



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

PEDRO CECÍLIO DA CRUZ NETO

**Captura de Valores Glicêmicos em Glicosímetros utilizando Deep Learning:
Estudo de viabilidade com YOLO**

Recife
2025

PEDRO CECÍLIO DA CRUZ NETO

**Captura de Valores Glicêmicos em Glicosímetros utilizando Deep Learning:
Estudo de viabilidade com YOLO**

Trabalho de conclusão do curso apresentado ao curso de Sistemas de Informação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Sistemas de informação.

Orientador (a): Adiel Teixeira de Almeida Filho

Recife

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Cruz Neto, Pedro Cecílio da.

Captura de Valores Glicêmicos em Glicosímetros utilizando Deep Learning: Estudo de viabilidade com YOLO / Pedro Cecílio da Cruz Neto. - Recife, 2025.

44 p. : il., tab.

Orientador(a): Adiel Teixeira de Almeida Filho

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Sistemas de Informação - Bacharelado, 2025.

Inclui referências.

1. visão computacional. 2. aprendizado profundo. 3. reconhecimento óptico de caracteres. 4. monitoramento de diabetes. 5. diabetes mellitus. 6. yolo. I. Almeida Filho, Adiel Teixeira de. (Orientação). II. Título.

000 CDD (22.ed.)

PEDRO CECÍLIO DA CRUZ NETO

**CAPTURA DE VALORES GLICÊMICOS EM GLICOSÍMETROS UTILIZANDO
DEEP LEARNING: ESTUDO DE VIABILIDADE COM YOLO**

Trabalho de conclusão do curso apresentado ao curso de Sistemas de Informação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Sistemas de informação.

Aprovado em: 06/08/2025.

BANCA EXAMINADORA

Prof. Dr. Adiel Teixeira de Almeida Filho (Orientador)

Universidade Federal de Pernambuco - UFPE

Prof^a. Dr^a. Magdala de Araujo Novaes (Examinador Interno)

Universidade Federal de Pernambuco - UFPE

Dedico este trabalho, antes de tudo, aos meus pais,
Andreia e Cleiton, que me fizeram ser quem sou.
Com amor, paciência e coragem, me ensinaram a
acreditar nos meus caminhos.

Aos meus avós, Antônio de seu Dito, dona
Maria, Pedro Cecílio e dona Cleonice, que com força
e ternura, ajudaram a construir os alicerces da
família e deixaram marcas profundas na minha vida.

Aos meus irmãos, Artur e Gabriel, por tudo
que dividimos, risos, desafios, lembranças, e por
serem, todos os dias, motivo de orgulho para mim.

Aos meus tios Júnior (Dedé), Osmário
(Marinho), Luciano (tio Ciano) e Luciana (minha
madrinha, tia Lu), por sempre estarem por perto.

À minha namorada, Nádia (e Lua também),
pelo suporte (em vários sentidos) durante um bom
pedaço dessa jornada.

Dedico também às minhas amigas, Amanda
e Carol, que sempre acreditaram em mim como
pessoa e como alguém capaz de conquistar coisas
boas, como esta graduação.

Aos amigos do peito, os rolezeiros, Wytoria,
João Pedro, Júnior, Thaís, Gilberto, Pedro Luna,
Clara, Davi e Gervásio, por me carregarem em
muitos momentos, com pedacinhos de críticas
sinceras, conselhos, humor, pizza, festas e, acima de
tudo, companheirismo.

Aos amigos que fiz em Recife, Salgueiro e
Petrolina também, todos eles.

A Nikko, Melissa, Ed, Napoleon, Brutus,
Mickey e todos os cachorros, gatos e animais do
mundo todo.

AGRADECIMENTOS

Agradeço, com carinho, a todas as pessoas mencionadas na dedicatória, por fazerem parte também da construção deste trabalho.

Ao meu orientador, professor Adiel, pela orientação durante o desenvolvimento do projeto, e aos professores do CIn e da UFPE, que contribuíram ao longo da graduação com saberes essenciais para minha formação.

Aos colegas com quem compartilhei estudos e projetos durante a graduação, e aos amigos que fiz durante meu estágio no Projeto da Motorola, Gabriel, Anderson, Arthur, Rodolfo, Letícia, Wallace, Mateus, Amanda, Dayvison e tantos outros, por todos os aprendizados, trocas e risadas.

E aos meus tios, mais uma vez, Osmário, Luciana e Luciano, por contribuírem com observações valiosas, baseadas em suas experiências na área da saúde. Cada comentário e incentivo de vocês deixou este trabalho mais forte.

“A brilliant man once said, ‘any sufficiently advanced technology is indistinguishable from magic.’” (Doctor Who, episódio “The Witchfinders”, 2018).

RESUMO

O monitoramento regular da glicemia é essencial para o controle do diabetes, especialmente entre usuários de glicosímetros sem conectividade digital. Neste contexto, este trabalho propõe uma solução baseada em visão computacional para automatizar a leitura dos valores exibidos em displays de sete segmentos desses dispositivos. A abordagem utiliza o modelo YOLOv8, uma arquitetura de detecção de objetos de única etapa (single-stage), para identificar simultaneamente os dígitos individuais presentes na tela. A partir dessas detecções, é aplicada uma heurística de ordenação posicional que reconstrói automaticamente o valor glicêmico exibido. Foram avaliadas diferentes versões do modelo (v8n, v8m, v8l, v8x), sendo o YOLOv8m (medium) o que apresentou melhor desempenho, alcançando 94,10% de precisão, 93,41% de revocação e mAP@0.5 de 94,98%. Os resultados demonstram a viabilidade técnica da proposta e sua potencial aplicação em sistemas móveis de baixo custo ou plataformas de saúde digital. A solução apresenta vantagens significativas sobre métodos tradicionais baseados em OCR, especialmente pela robustez em condições adversas de iluminação e ângulo. Este trabalho contribui para a construção de alternativas na automação no monitoramento glicêmico e abre caminhos para futuras implementações em ambientes clínicos e de saúde pública.

Palavras-chave: visão computacional, aprendizado profundo, reconhecimento óptico de caracteres, monitoramento de diabetes, diabetes mellitus, monitoramento glicêmico, You Only Look Once, yolo.

ABSTRACT

Regular blood glucose monitoring is essential for diabetes management, especially among users of glucometers without digital connectivity. In this context, this study proposes a computer vision-based solution to automate the reading of values displayed on seven-segment screens of such devices. The approach uses the YOLOv8 model—a single-stage object detection architecture—to simultaneously detect individual digits present on the screen. Based on these detections, a positional sorting heuristic is applied to automatically reconstruct the displayed glycemic value. Different versions of the model (v8n, v8m, v8l, v8x) were evaluated, with YOLOv8m (medium) showing the best performance, reaching 94.10% precision, 93.41% recall, and a mAP@0.5 of 94.98%. The results demonstrate the technical feasibility of the proposal and its potential for integration into low-cost mobile systems or digital health platforms. The solution presents significant advantages over traditional OCR-based methods, particularly in terms of robustness under adverse lighting and angle conditions. This work contributes to the construction of alternatives in automation in glycemic monitoring and paves the way for future implementations in clinical and public health settings.

Keywords: computer vision, deep learning, optical character recognition, diabetes monitoring, diabetes mellitus, glycemic monitoring, You Only Look Once, yolo.

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 MOTIVAÇÕES.....	14
1.2 PROBLEMA DA PESQUISA.....	16
1.3 OBJETIVOS.....	17
1.3.1 Objetivo Geral.....	17
1.3.2 Objetivos Específicos.....	17
1.4 ESTRUTURA DO TRABALHO.....	18
2 REVISÃO BIBLIOGRÁFICA.....	19
2.1 DIABETES MELLITUS E MONITORAMENTO GLICÊMICO.....	19
2.2 VISÃO COMPUTACIONAL EM DISPOSITIVOS MÉDICOS.....	21
2.3 OCR X YOLO.....	22
2.4 YOLO, UM DETECTOR SINGLE-STAGE.....	23
2.5 TRABALHOS RELACIONADOS.....	25
3 METODOLOGIA.....	28
3.2 AMBIENTE.....	28
3.3 CONJUNTO DE DADOS.....	29
3.4 AGRUPAMENTO POSICIONAL E REGISTRO TEMPORAL DE VALORES.....	32
4 RESULTADOS E DISCUSSÃO.....	35
4.1 AVALIAÇÃO DOS RESULTADOS.....	35
5 CONCLUSÃO.....	43
REFERÊNCIAS.....	45

1 INTRODUÇÃO

1.1 MOTIVAÇÕES

O diabetes mellitus é uma doença crônica caracterizada pela elevação persistente dos níveis de glicose no sangue, e constitui um dos maiores desafios de saúde pública no Brasil. Segundo o Ministério da Saúde (BRASIL, [s.d.]b), estima-se que cerca de 6,9% da população brasileira viva com diabetes, sendo essencial o controle rigoroso da glicemia para a prevenção de complicações como doenças cardiovasculares, retinopatia e insuficiência renal. Para muitos pacientes, o registro diário da glicemia é uma prática essencial, frequentemente realizada de forma manual, por meio de anotações em cadernos ou planilhas. No entanto, esse método de anotação manual apresenta desafios significativos, como erros de transcrição, perda de registros e dificuldades no acompanhamento da evolução dos dados ao longo do tempo.

