

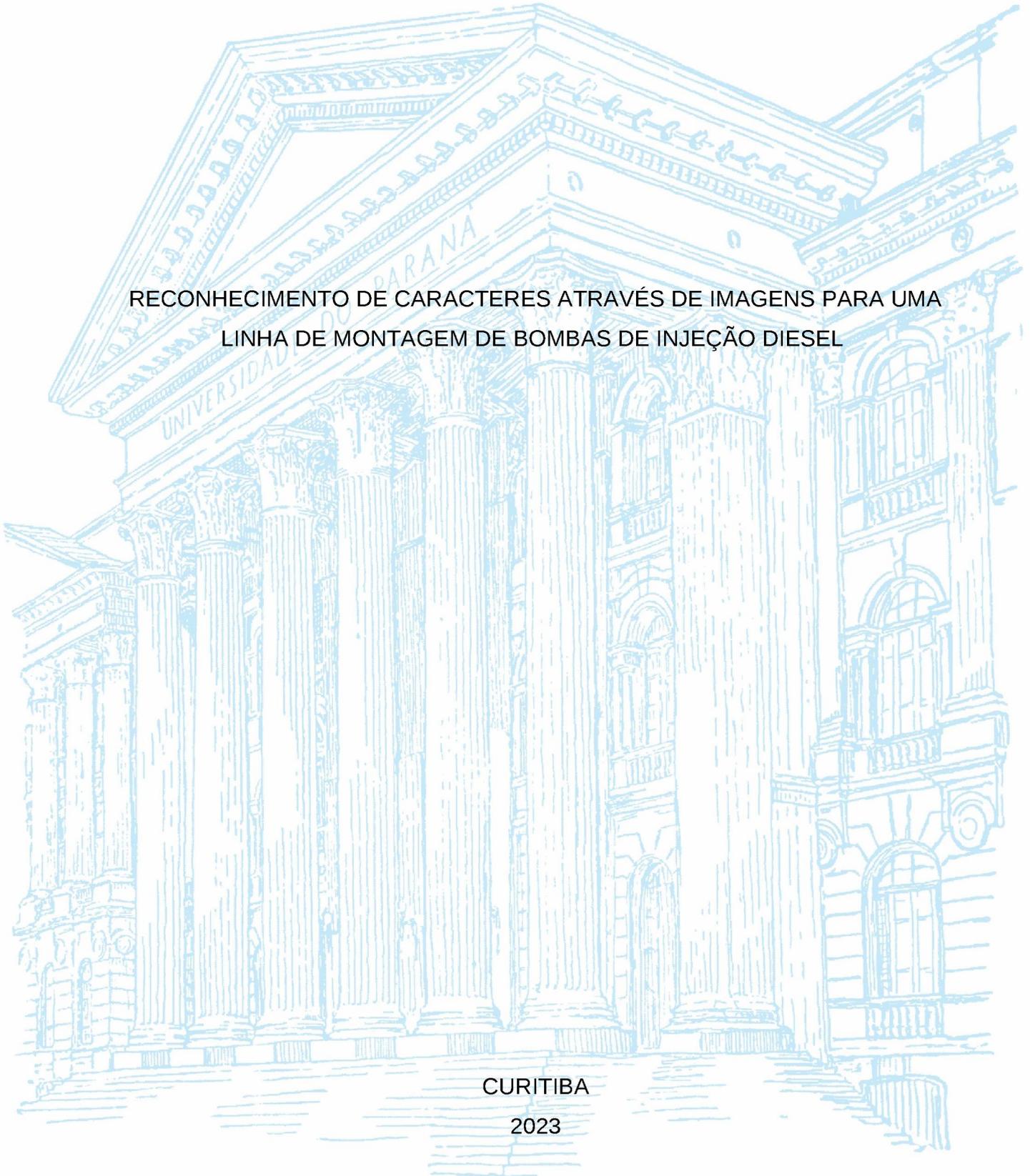
UNIVERSIDADE FEDERAL DO PARANÁ

SAMUEL KOJICOVSKI

RECONHECIMENTO DE CARACTERES ATRAVÉS DE IMAGENS PARA UMA
LINHA DE MONTAGEM DE BOMBAS DE INJEÇÃO DIESEL

CURITIBA

2023



SAMUEL KOJICOVSKI

RECONHECIMENTO DE CARACTERES ATRAVÉS DE IMAGENS PARA UMA
LINHA DE MONTAGEM DE BOMBAS DE INJEÇÃO DIESEL

Trabalho de Conclusão de Curso apresentado ao curso de Especialização em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Jaime Wojciechowski.

