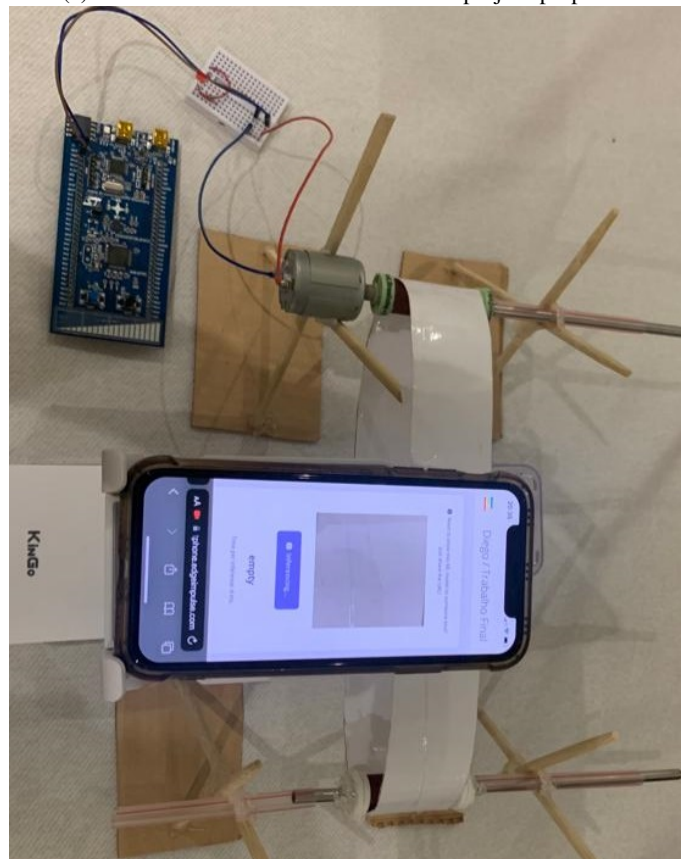
Em relação ao *dataset*, uma possível melhoria pode ser alcançada a partir do aumento do número de amostras de cada classe, considerando por exemplo câmeras de baixo custo com resoluções e nitidez inferiores, além de diferentes ambientes industriais onde a iluminação diária pode ser alterada no decorrer do tempo e consequentemente influenciar no processo de captura de imagem.

Apesar do objetivo deste trabalho propôr uma alternativa de complexidade reduzida, tem-se como desafio uma redução ainda maior. Visto que a memória RAM mínima comercializada em um dispositivo IoT capaz de suportar o modelo proposto deve ser de 512 KB, já considerando memória livre suficiente para realizar outras tarefas de uma possível aplicação, é interessante avaliar a possibilidade de redução de complexidade do modelo de modo que o mesmo possa ser suportado em dispositivos inferiores, como por exemplo microcontroladores de 256 KB ou até mesmo 128 KB de memória RAM. Além disso, esta análise também deve ser realizada para o uso de memória *flash*, ao mesmo tempo em que se mantém um bom desempenho do modelo em termos de tempo de inferência, acurácia e perda.

Como possíveis aplicações adicionais e trabalhos futuros para o algoritmo de ML proposto, destaca-se a inclusão de detecção de anomalias, que para um classificador de imagens pode ser um grande desafio devido ao número gigantesco de características exploradas pelo conjunto de dados, tornando a determinação das características mais impactantes uma tarefa difícil. Além disso, para o controle de estoque, é possível realizar diferentes aplicações de uso do algoritmo treinado, como permitir que o próprio microcontrolador controle o acionamento, parada, e velocidade da esteira, além de executar ações nas suas saídas baseados na classificação realizada, como por exemplo enviar dados em tempo-real referentes à



(a) Visão lateral do cenário de testes do projeto proposto.



(b) Visão superior do cenário de testes do projeto proposto.

Fig. 6: Teste com celular classificando imagens de *hardwares* de computadores se movimentando por uma esteira.

quantidade de cada item do estoque à um servidor *online* via MQTT ou mesmo a um *display* local.

Por fim, tendo em vista que o caso de uso proposto trata-se de um protótipo cujo intuito é demonstrar um cenário industrial com uma esteira em movimento para classificação de produtos em estoque, é importante que o modelo seja testado também no cenário real e com diferentes dispositivos IoT, afim de determinar o mais adequado para a aplicação a ser realizada.