Embora existam glicosímetros modernos com conectividade digital, como Bluetooth e USB, grande parte dos dispositivos ainda opera de forma offline, exigindo que o paciente registre manualmente os valores exibidos na tela (FINNEGAN et al., 2019). Isso pode levar a dificuldades na gestão do tratamento e na interpretação dos dados pelo profissional de saúde, afetando a personalização do cuidado ao paciente. Além disso, pacientes idosos ou com dificuldades motoras podem encontrar obstáculos adicionais no registro manual, tornando a automação desse processo uma necessidade urgente (GROUP, 2022).

Durante o desenvolvimento de soluções digitais para o monitoramento da glicemia em pessoas com diabetes, um dos maiores entraves identificados é a necessidade de entrada manual dos valores aferidos. Em estudo realizado por Zhang et al. (2018), observou-se que 70% dos pacientes com diabetes tipo 1 consideram a inserção manual dos dados de glicose como uma tarefa incômoda ou levemente incômoda, reforçando a importância de alternativas automatizadas para o registro desses dados com maior eficiência e menor esforço do usuário.

Diante de um contexto em que as tecnologias de visão computacional e inteligência artificial se apresentam como alternativas viáveis para automatizar a leitura e registro de valores glicêmicos (e de pressão arterial, oximetria, etc), o reconhecimento óptico de caracteres (OCR) vem sendo uma técnica amplamente empregada para extrair informações textuais de imagens, com bibliotecas de acesso livre e modelos populares. No entanto, essa técnica pode acabar enfrentando

difficultades significativas em displays de sete segmentos - comuns em glicosímetros e outros aparelhos de cuidados em saúde - devido à características como a sua baixa variabilidade visual e limitações estruturais.

Apesar do potencial da automação no autocuidado do diabetes, Kirwan et al. (2013) alertam que a eliminação completa da entrada manual pode comprometer a percepção ativa do paciente sobre sua condição, levando a uma gestão menos consciente da doença. Ou seja, em certos contextos, manter algum nível de entrada manual pode contribuir para uma reflexão mais ativa sobre o próprio processo de autocuidado. Dugas et al. (2020) destacam que, embora a redução da carga de entrada de dados seja um dos principais objetivos das soluções digitais em saúde (como aplicativos e tecnologias que automatizam o controle do tratamento), pacientes estão dispostos a investir esse esforço adicional quando isso proporciona maior compreensão sobre sua jornada de autogestão da diabetes. Nesse sentido, a proposta deste trabalho não busca substituir a responsabilidade do paciente, mas sim estudar alternativas e oferecer uma proposta inicial de apoio tecnológico que torne mais fácil, rápido e preciso o registro dos dados, mantendo a atenção do usuário no processo e possibilitando o compartilhamento facilitado com profissionais de saúde.

Soluções baseadas em OCR tradicional, como o Tesseract por exemplo, enfrentam dificuldades significativas na leitura de dígitos em displays de sete segmentos (comuns em glicosímetros) devido à natureza segmentada desses dígitos e à alta dependência de pré-processamentos específicos (GHUGARDARE et al., 2009; LIU, 2016; PRAKRUTHI et al., 2017). Mesmo com ajustes de contraste, remoção de ruído ou segmentação adaptativa, os erros de leitura persistem, especialmente em ambientes com iluminação desfavorável, reflexos ou distorções ópticas (PRAKRUTHI et al., 2017; LOW, L. M. et al., 2024). Por outro lado, abordagens baseadas em redes neurais profundas (deep learning), como os modelos de detecção da família do YOLO, oferecem maior robustez e generalização. Sua aplicação na detecção de dígitos em displays segmentados tem se mostrado promissora, dispensando muitas das etapas manuais de segmentação e alcançando desempenho bom, mesmo em condições adversas (BONILLA-RIVAS et al., 2022; LOW, L. M. et al., 2024).

Esse tipo de abordagem tem ganhado relevância na academia, utilizando a capacidade de captura e processamento de imagens dos celulares, uma vez que

“estão disponíveis para quase todos e são capazes de executar tarefas computacionais leves [...] a imagem do dispositivo médico é capturada pelo celular, e então o processamento de imagem é aplicado para reconhecer automaticamente os valores aferidos” (ASADI; GOUDARZI, 2023).

No Reino Unido, por exemplo, o NHS lançou em parceria com a empresa GlucoRx um aplicativo de leitura automática de telas de glicosímetros, desenvolvido por pesquisadores da Universidade de Cambridge, utilizando visão computacional em tempo real (CHARLES, J. et al., 2020). O aplicativo foi projetado para usuários com limitações motoras ou cognitivas, ampliando a acessibilidade à gestão do diabetes em populações mais vulneráveis (UNIVERSITY OF CAMBRIDGE, 2020). Ou seja, essas alternativas podem representar avanços significativos tanto do ponto de vista tecnológico quanto da saúde pública, dispensando sensores adicionais ou integração com o hardware dos dispositivos de saúde em questão. Essas características presentes que incorporam técnicas modernas envolvendo aprendizado profundo, tornam soluções semelhantes escaláveis em cenários de baixa disponibilidade tecnológica e úteis em sistemas de telemedicina ou aplicativos móveis voltados para gestão de doenças crônicas como a diabetes.

1.2 PROBLEMA DA PESQUISA

Com a ausência de conectividade digital na maior parte dos glicosímetros disponíveis no mercado, surge a necessidade de métodos automatizados e acessíveis para captura de valores glicêmicos exibidos em seus visores. Dependendo da situação, os visores podem conter, além dos dígitos de interesse, múltiplos elementos visuais, como relógios, ícones, códigos ou unidades de medidas (que por questões de praticidade serão ignorados inicialmente na proposta), dificultando a segmentação dos dados mais importantes (o índice glicêmico em si).

Diante do exposto, este estudo busca responder à seguinte questão de pesquisa:

Como um modelo de detecção de objetos, como o YOLO, pode ser utilizado para identificar, classificar e reconstruir automaticamente os valores glicêmicos exibidos em glicosímetros, que contam com visores de sete segmentos, a partir de imagens capturadas por dispositivos móveis?

A pesquisa investigará a viabilidade de usar o YOLOv8 (e suas versões) para a detecção direta de dígitos individuais e, a partir disso, propor um método baseado

numa heurística de ordenação posicional para compor o valor numérico final de forma automática e confiável. Essa abordagem será comparada a soluções anteriores baseadas em OCR tradicional, outras técnicas que usem aprendizado de máquina ou aprendizado profundo, e versões diferentes do YOLO, considerando fatores como precisão, robustez e adaptabilidade.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

O objetivo deste trabalho é estudar, avaliar soluções já criadas e propor uma solução baseada no modelo YOLOv8 para detectar automaticamente valores glicêmicos exibidos em visores de glicosímetros, a partir de imagens capturadas por dispositivos móveis.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral, este trabalho seguirá os seguintes objetivos específicos:

1. Avaliar os limites do OCR tradicional e o uso de modelos de aprendizado profundo em contextos com displays de sete segmentos em dispositivos médicos como glicosímetros.
2. Treinar e validar um modelo YOLOv8 para detecção de dígitos individuais (0 a 9) a partir de um dataset anotado (disponível abertamente no roboflow, com 7449 imagens)
3. Desenvolver um algoritmo baseado em uma heurística de agrupamento posicional para reconstruir os valores numéricos a partir das detecções.
4. Testar a solução em imagens externas ao treinamento, simulando um uso prático em aplicação móvel.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

- Capítulo 1 – Introdução: Apresenta o contexto do problema, relevância do monitoramento glicêmico automatizado, os objetivos gerais e específicos do estudo, além da justificativa e da organização do documento.
- Capítulo 2 – Revisão Bibliográfica: Aborda os fundamentos sobre o diabetes mellitus, métodos tradicionais de registro glicêmico, tecnologias emergentes em saúde digital (mHealth), reconhecimento óptico de caracteres (OCR), redes neurais convolucionais, detectores de objetos, e uma análise de trabalhos relacionados focados na leitura de displays de sete segmentos. Também introduz o modelo YOLOv8 e suas variantes, com base em estudos recentes.
- Capítulo 3 – Metodologia: Descreve a construção e anotação do conjunto de dados utilizado, o processo de treinamento e validação dos diferentes modelos YOLOv8 (nano, medium, large e extra large), os critérios adotados nas métricas de avaliação e o mecanismo de pós-processamento via heurística posicional para reconstrução dos valores glicêmicos.
- Capítulo 4 – Resultados e Discussão: Apresenta os resultados quantitativos obtidos, incluindo métricas como precisão, revocação, F1-score e mAP, além da matriz de confusão e curvas de desempenho. Discute também as observações qualitativas sobre as predições visuais e a eficácia da heurística implementada.
- Capítulo 5 – Conclusão: Resume os principais achados do estudo, avalia as contribuições práticas e científicas da solução proposta, e sugere aprimoramentos e linhas de pesquisa para continuidade do trabalho.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta a fundamentação teórica necessária para o desenvolvimento deste trabalho, abordando aspectos sobre diabetes mellitus e a importância do monitoramento glicêmico, conceitos fundamentais de visão computacional e reconhecimento óptico de caracteres (OCR), bem como técnicas e ferramentas utilizadas para a leitura automática de valores glicêmicos em displays de glicosímetros.