CURITIBA

2023

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **SAMUEL KOJICOVSKI** intitulada: **RECONHECIMENTO DE CARACTERES ATRAVES DE IMAGENS PARA UMA LINHA DE MONTAGEM DE BOMBAS DE INJECAO DIESEL**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 15 de Junho de 2023.



JAIME WOJCIECHOWSKI

Presidente da Banca Examinadora



RAZER ANTHOM NIZER ROJAS MONTAÑO

Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Reconhecimento de caracteres através de imagens para uma linha de montagem de bombas de injeção diesel

SAMUEL KOJICOVSKI

Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
kojicovski@gmail.com

Prof. Dr. Jaime Wojciechowski

Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
jaimewo@ufpr.br

Resumo—O aumento contínuo da orientação ao cliente no mercado tem impulsionado a demanda por tecnologias avançadas, como a Inteligência Artificial (IA), em diversas indústrias. Nesse contexto, este artigo acadêmico apresenta um estudo que se concentra no reconhecimento de caracteres em imagens de Bombas de Injeção Diesel (BID) em uma indústria automotiva, utilizando técnicas de IA. O desenvolvimento desse estudo foi conduzido utilizando a linguagem de programação Python, juntamente com os pacotes Tesseract e OpenCV, amplamente reconhecidos por suas capacidades de processamento de imagem e reconhecimento de texto. A precisão do sistema desempenha um papel crucial para a linha de produção, com uma taxa de acerto mínima exigida de 98%. Os resultados alcançados demonstram que o software desenvolvido é capaz de atender às expectativas de acurácia para os quatro tipos de bombas distintas, tornando-se uma solução eficaz para o problema proposto.

Palavras-chave—Bombas de Injeção Diesel, Inteligência Artificial, Python, indústria

Resumo—The continuous increase in customer orientation in the market has driven the demand for advanced technologies, such as Artificial Intelligence, across various industries. In this context, this academic article presents a study focusing on character recognition in images of Diesel Injection Pumps in the automotive industry, using AI techniques. The development of this study was conducted using the Python programming language, along with the widely recognized Tesseract and OpenCV packages, known for their image processing and text recognition capabilities. System accuracy plays a crucial role in the production line, with a minimum required success rate of 98%. The achieved results demonstrate that the developed software is capable of meeting the accuracy expectations for the four distinct types of pumps, making it an effective solution for the proposed problem.

Index Terms—Diesel Injection Pumps, Artificial Intelligence, python, industry

I. DESENVOLVIMENTO

A mudança do mercado com orientação ao cliente tem crescido fortemente nos últimos anos [1]. Essa nova orientação faz com que clientes busquem pelo produto certo, no lugar correto, com preço justo e rapidez [1]. As indústrias automotivas, tecnológicas e biológicas, trouxeram consigo essa nova orientação, fazendo com que outros segmentos a segurem nesse rápido desenvolvimento com

a inserção de novas tecnologias na indústria, conhecida também como Indústria 4.0 [2]. Esses novos modos de atendimento ao mercado e uso de novas tecnologias levam também a aplicação de automação as linhas de montagem industriais, porém, existem ainda muitos processos que são feitos por mão de obra humana [3].

Um dos processos que ainda utiliza mão de obra humana e que pode ser substituído pela automação é a identificação de caracteres, como a leitura de códigos de peças, conhecido como *Optical Character Recognition* (OCR). Esses sistemas utilizam câmeras para a identificação de caracteres. As imagens capturadas pelas câmeras são usadas como entrada de dados, passando por um processo de processamento desses dados e, em seguida, gerando uma saída, que pode ser na forma de imagem, texto ou fala. [4].