## V. CONCLUSÕES

O presente trabalho se destacou ao desenvolver um classificador eficiente para o controle de estoque em processos industriais, direcionado especificamente para dispositivos de baixo custo, baixa capacidade computacional e baixo consumo energético. A busca por uma solução sustentável e acessível revela-se essencial para permitir que empresas de pequeno porte possam automatizar seus processos de controle de estoque, contribuindo não apenas para a eficiência operacional, mas também para a promoção de práticas mais econômicas e energeticamente conscientes.

Os resultados obtidos, com uma acurácia de 100%, uma perda de 0,02 e um tempo de inferência de 15ms, demonstram a eficácia do modelo proposto. A ênfase na otimização do consumo de memória RAM e flash atende aos pré-requisitos estabelecidos, possibilitando a implementação do classificador em dispositivos com recursos limitados. A busca contínua pela redução do consumo de memória abre portas para uma aplicação ainda mais ampla em dispositivos de baixa capacidade, tornando a solução ainda mais acessível.

As perspectivas futuras destacam a importância da otimização contínua do modelo, visando uma redução ainda maior no consumo de memória. A expansão do conjunto de dados e os testes em ambientes industriais reais proporcionarão uma validação mais robusta da aplicabilidade do classificador em cenários práticos. Dessa forma, este trabalho não apenas apresenta uma solução promissora para o controle de estoque em empresas de pequeno porte, mas também estabelece uma base sólida para avanços futuros na integração de algoritmos de ML em contextos industriais com recursos limitados.

## REFERENCES

- [1] X. Rao, "Inventory Management Algorithm and System Implementation Based on Cost Control," 2023 International Conference on Networking, Informatics and Computing (ICNETIC), Palermo, Italy, 2023, pp. 669-673, doi: 10.1109/ICNETIC59568.2023.00143.
- [2] R. Raffik, D. Rakesh, M. Venkatesh and P. Samvasan, "Supply Chain Control and Inventory Tracking System using Industrial Automation Tools and IIoT," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675774.
- [3] Z. Bingqing, "Classification and inventory control of spare parts based on compound Poisson Demand," 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 2020, pp. 681-685, doi: 10.1109/ICMCCE51767.2020.00150.
- [4] S. K. UmaMaheswaran, V. K. Nassa, B. P. Singh, U. K. Pandey, H. Satyala and M. K. Chakravarthi, "An Inventory System Utilizing Neural Network in The Prediction of Machine Learning Techniques," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1087-1091, doi: 10.1109/ICACITE53722.2022.9823505.
- [5] L. Zhu and P. Spachos, "Towards Image Classification with Machine Learning Methodologies for Smartphones," Machine Learning and Knowledge Extraction, vol. 1, no. 4, pp. 1039-1057, Oct. 2019, doi: 10.3390/make1040059.
- [6] W. Zhengcun and X. Zhongjun, "Product Quality Detection and Recognition based on Vision and Deep Learning," 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC), Zhanjiang, China, 2020, pp. 863-866, doi: 10.1109/YAC51587.2020.9337602.
- [7] A. D. Patel and A. R. Chowdhury, "Vision-based Object Classification using Deep Learning for Inventory Tracking in Automated Warehouse Environment," 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea (South), 2020, pp. 145-150, doi: 10.23919/ICCAS50221.2020.9268394.
- [8] G. Alnowaini, A. Alttal and A. Alhaj, "Design and simulation robotic arm with computer vision for inspection process," 2021 International Conference of Technology, Science and Administration (ICTSA), Taiz, Yemen, 2021, pp. 1-6, doi: 10.1109/ICTSA52017.2021.9406541.
- [9] A. Narzary and S. Ashok, "Real-time monitoring of Conveyor using Computer Vision and IoT," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT), Kannur, India, 2019, pp. 73-78, doi: 10.1109/ICI-CICT46008.2019.8993297.
- [10] S. S. Chiwande, P. Meshram, A. Charde, S. Bhawe and S. Nagdeote, "Machine Monitoring for Industry using Computer Vision," 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2022, pp. 1-4, doi: 10.1109/IATMSI56455.2022.10119424.
- [11] Roboflow, Inc. *Roboflow*. Disponível em: <https://www.roboflow.com>. Acesso em: 20 de novembro de 2023.
- [12] Edge Impulse, Inc. *Edge Impulse*. Disponível em: <https://www.edgeimpulse.com>. Acesso em: 10 de novembro de 2023.
- [13] GitHub, Inc. *GitHub*. Disponível em: <https://www.github.com>. Acesso em: 22 de novembro de 2023.