A revisão bibliográfica se baseia em estudos recentes sobre monitoramento digital da diabetes, OCR aplicado a dispositivos médicos e visão computacional para reconhecimento de números em telas de sete segmentos, garantindo um embasamento teórico sólido para a discussão proposta.

2.1 DIABETES MELLITUS E MONITORAMENTO GLICÊMICO

O diabetes mellitus é uma doença crônica metabólica caracterizada pela hiperglicemia persistente, causada por falhas na produção e/ou na ação da insulina, hormônio responsável pela regulação da glicose no sangue. No Brasil, estima-se que mais de 13 milhões de pessoas convivam com a doença, sendo considerada uma das principais causas de mortalidade e hospitalizações no país (BRASIL, [s.d.]b). O controle glicêmico adequado é fundamental para evitar complicações como infarto, acidente vascular cerebral, amputações, insuficiência renal e retinopatia diabética, que podem comprometer gravemente a qualidade de vida dos pacientes (BRASIL, [s.d.]a).

Para isso, é essencial que a pessoa com diabetes realize o monitoramento regular da glicemia e mantenha registros precisos dos valores obtidos ao longo do dia. Esses dados são fundamentais para a definição da conduta clínica, possibilitando ajustes na alimentação, no uso de insulina e em outras medicações de acordo com a evolução do quadro (BRASIL, [s.d.]c). O monitoramento glicêmico pode ser feito de diferentes formas:

1. Glicosímetros tradicionais: Pequenos dispositivos portáteis que requerem uma amostra de sangue para medir a glicose. Os valores são exibidos em um display de sete segmentos, e o paciente deve anotá-los manualmente (FINNEGAN et al., 2019).

2. Dispositivos conectados (Bluetooth/USB): Alguns modelos modernos permitem a transferência automática dos dados para aplicativos móveis ou sistemas de saúde, reduzindo a necessidade de anotações manuais (GROUP, 2022).
3. Monitores contínuos de glicose (CGMs): Sensores implantáveis que monitoram a glicose em tempo real e transmitem os dados para um smartphone ou smartwatch.

Apesar do avanço dos CGMs, os glicosímetros tradicionais ainda são os mais utilizados devido ao menor custo e maior acessibilidade. No entanto, como esses dispositivos normalmente não possuem conectividade digital, o registro dos valores ainda depende da anotação manual, o que pode gerar erros e dificuldades no acompanhamento médico.

No entanto, os métodos tradicionais de monitoramento glicêmico apresentam diversas limitações, como (FINNEGAN et al., 2019):

- Erro humano no registro manual dos valores;
- Dificuldade de acompanhar tendências glicêmicas ao longo do tempo;
- Risco de perda ou omissão de dados;
- Baixa acessibilidade de glicosímetros conectados a aplicativos móveis.

Para superar essas limitações, e aproximar a experiência de uso de dispositivos mais modernos e custosos, dos mais amplamente usados e simples, o uso de tecnologias de visão computacional e reconhecimento óptico de caracteres (OCR) surge como uma alternativa, permitindo a extração automática dos valores exibidos nos glicosímetros que não possuem características de conectividade (ASADI; GOUDARZI, 2023). Ou seja, é interessante remover ou reduzir, como mostrado na figura 1, processos manuais (setas vermelhas) que possam acarretar em erros de registro.

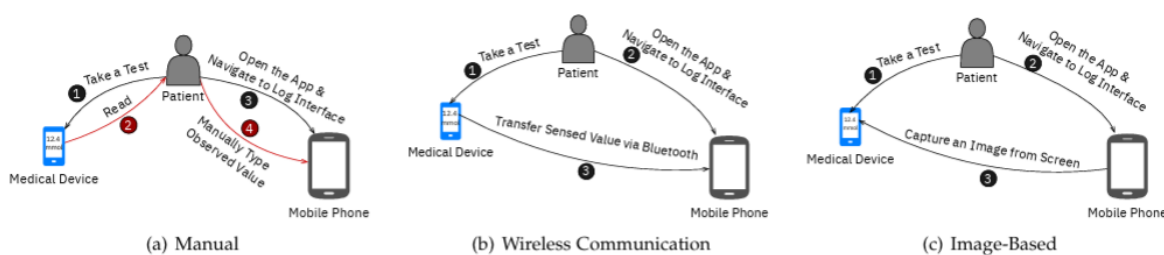


Figura 1: Métodos de monitoramento digitais. Intervenções humanas em vermelho. Fonte: Asadi; Goudarzi (2023).

2.2 VISÃO COMPUTACIONAL EM DISPOSITIVOS MÉDICOS

A visão computacional é uma área da inteligência artificial que permite a interpretação de imagens e vídeos por meio de técnicas de processamento de imagem e aprendizado de máquina. Uma das aplicações mais comuns dessa tecnologia na área da saúde é o Reconhecimento Óptico de Caracteres (OCR - Optical Character Recognition), que possibilita converter textos em imagens para formatos processáveis por computador (SMITH, 2007). Ela tem sido amplamente utilizado na área da saúde, não somente para dispositivos digitais mas também para diversas aplicações, incluindo:

- Detecção de dados de displays digitais em indústrias e laboratórios (GHOSH, S.; SHIT, S., 2014);
- Sistema de reconhecimento de registros médicos (LIU, H. et al., 2022);
- Classificação automática de documentos médicos escaneados (GOODRUM, H.; ROBERTS, K.; BERNSTAM, E. V., 2020);
- Identificação de valores em dispositivos médicos, como monitores de pressão arterial e glicosímetros (FINNEGAN et al., 2019);
- Otimização de gerenciamento de estoque farmacêutico (KUMAR ANANNYA CHULI, S., 2023);
- Reconhecimento de embalagens de remédios (LAKSHMIPATHY, A. et al., 2024);
- Extração em tempo real de informações de monitores médicos em salas de cirurgia (ZOU, Y. et al., 2024).

2.3 OCR X YOLO

Tradicionalmente, o reconhecimento de caracteres em visores de dispositivos médicos, como glicosímetros, foi realizado com base em técnicas de processamento de imagem clássicas. Essas técnicas (Figura 2) geralmente envolvem etapas de binarização, thresholding adaptativo, segmentação e uso de tabelas de lookup ou classificadores simples, como MLPs (Multi-Layer Perceptrons), para reconhecimento dos dígitos, com diversas limitações, especialmente em condições de variação de iluminação, reflexos, ângulos de captação e interferência de outros símbolos, sendo pouco robustas em cenários do mundo real. Embora métodos tradicionais sejam eficientes em ambientes controlados, “possuem baixa generalização frente à variabilidade de displays, ruídos e condições de captura” (FERREIRA, P. et al., 2025).

Além disso, um desafio recorrente está relacionado à leitura de dígitos de 7 segmentos, cujo design simplificado causa ambiguidade em dígitos visualmente semelhantes (como “2” e “5”) e pode comprometer o desempenho de algoritmos de OCR convencionais. A construção dos caracteres em 7 segmentos também acaba demandando um esforço de pré-processamento para “unir” os espaços entre eles.

A figura 2 mostra, em a, uma arquitetura simplificada normalmente usada, em um exemplo de detecção de um termômetro, que requer, além das etapas habituais de augmentation, diversas etapas de pré-processamento, como recortar a tela do dispositivo, e pós-processamento, como usar filtros e manipulação de imagem para remover elementos indesejados e tornar os dígitos mais “reconhecíveis”, para daí então partir para a detecção e classificação. Nota-se (no destaque pontilhado em vermelho) que cada etapa necessita de um nível mais alto de atenção e trabalho, muitas vezes fazendo uso de modelos e redes neurais distintas em cada uma dessas etapas. Em comparação (notar o pontilhado em vermelho novamente), a arquitetura proposta (em b), agora em um exemplo de detecção num glicosímetro, incorpora esses processos de forma mais fácil e direta.