Na indústria automotiva existem diversos tipos de automatização realizadas em linhas de montagens, sendo uma delas a utilização de sistemas de OCR. Nas linhas de montagens, há diversos itens, como números de série, que precisam ser controlados para existir um fluxo de funcionamento na linha de produção. Portanto, a proposta deste trabalho é a aplicação de um método de OCR para identificação de caracteres em uma linha de montagem de Bombas de Injeção Diesel (BID) utilizando a linguagem Python.

A. Descrição dos dados

A base de dados utilizada neste estudo consiste em 4888 imagens de bombas de injeção a diesel, obtidas por meio de uma câmera instalada em uma das etapas iniciais do processo de montagem de uma indústria automotiva.

Essas imagens estão classificadas em quatro tipos distintos: 7434, 7433, 7431 e P019. Cada tipo corresponde a um modelo específico de bomba de injeção a diesel com características particulares, que não serão detalhadas neste artigo. A base de dados é composta por 663 imagens do modelo P019, 1138 imagens do modelo 7431, 1912 imagens do modelo 7433 e 1175 imagens do modelo 7434, conforme apresentado na Tabela I.

Tabela I
CLASSIFICAÇÃO DAS IMAGENS DE BOMBAS DE INJEÇÃO A DIESEL

Tipo de Bomba	Quantidade de Imagens
7434	1175
7433	1912
7431	1138
P019	663

Fonte: O autor (2023)

Todas as imagens apresentam características similares, sendo fotografias tiradas da parte frontal da peça. As imagens possuem dois tipos de extensões: a extensão *Joint Photographic Experts Group* (JPG) e a extensão *Windows Bitmap* (BMP), ambas com uma resolução de 800 x 600 pixels. O que diferencia as BID são os caracteres a serem identificados, conforme ilustrado nas Figuras 1 e 2.



Figura 1. Bomba de injeção a diesel: modelo 7434.
Fonte: O autor (2023)



Figura 2. Bomba de injeção a diesel: modelo P019.
Fonte: O autor (2023)

B. Métodos

Nesta seção serão descritos os métodos aplicados para análise das imagens de BID e identificação dos caracteres na imagem. Os métodos utilizados consistem em 3

principais etapas: 1) montagem da base de dados; 2) pré-processamento dos dados; 3) identificação de caracteres com a utilização do Tesseract, uma ferramenta de software livre de identificação de caracteres desenvolvido pela Hewlett-Packard e Google, hoje mantida por Stefan Weil no GitHub [10]. As etapas deste desenvolvimento estão descritas a seguir e o seu diagrama pode ser visto na Figura 3:

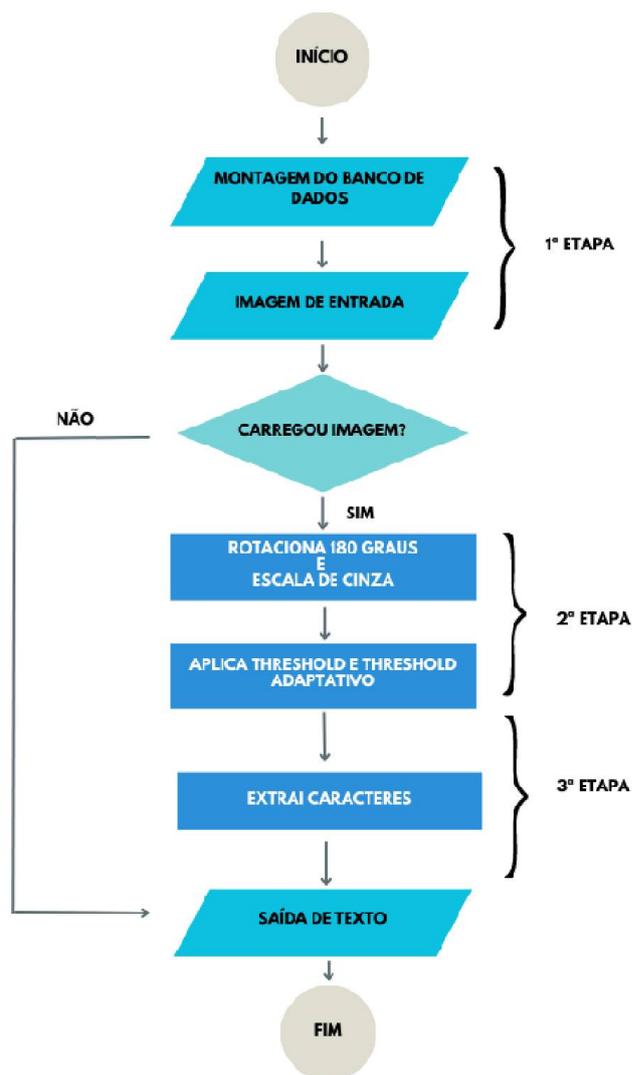


Figura 3. Fluxograma do software.
Fonte: O autor (2023)

1) *Montagem da base de dados*: Os dados utilizados para a construção da base foram retirados de uma linha de montagem com auxílio de uma câmera industrial em uma indústria automotiva em Curitiba, Paraná, Brasil. A companhia possui quatro tipos de BID diferentes na linha de montagem escolhida, modelos: P019, 7431, 7433 e 7434. Separadas previamente por tipo de BID, as imagens serão pré-processadas para então à aplicação da biblioteca pytesseract para identificação dos caracteres.

2) *Pré-processamento dos dados*: Nesta etapa as imagens são pré-processadas para que a ferramenta de OCR consiga identificar de uma forma mais precisa os caracteres desejados, sendo assim, a etapa mais importante de todo o processo de identificação.

No pré-processamento de imagens, dependendo do tipo de aplicação, existem diversas formas de obter o resultado desejado. De acordo com [5], as técnicas de processamento de imagens digitais garantem a versatilidade, repetibilidade e preservação da precisão dos dados originais e as técnicas envolvidas para este pré-processamento podem ser na melhoria da imagem, segmentação e extração de recursos da imagem.

O pré-processamento é realizado para identificar a região de interesse da imagem, conhecida como ROI (*Region of Interest*) no contexto de IA. Neste trabalho, a região de interesse refere-se aos caracteres que precisam ser identificados. Para isso, a imagem é primeiro girada em 180 graus e, em seguida, convertida para escala de cinza. Em seguida, são aplicados métodos de segmentação da imagem. Utilizando a biblioteca OpenCV e a função *threshold*, os valores de intensidade dos *pixels* da imagem são truncados através do parâmetro *TRUNC*, mantendo apenas os *pixels* com maiores valores de intensidade, conforme mostrado na expressão 1. Após a aplicação do filtro, a saída da nova imagem pode ser vista na Figura 4.



Figura 4. ROI após aplicação do método *threshold TRUNC*.
Fonte: O autor (2023)

$$dts(x, y) \begin{cases} threshold & \text{se } src(x, y) > thresh \\ src(x, y) & \text{caso contrário} \end{cases} \quad (1)$$

Onde, *dts* será a imagem resultado, *src* a imagem de entrada e *thresh* o valor que será truncado.

Em seguida, são aplicados dois filtros adaptativos utilizando a função *adaptiveThreshold*. O primeiro filtro utiliza um algoritmo Gaussiano que realiza a soma ponderada dos *pixels* em uma vizinhança, produzindo um resultado. O segundo filtro retorna a média simples dos *pixels* na vizinhança. Nas Figuras 5 e 6, é possível visualizar os resultados de ambos os filtros, respectivamente. Embora haja uma pequena diferença na visualização das Figuras 5 e 6, a utilização desses filtros resulta em um grande ganho na taxa de acurácia na detecção dos caracteres pelo OCR.