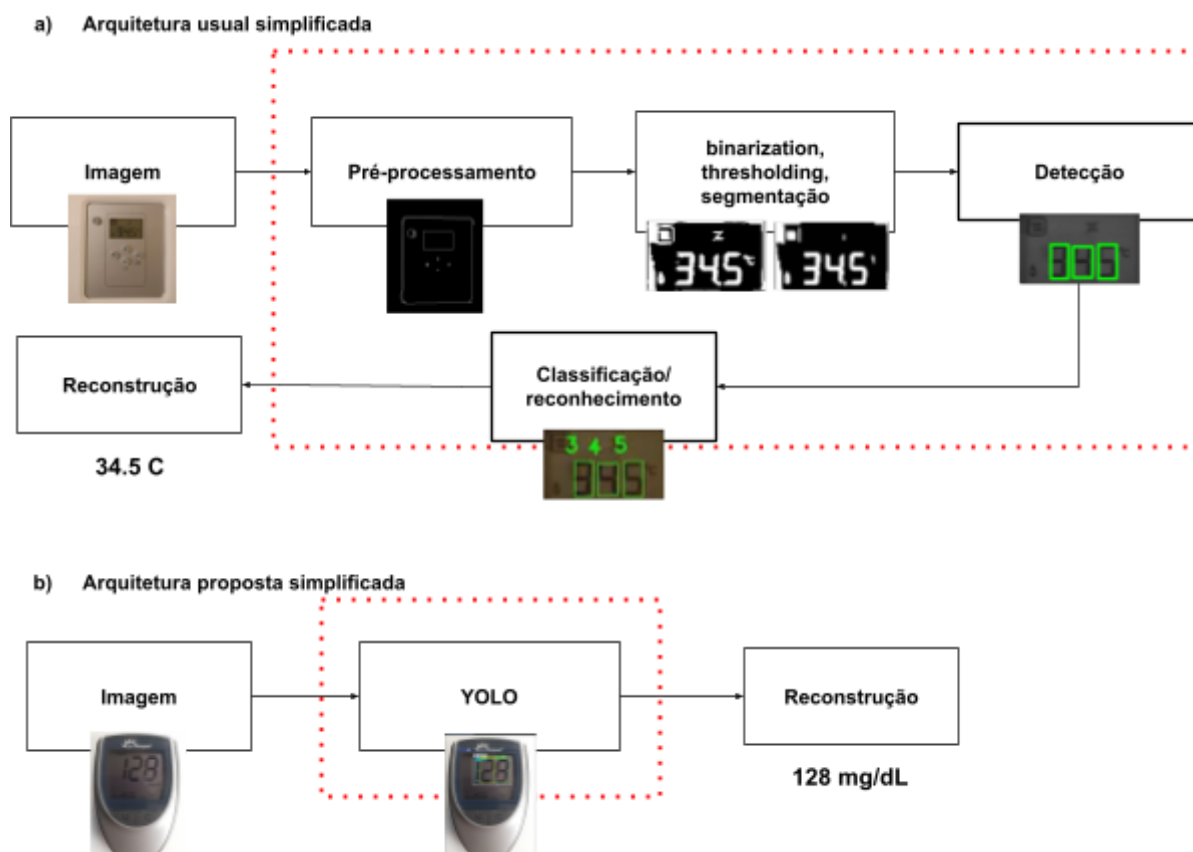


Figura 2: Arquitetura usual e proposta, respectivamente, simplificadas. Fonte: O autor (2025).

2.4 YOLO, UM DETECTOR SINGLE-STAGE

A detecção de dígitos em visores de glicosímetros exige rapidez e precisão, especialmente em soluções voltadas para dispositivos móveis. Neste contexto, o modelo adotado neste trabalho foi o YOLOv8 (You Only Look Once, versão 8), uma arquitetura moderna e eficiente de detecção de objetos em tempo real, baseada em aprendizado profundo. Seu principal diferencial está na abordagem de detecção em estágio único (single-stage), na qual as tarefas de localização dos objetos (caixas delimitadoras) e de classificação são executadas de forma simultânea e direta, sem etapas intermediárias.

Para contextualizar essa escolha, é importante compreender o funcionamento dos detectores em dois estágios (two-stage), representados na Figura 3 pela arquitetura do Faster R-CNN. Nesse tipo de modelo, uma Rede de Propostas Regionais (RPN) primeiro identifica regiões de interesse com possível presença de objetos, e só então uma segunda rede realiza a regressão das caixas e a classificação das classes correspondentes. Essa abordagem favorece a precisão em cenários mais

complexos, mas introduz maior latência e demanda computacional, tornando-se menos adequada para aplicações em tempo real ou em dispositivos móveis.

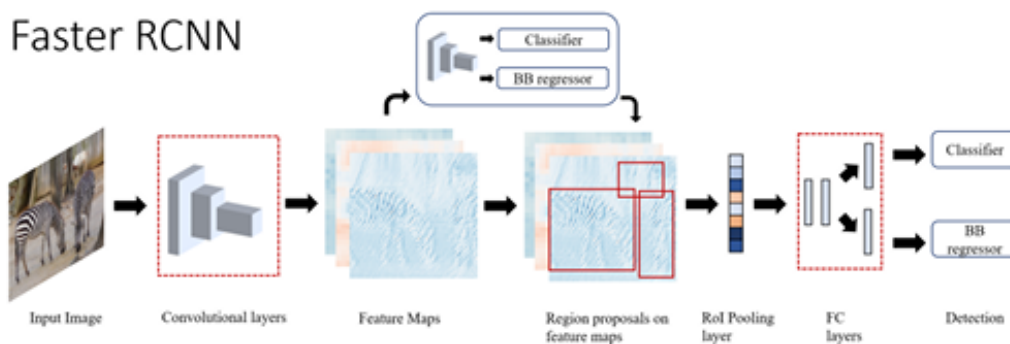


Figura 3: Ilustração da arquitetura interna de um exemplo de detector de objetos two stage (Faster RCNN). Fonte: Zaidi, S. S. A. et al. (2021).

Por outro lado, modelos como o YOLO, ilustrado na Figura 4, realizam a detecção diretamente sobre a imagem em uma única passada pela rede, combinando a predição das caixas com a classificação das classes. Isso confere ao modelo mais velocidade e simplicidade de implementação, sem a necessidade de componentes intermediários ou múltiplos estágios de processamento.

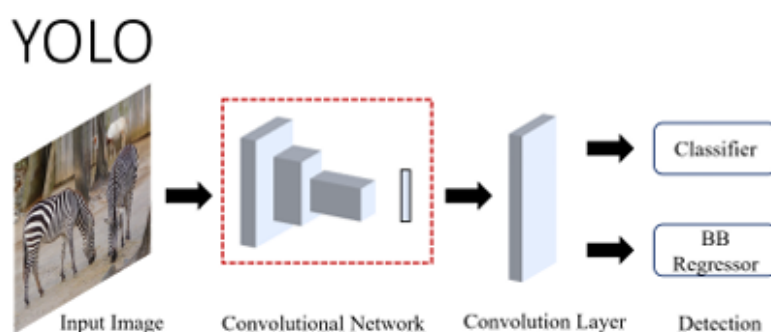


Figura 4: Ilustração da arquitetura interna de um exemplo de detector de objetos single-stage (YOLO). Fonte: Zaidi, S. S. A. et al. (2021).

Segundo Zaidi et al. (2021), essa integração direta entre regressão e classificação torna os detectores single-stage, como o YOLO, substancialmente mais rápidos que seus equivalentes two-stage. Embora esses últimos historicamente tenham oferecido maior acurácia em alguns contextos, os avanços recentes nas versões do YOLO vêm reduzindo essa diferença, consolidando-o como uma alternativa robusta

e eficaz para cenários diversos, incluindo a leitura de displays de sete segmentos em dispositivos médicos.

Essa decisão se baseia também em evidências práticas da literatura, como o estudo de Lobo et al. (2023), que avalia diferentes configurações de detecção para dispositivos médicos. Embora abordagens híbridas, que combinam detecção com segmentação ou classificação especializada, possam apresentar ganhos marginais de desempenho, sua complexidade eleva os custos computacionais e torna mais difícil a implementação em aplicações móveis. Já o YOLOv8 tem se mostrado eficaz mesmo sob variações de iluminação, ruído e ângulo, tornando possível uma solução mais simples, rápida e escalável, especialmente útil para a leitura automatizada de dígitos segmentados em glicosímetros.

Além disso, a aplicação de uma heurística baseada na ordenação horizontal das detecções permite reconstruir de forma confiável os valores glicêmicos completos, dispensando redes adicionais e mantendo a leveza do sistema, o que favorece sua futura integração em dispositivos móveis com capacidade computacional limitada.

2.5 TRABALHOS RELACIONADOS

Diversas abordagens têm sido propostas na literatura para automatizar a leitura de visores com dígitos de sete segmentos em dispositivos médicos, especialmente glicosímetros, monitores de pressão e oxímetros. Os métodos mais tradicionais, como o desenvolvido por Finnegan et al. (2019), consistem em pipelines com múltiplas etapas: a imagem capturada por smartphones é aprimorada por meio de pré-processamento, e os segmentos dos dígitos são identificados usando regiões extremas estáveis (MSER). Após isso, as regiões são agrupadas para formar dígitos inteiros, que por fim são classificados usando uma MLP treinada com vetores HOG. Essa abordagem demonstrou robustez, alcançando 93% de acurácia em testes com diferentes dispositivos, mas exige uma sequência rígida de etapas e pode ser sensível a erros em fases intermediárias.

Visando maior flexibilidade, Charles, J. et al. (2020) propuseram um sistema composto por dois módulos baseados em redes neurais convolucionais. O primeiro módulo realiza a detecção da tela e aplica correção de perspectiva, enquanto o segundo identifica os caracteres exibidos utilizando uma CNN leve (LeDigit). A proposta mostrou-se eficiente mesmo com treinamento feito apenas a partir de

dados sintéticos, oferecendo resultados promissores em tempo real em dispositivos móveis.

Com a evolução dos detectores de objetos baseados em deep learning, alguns estudos passaram a avaliar diretamente modelos como YOLO e SSD para essa tarefa. Ferreira et al. (2025) conduziram um estudo comparativo entre várias versões dos modelos YOLO (v3u, v5mu, v8n, v8m, v8l) e SSD MobileNet V2 FPNLite, utilizando imagens reais de displays médicos. O modelo YOLOv8l (large) obteve o melhor desempenho geral, demonstrando maior robustez frente a condições adversas como reflexos e distorções. A pesquisa também testou uma abordagem híbrida com CNNs para refinar a classificação, mas concluiu que o uso direto de detectores como o YOLO se mostrou mais eficaz e simples em termos de arquitetura.