A parametrização dos filtros foi feita através de tentativa e erro, analisando o melhor resultado de segmentação para o reconhecimento dos caracteres pelo Tesseract. Para o filtro Gaussiano, utilizado através do método *ADAPTIVE_THRESH_GAUSSIAN_C*, foram utilizados como parâmetros 31 *pixels* de vizinhança e a constante C, utilizada para subtrair do resultado, com valor igual a 4. Para o filtro de média, utilizado através da

método *ADAPTIVE_THRESH_MEAN_C*, foram utilizados 21 *pixels* de vizinhança e a constante C igual a 8.

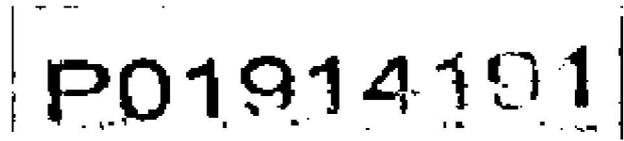


Figura 5. ROI após aplicação do método *Adaptivethreshold GAUSSIAN*.
Fonte: O autor (2023)

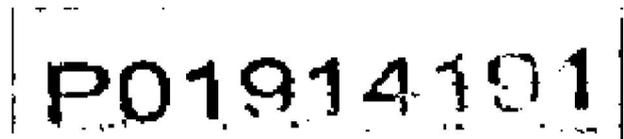


Figura 6. ROI após aplicação do método *Adaptivethreshold MEAN*.
Fonte: O autor (2023)

3) *Identificação dos caracteres com Tesseract*: Nesta etapa, as imagens já estão prontas para serem utilizadas pela ferramenta Tesseract. O Tesseract é uma ferramenta OCR desenvolvida para uso via linha de comando feito em C++, porém, possui uma versão empacotada para a linguagem Python, chamada Pytesseract. A ferramenta pode ser utilizada para mais de 100 línguas e pode ser treinada para novas formas de caracteres. Para este trabalho não foi necessário o treinamento de uma nova língua, pois o que está sendo identificado não são palavras, mas caracteres isolados. O método do Pytesseract utilizado para fazer a identificação dos caracteres é o *image_to_string*, que possui três parâmetros de entrada:

- 1) a imagem de entrada;
- 2) a linguagem a ser utilizada. Para esta aplicação foi utilizada a língua inglesa;
- 3) configuração do Tesseract para a detecção dos caracteres desejados.

Na configuração do Tesseract, é possível inserir alguns parâmetros específicos, conforme Tabela II, para o que deseja identificar, como por exemplo, se irá identificar uma palavra ou um caractere isolado. Essa configuração pode ser feita através do parâmetro *psm* e o valor utilizado para esse parâmetro foi 6, pois apenas um caractere por vez será identificado. Outro parâmetro é o *oem*, que indica qual o motor OCR será utilizado. Para este trabalho foi utilizado o motor OCR Tesseract juntamente com o *Long Short Term Memory (LSTM)*, uma rede pré-treinada [10]. Por último, o parâmetro de configuração *tessedit_char_whitelist* possibilita a inserção de uma lista de caracteres desejados que funciona como um filtro, o qual foram escolhidos os caracteres P0123456789.

Após a configuração do Tesseract, o retorno do método *image_to_string* é uma *string* com os caracteres encontrados na imagem, como pode ser visto no exemplo da Figura 7.

Tabela II
PARÂMETROS DE CONFIGURAÇÃO DO TESSERACT

Parâmetro	Valor
psm	6
oem	Tesseract + LSTM
tessedit_char_whitelist	P0123456789

Fonte: O autor (2023)



Figura 7. ROI da imagem de entrada com caracteres identificados.
Fonte: O autor (2023)

C. Tecnologias

O *script* foi desenvolvido em um computador pessoal Lenovo Ideapad Gaming 3i com processador i5 de 8 núcleos a 2,5 GHz, com sistema operacional Linux Ubuntu 22.04.1 LTS. Foram utilizadas as seguintes aplicações e bibliotecas:

- Linguagem Python, versão 3.10.4;
- Pacote Pytesseract, versão 0.3.10;
- Pacote Opencv-python, versão 4.6.0.66;

II. RESULTADOS E DISCUSSÕES

Nesta seção são apresentadas discussões e resultados obtidos no experimento de identificação de caracteres com a ferramenta Tesseract. Para cada tipo de BID, foi realizada a identificação dos caracteres separadamente, para que pudesse ser medido o número de acertos de acordo com a quantidade de imagens disponíveis de cada bomba de injeção. A subseção a seguir apresenta a acurácia da técnica utilizada e, a subseção seguinte, discussões sobre o trabalho desenvolvido.