Seguindo essa linha, Lobo et al. (2023) propuseram uma estrutura integrada para escaneamento e leitura de displays, combinando detecção por YOLOv3 e SSD com uma heurística de agrupamento baseada em DBSCAN. Embora o YOLOv3 tenha obtido melhores métricas em recall e F1-score, os autores optaram pelo SSD em função de sua leveza e compatibilidade com aplicações móveis. O estudo reforça a aplicabilidade de modelos de detecção de objetos como alternativa eficiente ao OCR tradicional.

Além desses, Low et al. (2024) exploraram o uso de bibliotecas modernas de OCR como PaddleOCR e MMOCR, testando modelos como DBNet, PANet, CRNN e PARSeq sobre dados com alto nível de ruído e reflexos. Os resultados indicaram que, embora o DBNet tenha bom desempenho na detecção (F1-score de 80%), a etapa de reconhecimento ainda enfrenta limitações — o melhor resultado foi do PARSeq, com 56,97% de acurácia após ajustes.

Por fim, Moreira (2022) propôs o uso de EfficientDet e EfficientDet-Lite para leitura direta de dígitos em visores médicos. Os modelos foram ajustados a partir de pesos pré-treinados no MS-COCO e avaliados em imagens reais com diferentes condições de captura. A variante EfficientDet-Lite1 obteve desempenho excepcional, com F1-score de 0,997 e total compatibilidade com execução em smartphones, reforçando a viabilidade de soluções baseadas em detecção direta, mesmo em dispositivos com baixo poder computacional.

3 METODOLOGIA

Este capítulo discorre sobre o ambiente e especificações do treinamento e validação propostos, bem como uma análise sobre o conjunto de dados, como ele foi obtido e manipulado para os treinamentos. Os passos e comandos utilizados para o treinamento do modelo de detecção YOLOv8 também são declarados. Além disso, o script de reconstrução dos valores presentes nas telas, através de uma heurística posicional, é mostrado.

3.1 AMBIENTE

Os experimentos deste trabalho foram realizados em ambiente local (numa máquina com cpu de 6 núcleos Ryzen 5 2600 3.4 GHz, 16 GB ddr4 2400 MHz, GPU Nvidia GeForce RTX3060 12 GB de VRAM), usando Jupyter Notebook, um ambiente de notebooks que oferece suporte a execução de código em Python, fazendo uso do poder de processamento disponível e possibilitando a geração de reports e análises reprodutíveis e disponíveis em outros ambientes. A versão da biblioteca usada para o modelo foi a Ultralytics YOLOv8, instalada via `pip install ultralytics` num ambiente virtual (venv) criado para o trabalho.

3.2 CONJUNTO DE DADOS

Os dados utilizados no estudo foram obtidos a partir do repositório público Glycometre¹ hospedado no Roboflow Universe, e posteriormente duplicado em um fork específico para este projeto: Glycometre (glucoocr). O dataset é composto por 7449 imagens, em maior parte de glicosímetros, capturadas por dispositivos móveis em cenários reais (Figura 5), no formato .jpg, com resolução de 640x640 pixels, anotadas com caixas delimitadoras (bounding boxes), em arquivos de texto (txt), no formato compatível com o modelo YOLO. Cada linha dos arquivos de texto contém a classe e as coordenadas da sua respectiva bounding box.

A base foi dividida automaticamente pelo Roboflow nos seguintes subconjuntos:

- Treinamento (train): 70% (5219 imagens)
- Validação (val): 20% (1508 imagens)
- Teste (test): 10% (722 imagens)

¹ Link do repositório do Roboflow: <https://app.roboflow.com/glucoocr/glycometre-fi2xq/2>

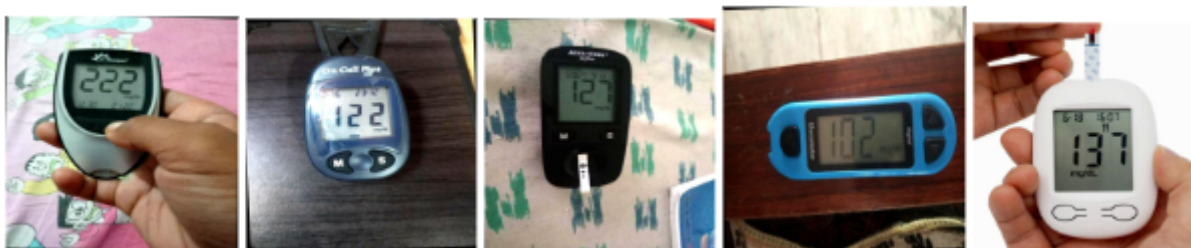


Figura 5: Exemplos de imagens presentes no dataset. Fonte: o autor (2025).

Não foram aplicadas, inicialmente, etapas de pré-processamento ou aumentos artificiais na base (augmentation) original, para que fossem preservadas as características originais das imagens reais capturadas de diversos modelos de glicosímetros e dispositivos médicos similares (oxímetros, medidores de pressão arterial e de termômetros). No entanto, na tentativa de melhorar os resultados e contornar problemas percebidos na validação (Figura 6), em relação a falta de imagens em condições mais adversas, (com inclinações, inversões e mudanças de direção).



Figura 6: Exemplo de predição sem e com o augmentation na base de treinamento, respectivamente. Fonte: O autor (2025).

Foi criada então uma nova versão, dentro do próprio Roboflow, do dataset, com o uso da ferramenta automática de augmentation da plataforma, que acrescentou imagens ao subconjunto de treinamento da base (Figura 7), que passou a ter 17850 imagens ao todo, com 15620 imagens no subconjunto de treinamento.



Figura 7: Exemplos de imagens com augmentation aplicado. Fonte: o autor (2025).

A base então passou a ter os seguintes subconjuntos:

- Treinamento (train): 88% (15620 imagens)
- Validação (val): 8% (1508 imagens)
- Teste (test): 4% (722 imagens)

Cada imagem contém anotações pertencentes a uma ou mais das seguintes classes:

Tabela 1: Classes presentes no dataset. Fonte: O autor (2025).

Faixa	Classe	Descrição
0–9	Dígitos	Dígitos individuais (display de 7 segmentos)
10	Caixa do número	Box que contorna todo o valor glicêmico exibido
12	Tela do glicosímetro	Box que contorna a tela do aparelho
11, 14	Telas de outros dispositivos	Ex: medidores de pressão arterial e termômetros
15	Pontos e símbolos auxiliares	Ex: vírgulas, separadores e unidades como “mmol/L”

Durante o treinamento, através do framework da Ultralytics e da biblioteca importada do YOLO, gera uma representação gráfica da distribuição das labels, com sua contagem, posição das bounding boxes (caixas delimitadoras de cada classe), coordenadas (x, y) e altura e comprimento das classes:

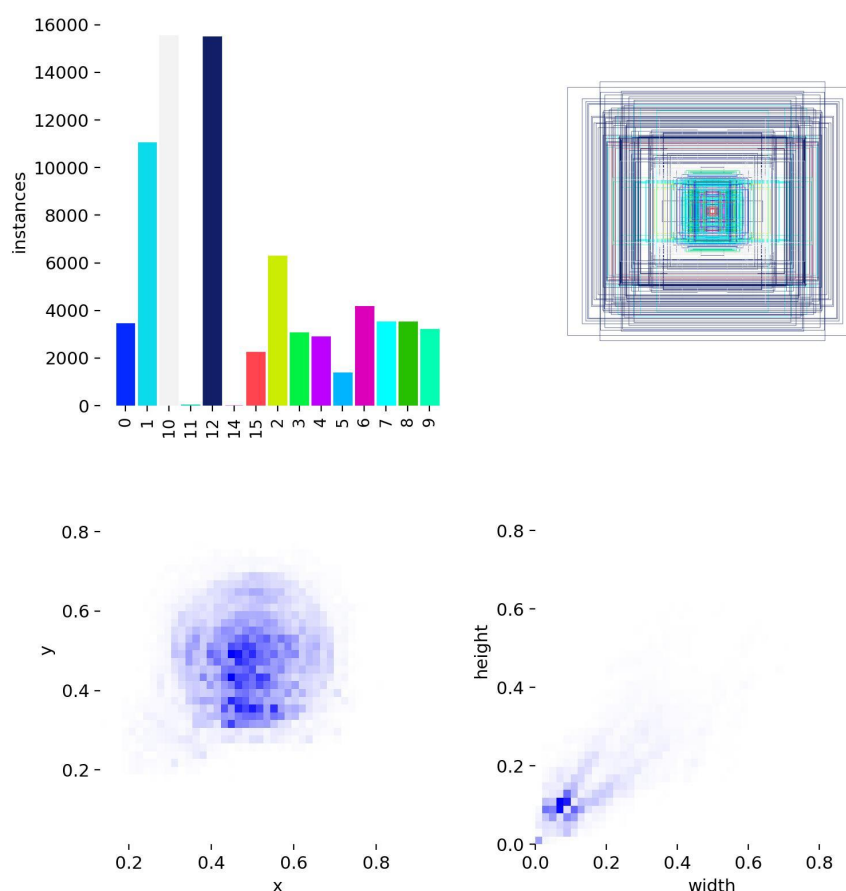


Figura 8: Contagem, posicionamento, distribuição das coordenadas e das dimensões das labels (anotações/boxes), respectivamente. Fonte: O autor (2025).