A. Medida de qualidade da aplicação

A medida de qualidade para o *software* desenvolvido foi a acurácia. Utilizando o Tesseract combinado com pré-processamento das imagens, a acurácia foi calculada através de uma regra de três simples, no qual, a quantidade de acertos multiplicado por cem é dividido pelo o número total de imagens de uma BID, equação 2. Dessa forma, obtém-se o resultado em porcentagem da acurácia na identificação dos caracteres.

$$acurácia = \frac{A \times 100}{TI} \quad (2)$$

Onde, A são acertos e TI, total de imagens.

Embora o mesmo software faça a detecção de todos os tipos de BID, foram realizados testes de acurácia de forma individual para cada tipo de bomba. Na tabela III é possível ver as acurácias para cada BID, assim como o teste com todos tipos de bombas. Para a bomba P019 a taxa de acerto atingiu 98,19%. As BID 7431 e 7433 alcançaram taxas de acerto de 98,15% e 98,48%, respectivamente. Já a BID 7434 obteve o maior índice de acerto, com uma taxa de 98,72%. Para os testes envolvendo todos os tipos de BID como entrada do software, a acurácia foi de 98,38%. Todos valores obtidos através da equação 2 já mencionada.

Tabela III
TAXA DE ACERTO PARA DETECÇÃO DE CARACTERES

Imagem	Taxa de Acerto (%)
P019	98,19
7431	98,15
7433	98,48
7434	98,72
Todas	98,38

Fonte: O autor (2023)

B. Discussões

O software de detecção de caracteres desenvolvido superou as expectativas ao alcançar uma taxa de acurácia superior a 98%. Essa taxa de acerto é crucial, pois corresponde ao limite tolerado para a linha de produção das bombas de injeção. Quando as peças são detectadas de forma incorreta, é necessário retrabalho por parte dos operadores, o que resulta em aumento do tempo de produção e, conseqüentemente, em custos mais elevados. A detecção errada pode ocorrer devido a sujeira, óleo ou impurezas na superfície onde os caracteres são marcados. Além disso, a falta de um caractere na peça, devido a um erro de gravação em etapas anteriores da sua produção, pode levar ao descarte da peça caso não seja possível uma inserção manual por um operador.

Esses fatores ressaltam a importância de um sistema preciso de detecção de caracteres, como o desenvolvido neste estudo, que visa otimizar o processo de produção e reduzir custos. A utilização desse software proprietário substitui a necessidade de intervenção manual para identificação de caracteres nas bombas de injeção, permitindo um fluxo contínuo e eficiente na linha de produção. Com isso, é possível minimizar os erros de detecção, garantir a conformidade das peças e aumentar a produtividade geral do processo de montagem.

Nesse sentido, um dos objetivos do software desenvolvido, além de uma acurácia acima de 98%, é o de substituir soluções de empresas terceiras que atualmente desempenham o papel de identificação de caracteres das BID. Além de oferecer um custo menor por ser um software

proprietário da companhia, essa solução proporciona a flexibilidade necessária para se adaptar a diferentes linhas de produção que requerem esse tipo de identificação.

Uma das vantagens da utilização da ferramenta Tesseract para a detecção dos caracteres é a facilidade de uso de sua *Application Programming Interface* (API). Combinado com a aplicação de bons filtros nas imagens, é possível transformar o software em uma solução pronta para atender às demandas das linhas de produção, como realizado neste trabalho. No entanto, vale ressaltar que o Tesseract possui diversas outras aplicações possíveis. Por exemplo, no estudo conduzido por [8], foi realizado o treinamento do Tesseract para reconhecimento de texto manuscrito em uma fonte específica. Já no trabalho desenvolvido por [9], foi explorada a capacidade do Tesseract em extrair informações de imagens complexas, em conjunto com a biblioteca OpenCV.