As principais características da configuração foram:

- Modelo: YOLOv8n (nano), YOLOv8m (medium), YOLOv8l (large), YOLOv8x (extra large), pré treinados em COCO, com o tamanho variando de acordo com a versão
- Tarefa: detecção (detect)
- Tamanho da imagem: 640x640
- Épocas de treinamento: 50
- Batch size: padrão (adaptado ao ambiente do jupyter notebook)
- Dataset: fornecido via link direto do Roboflow, com um args.yaml, indicando os hiperparâmetros:

```
from roboflow import Roboflow
rf = Roboflow(api_key="[REDACTED]")
project = rf.workspace("glucoocr").project("glycometre-fi2xq")
version = project.version(2)
dataset = version.download("yolov8")
```

- Comando de treinamento executado:

```
from ultralytics import YOLO
model = YOLO("yolov8m.pt") #o nome do modelo muda pra cada versão
model.train(data=f"glycometre-2/data.yaml", device=0, epochs=50,
imgsz=640)
```

Durante o treinamento, o próprio framework realiza avaliações periódicas sobre o conjunto de validação, calculando métricas como precisão, recall, F1-score e mAP (mean average precision). Os modelos intermediários foram salvos automaticamente, sendo selecionado o arquivo `best.pt` com base no melhor desempenho observado. Através dos pesos salvos no `best.pt` é possível carregar o modelo para fazer as predições.

3.3 AGRUPAMENTO POSICIONAL E REGISTRO TEMPORAL DE VALORES

Embora o modelo YOLOv8 seja capaz de detectar múltiplos dígitos simultaneamente em uma imagem, o valor glicêmico completo precisa ser reconstruído a partir dessas detecções. Para isso, foi implementada uma heurística de pós-processamento baseada no agrupamento posicional dos dígitos, ordenando-os da esquerda para a direita, conforme sua localização horizontal na imagem.

A heurística proposta realiza as seguintes etapas:

- Filtragem das detecções: são consideradas apenas as predições correspondentes aos dígitos de 0 a 9, descartando classes auxiliares como a classe 10 (bounding box do número completo) e a classe 12 (bounding box da tela), que foram mantidas apenas para visualização e não apresentaram desempenho confiável.
- Cálculo da posição horizontal: para cada bounding box correspondente a um dígito, calcula-se a coordenada média do eixo x (centro horizontal da caixa).
- Ordenação sequencial: os dígitos detectados são ordenados em ordem crescente de posição horizontal, assumindo a estrutura usual dos displays numéricos de sete segmentos.
- Concatenação final: os dígitos ordenados são unidos para formar a sequência numérica completa, representando o valor glicêmico aferido.

A robustez da abordagem foi verificada em imagens com variações de ângulo, inclinação e iluminação, graças ao uso de data augmentation durante o treinamento (incluindo rotações, espelhamentos e ruídos). Essa estratégia eliminou a

necessidade de segmentações adicionais, facilitando o uso do sistema em dispositivos com menor poder computacional.

Complementarmente, o pipeline foi expandido para incluir a extração da data e hora da captura da imagem, com base nos metadados Exif. Essa informação é registrada junto ao valor numérico final em um arquivo de texto tabular, possibilitando rastreamento cronológico para uso pessoal ou clínico. A combinação dessas etapas compõe uma solução prática, leve e eficaz para a digitalização automatizada de leituras glicêmicas.

Abaixo, apresenta-se o script Python correspondente à implementação dessa heurística:

```
from ultralytics import YOLO
from PIL import Image
import piexif
import os
import datetime

def get_datetime_from_exif(image_path):
    try:
        exif_dict = piexif.load(image_path)
        dt_str = exif_dict["0th"].get(piexif.ImageIFD.DateTime)
        if dt_str:
            return datetime.datetime.strptime(dt_str.decode(),
"%Y:%m:%d %H:%M:%S")
    except Exception:
        pass
    return datetime.datetime.now()

def aplicar_heuristica_e_registrar(model_path, pasta_imagens,
saida_txt):
    model = YOLO(model_path)
    with open(saida_txt, 'a') as f:
        for nome_arquivo in sorted(os.listdir(pasta_imagens)):
            if not nome_arquivo.lower().endswith((".jpg", ".jpeg",
".png")):
                continue
            caminho_img = os.path.join(pasta_imagens, nome_arquivo)
            resultados = model.predict(caminho_img, conf=0.5)[0]
            caixas = resultados.boxes
            if caixas is None or len(caixas) == 0:
                continue

            digitos = []
            for i in range(len(caixas.cls)):
                classe = int(caixas.cls[i].item())
                if classe in range(10): # classes de 0 a 9
                    x1, y1, x2, y2 = caixas.xyxy[i].tolist()
                    x_centro = (x1 + x2) / 2
```

```

        digitos.append((x_centro, classe))

    if not digitos:
        continue

    digitos_ordenados = sorted(digitos, key=lambda x: x[0])
    valor_final = ''.join(str(c[1]) for c in
digitos_ordenados)

    data_hora =
get_datetime_from_exif(caminho_img).strftime("%Y-%m-%d %H:%M:%S")

    print(f"{data_hora} → {valor_final}")
    f.write(f"{data_hora}\t{valor_final}\n")

```

Essa estratégia representa um dos diferenciais da arquitetura proposta neste trabalho, oferecendo uma alternativa funcional e simplificada frente a soluções que demandam pipelines mais complexas ou dependem de OCR tradicional.

4 RESULTADOS E DISCUSSÃO

4.1 AVALIAÇÃO DOS RESULTADOS

A avaliação do desempenho do modelo foi realizada com base nas métricas geradas automaticamente pela ferramenta da Ultralytics após cada época de treinamento, além da execução posterior da validação no conjunto de teste com o comando (ULTRALYTICS, 2023):

```
model.val(data=f"glycometre-2/data.yaml", split="test")
```

As principais métricas analisadas foram:

- Precisão (precision): proporção de detecções corretas sobre o total de predições
- Revocação (recall): proporção de objetos reais detectados corretamente
- mAP@0.5: média da precisão por classe com interseção sobre união (IoU) $\geq 50\%$
- mAP@0.5:0.95: média ponderada em múltiplos limiares de IoU
- Matrizes de confusão e curvas PR: usadas para análise qualitativa dos erros

Também foram geradas imagens ilustrativas com as predições feitas no conjunto de validação, possibilitando uma inspeção visual da qualidade das detecções realizadas. Os resultados de cada treinamento e validação, para cada uma das versões do modelo, estão disponíveis em repositório² do github, junto dos procedimentos necessários para que se realizem os procedimentos de configuração do ambiente, download do dataset, treinamento e validação dos modelos.

Para fins de comparação e verificação de desempenho e métricas, o treinamento foi feito em variações do YOLOv8 que contêm diferentes tamanhos de backbone pré-treinados. Inicialmente utilizou-se a versão YOLOv8n (nano, a mais leve) e, após a adição de dados via augmentation, novos treinamentos foram realizados com as versões nano, m, l e x, configuradas com hiperparâmetros padrão que exploram maior poder computacional, buscando otimizar a precisão do modelo. A tabela comparativa (Tabela 1) demonstra o impacto dessas diferenças no tempo de treinamento e no uso de VRAM (memória de vídeo) do modelo no trabalho.

² Link do repositório do Github: <https://github.com/pedroceciliocn/tcc-pccn>

Tabela 2 - Tabela comparativa das versões do YOLOv8. Fonte: O autor (2025).

Versão	Tempo de Treinamento (aproximado)	Uso de VRAM	Nº de Parâmetros (milhões)	Tamanho Aproximado do Modelo	Indicação de Uso (segundo documentação)
YOLOv8n (nano)	~1 hora	2 GB	~3.2 M	~6 MB	Dispositivos móveis e baixa capacidade
YOLOv8m (medium)	~6 horas	6 GB	~25 M	~50 MB	Equilíbrio entre leveza e precisão
YOLOv8l (large)	~12 horas	9 GB	~43 M	~85 MB	Alta precisão com maior custo computacional
YOLOv8x (extra large)	~48 horas	12 GB	~68 M	~130 MB	Máxima acurácia, ideal para GPUs robustas

As diferentes versões do modelo YOLOv8 foram treinadas por 50 épocas sobre o conjunto de dados “Glycometre”. Ao fim da 50ª época, o modelo apresentou os seguintes valores consolidados, para cada uma das versões (Tabela 3), foi destacado em negrito o melhor valor para cada métrica:

Tabela 3 - Métricas de desempenho na validação (época 50). Fonte: O autor (2025).

Métrica	v8n (sem aug)	v8n	v8m	v8l	v8x
Precisão (precision)	93,33%	92,85%	94,10%	94,38%	94,24%
Revocação (recall)	93,19%	92,25%	93,41%	92,70%	93,25%
mAP@0.5	94,76%	94,16%	94,98%	94,89%	94,88%
mAP@0.5:0.95	78,22%	77,88%	79,87%	80,34%	80,71%

Com base nessas métricas a análise se concentrará nos resultados do modelo YOLOv8m (medium), que apresentou o melhor equilíbrio entre leveza e desempenho de detecção. Os resultados das outras versões estão disponíveis no repositório³. Todas as métricas, para cada uma das épocas, foram armazenadas no arquivo `results.csv`, também gerado automaticamente ao executar treinamento e validação. A figura 9 mostra a curva de perda durante o treinamento e validação,

³ Link do repositório do Github: <https://github.com/pedroceciliocn/tcc-pccn>

com quedas consistentes nos valores de `box_loss`, `cls_loss` e `dfl_loss`, o que indica boa convergência do modelo. Ao mesmo tempo, as métricas de precisão, recall e mAP evoluem positivamente ao longo das épocas, evidenciando aprendizado progressivo.

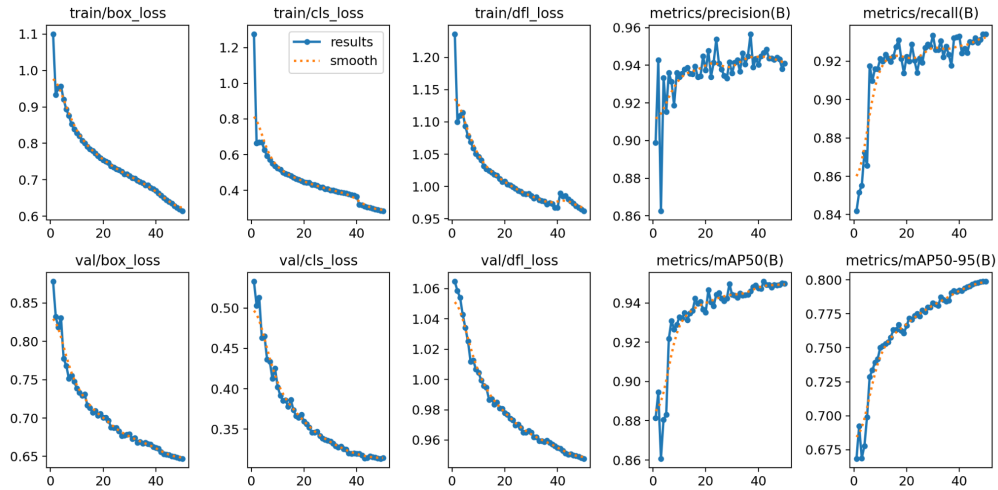


Figura 9: Curvas de perda e métricas de validação durante o treinamento. Fonte: O autor (2025).

A matriz de confusão apresentada na Figura 10 evidencia a acurácia do modelo na classificação correta dos dígitos (classes 0 a 9) e dos elementos auxiliares do visor (classes 10 a 15). A maior parte das previsões encontra-se concentrada na diagonal, demonstrando que o modelo está acertando a classe esperada na maioria dos casos. Pequenas confusões ocorrem entre classes de aparência visual próxima, como as classes 1 e 7 ou 2 e 5, o que é esperado dado o uso de visores de sete segmentos.

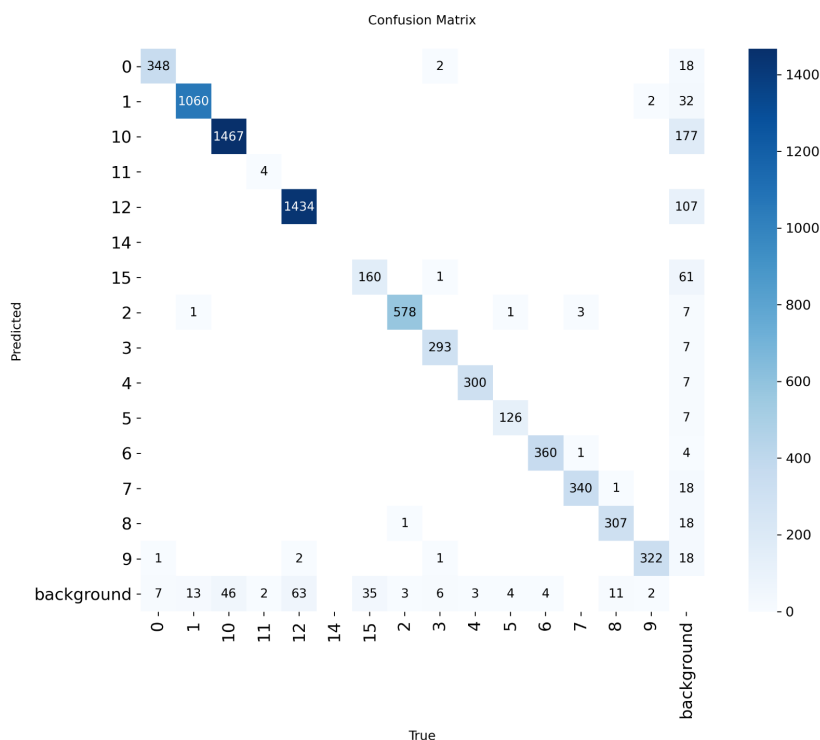


Figura 10: Matriz de confusão. Fonte: O autor (2025).

Já a curva de recall por classe em função do limiar de confiança (Figura 11) mostra a capacidade do modelo de detectar verdadeiramente as ocorrências das classes em diferentes níveis de exigência de certeza. O alto recall nas principais classes (dígitos 0–9) indica que o modelo raramente deixa de identificar um dígito presente na imagem, mesmo com limiares mais altos. Isso é particularmente importante em aplicações médicas, onde a omissão de um dígito pode gerar interpretações incorretas dos valores glicêmicos. A maioria das classes atinge recall superiores a 95% em limiares moderados (0.5 ~ 0.6), o que demonstra robustez do modelo. Uma exceção perceptível é a classe 15 (usada no dataset para pontos em glicosímetros e aparelhos que usam casas decimais), que apresenta desempenho inferior, o que pode estar associado à baixa frequência dessa classe. Outra é a classe 11, que também tem desempenho muito provavelmente baixo por conta da sua baixa representação.

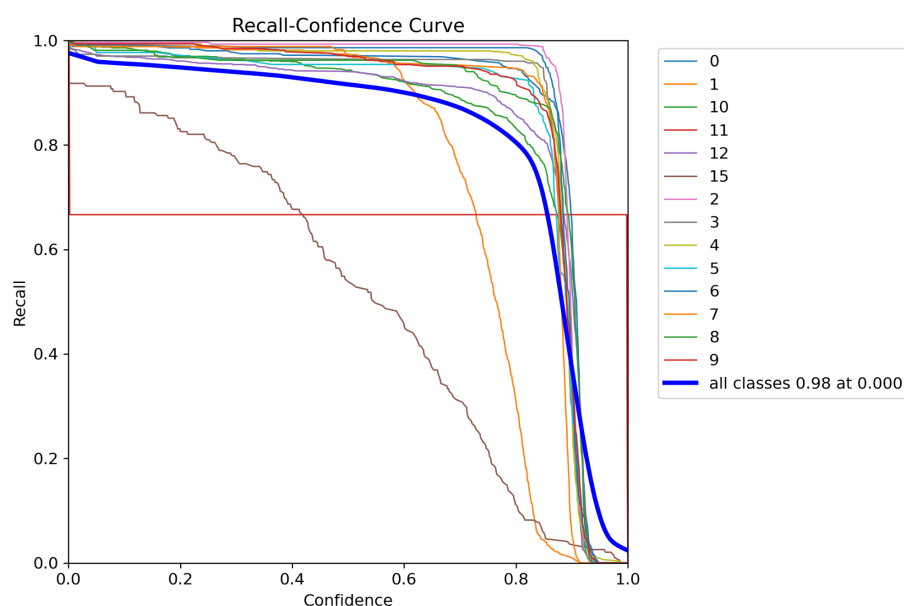


Figura 11: Curva recall x confiança. Fonte: O autor (2025).

A curva de F1-score por classe (Figura 12) reflete o equilíbrio entre precisão e revocação. Essa métrica é útil para avaliar o desempenho do modelo em cenários onde tanto falsos positivos quanto falsos negativos devem ser minimizados. As curvas de dígitos mais frequentes (como 1, 0, 2 e 6) se mantêm acima de 0.9, sugerindo que o modelo é confiável nessas categorias. Mais uma vez as classes 11 e 15 apresentam desempenho inferior.

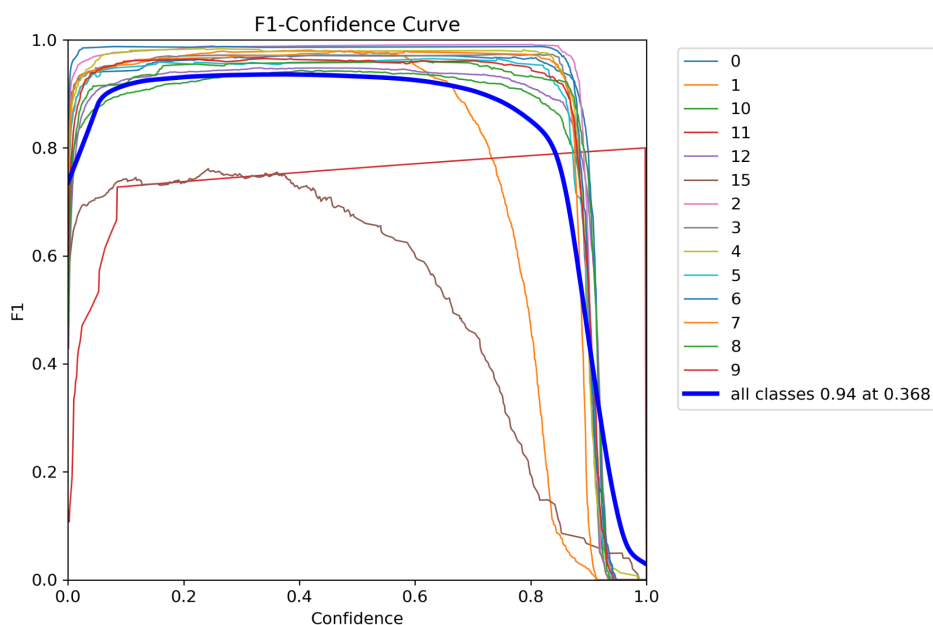


Figura 12: F1-score x confiança. Fonte: O autor (2025).