Ao longo do desenvolvimento deste estudo, o foco principal foi criar um sistema altamente eficiente capaz de lidar com a identificação de caracteres nas imagens das bombas de injeção diesel. Os resultados obtidos validam a eficácia do software desenvolvido, fornecendo uma solução confiável para a detecção precisa de caracteres para os tipos de imagens utilizadas.

III. TRABALHOS FUTUROS

A partir das discussões e resultados alcançados neste trabalho, ainda existem melhorias que podem ser aplicadas na detecção dos caracteres. Sendo assim, como proposta para trabalhos futuros, sugerem-se as seguintes ações:

- Realizar testes com diferentes tipos de filtros utilizando o OpenCV ou outras ferramentas de tratamento de imagem, visando aumentar a acurácia do software.
- Comparar a acurácia obtida com a utilização de outras ferramentas similares ao Tesseract, como o pacote EasyOCR.
- Realizar o treinamento de um novo idioma para o Tesseract utilizando as imagens das BID, visando possíveis melhorias na precisão dos resultados.
- Realizar testes na detecção de caracteres em imagens de outros tipos de peças na indústria, ampliando a aplicabilidade do software.

Essas sugestões visam aprimorar ainda mais o desempenho do sistema desenvolvido, possibilitando um maior número de aplicações e resultados mais precisos na detecção de caracteres em imagens industriais.

REFERÊNCIAS

- [1] P. Pierleoni, A. Belli, L. Palma, M. Palmucci and L. Sabbatini, "A Machine Vision System for Manual Assembly Line Monitoring," 2020 International Conference on Intelligent Engineering and Management (ICIEEM), 2020, pp. 33-38, doi: 10.1109/ICIEEM48762.2020.9160011.
- [2] L. Barreto, A. Amaral, T. Pereira, "Industry 4.0 implications in logistics: an overview," vol. 13, pp. 1245–1252, 2017.

- [3] Ribeiro, D., Matos, L.M., Cortez, P., Moreira, G., Pilastrri, A. (2021). A Comparison of Anomaly Detection Methods for Industrial Screw Tightening. In: , et al. Computational Science and Its Applications ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science(), vol 12950. Springer, Cham. https://doi.org/10.1007/978-3-030-86960-1_34
- [4] H. Singh and A. Sachan, "A Proposed Approach for Character Recognition Using Document Analysis with OCR," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 190-195, doi: 10.1109/ICCONS.2018.8663011.
- [5] B. Chitradevi1, P.Srimathi2 "An Overview on Image Processing Techniques," vol. 2, Issue 11, November 2014
- [6] S. Yadav and S. Shukla, "Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification," 2016 IEEE 6th International Conference on Advanced Computing (IACC), 2016, pp. 78-83, doi: 10.1109/IACC.2016.25.
- [7] S. Lee, K. Son, H. Kim and J. Park, "Car plate recognition based on CNN using embedded system with GPU," 2017 10th International Conference on Human System Interactions (HSI), 2017, pp. 239-241, doi: 10.1109/HSI.2017.8005037.
- [8] K. Uezima, L. et al. "Utilizando o Tesseract OCR para reconhecimento de textos a mão," 2021, (UPM) São Paulo, SP Brazil
- [9] Akinbade, D., Ogunde, A. O., Odum, M. O., & Oguntunde, B. O. (2020). An Adaptive Thresholding Algorithm-Based Optical Character Recognition System for Information Extraction in Complex Images. *Journal of Computer Science*, 16(6), 784-801. <https://doi.org/10.3844/jcssp.2020.784.801>.
- [10] Tesseract OCR. GitHub Repository. Disponível em: <<https://github.com/tesseract-ocr/tesseract>>. Acesso em: 15 jun. 2023.