A curva de precisão por classe (Figura 13) revela o quanto o modelo é capaz de evitar falsos positivos, ou seja, o quão confiável é uma detecção quando ela de fato ocorre. O modelo alcança precisão superior a 95% para quase todas as classes com confiança acima de 0.6, sendo a classe 6 um dos maiores destaques, chegando a 99,3%. Isso significa que, ao prever essa classe, o modelo está normalmente correto.

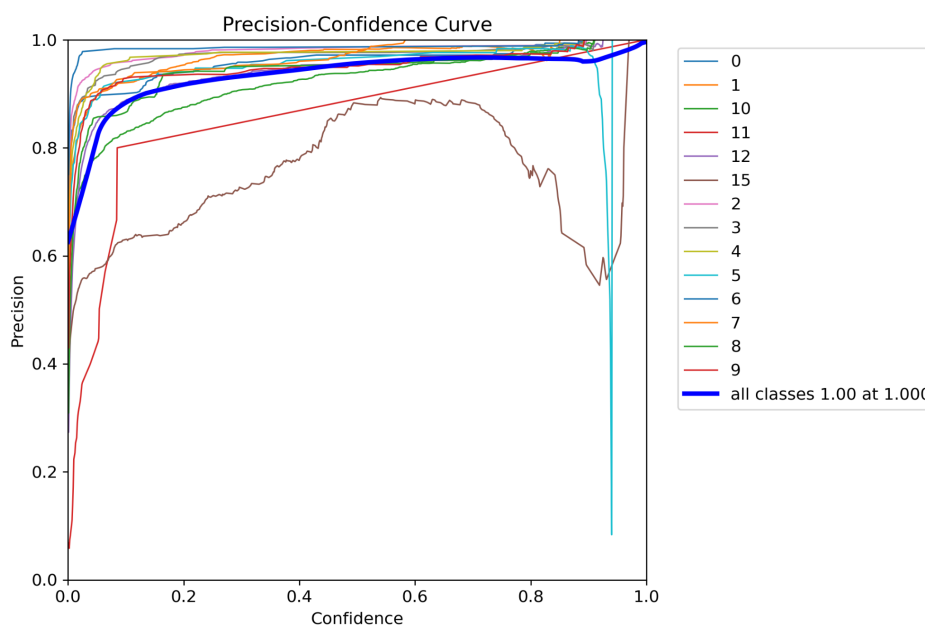


Figura 13: Curva precisão x confiança. Fonte: O autor (2025).

Na curva de Precision-Recall (Figura 14), é possível visualizar de forma mais abrangente o desempenho do modelo para cada classe, considerando a troca entre precisão e revocação em diferentes pontos de decisão. O modelo atinge mAP@0.5 de 0.950, o que reforça seu bom desempenho geral. A curva geral (linha azul) mantém níveis altos de equilíbrio entre precisão e revocação, destacando a confiabilidade significativa da abordagem usando YOLOv8m (medium).

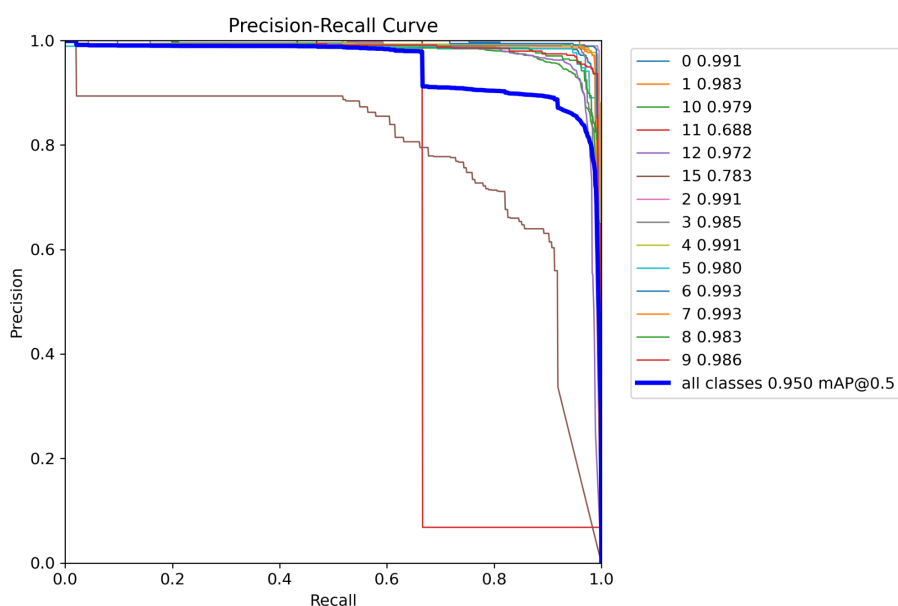


Figura 14: Curva de precisão x revocção por classe (mAP@0.5 global = 0.950). Fonte: O autor (2025).

A performance do modelo também pode ser observada nas imagens de validação com predições (Figuras 15 e 16).



Figura 15: Exemplos visuais de detecções. Fonte: O autor (2025).

Nelas, é possível notar que o YOLOv8m (medium) foi capaz de detectar corretamente os dígitos principais exibidos nas telas dos glicosímetros, mesmo em condições variadas de iluminação, ângulo, reflexos e presença de mãos ou outros

objetos no fundo. A sobreposição dos bounding boxes evidencia a capacidade do modelo de separar com precisão cada dígito individual, o que é essencial para a reconstrução correta do valor glicêmico por meio da heurística posicional.



Figura 16: Exemplos visuais de detecções em configurações mais diversas. Fonte: O autor (2025).

Esses resultados sugerem que há viabilidade do uso do YOLOv8m (medium) para a tarefa de leitura automática de telas de glicosímetros. A acurácia das detecções, aliada à sua aparente robustez frente à variabilidade visual do dataset, reforça a adequação do modelo para futuras aplicações móveis ou integração com sistemas de prontuário eletrônico. A escolha do modelo m representa um bom equilíbrio entre precisão e eficiência computacional, sendo mais leve que as versões l (large) e x (extra large), mas entregando desempenho similar ou superior em algumas métricas.

5 CONCLUSÃO

Neste trabalho, foi proposta, desenvolvida e avaliada uma abordagem baseada em aprendizado profundo para a leitura automatizada de valores glicêmicos exibidos em glicosímetros com displays de sete segmentos, utilizando um modelo da família YOLOv8. A motivação central surgiu da necessidade recorrente de soluções que minimizem o esforço de entrada manual de dados por pacientes com diabetes, aumentando a precisão e a praticidade no registro de medições de glicemia, sobretudo em contextos nos quais os dispositivos de medição não possuem conectividade digital nativa.

A partir de um conjunto de dados, contendo inicialmente mais de 7 mil imagens rotuladas, foi possível treinar e validar diferentes variações do modelo YOLOv8 (nano, medium, large e x-large), destacando-se o YOLOv8m (medium) como o que apresentou o melhor equilíbrio entre desempenho e custo computacional. Os resultados obtidos demonstraram métricas robustas, com valores superiores a 94% de precisão e revocação, além de um mAP@0.5 de 94,98% e mAP@0.5:0.95 de 79,87%, indicando a efetividade do modelo na tarefa de detecção de múltiplos dígitos em condições reais de captura de imagem.

A adoção de uma heurística de agrupamento posicional baseada na ordenação horizontal das detecções permitiu a reconstrução confiável dos valores exibidos na tela dos dispositivos, dispensando etapas complexas de pré-processamento, segmentação ou classificação posterior. Essa simplificação arquitetural representa uma alternativa em relação às abordagens tradicionais baseadas em OCR e reforça a viabilidade de implantação dessa solução em dispositivos móveis e sistemas de saúde de baixo custo, com potencial aplicação em contextos de mHealth e telemonitoramento.

Do ponto de vista técnico, o estudo contribui para o campo de visão computacional aplicada à saúde, ao demonstrar que modelos de detecção de objetos modernos, como o YOLOv8, são eficazes não apenas na identificação de objetos genéricos, mas também em tarefas especializadas como a leitura de displays segmentados. A proposta aqui documentada poderá servir como base para o desenvolvimento de aplicações móveis assistivas ou como subsídio para plataformas de monitoramento clínico integradas a prontuários eletrônicos.

Como limitações, no que tange a proposta do trabalho, destacam-se a dependência da qualidade da imagem capturada (reflexos, ângulo e foco ainda impactam o desempenho), a generalização da solução proposta no que tange a dispositivos com displays de sete segmentos, a falta da detecção de informações agregadas aos índices glicêmicos e a ausência de testes em ambientes clínicos com usuários finais. Já quanto ao estudo dos impactos do que foi proposto, pontua-se a falta de levantamentos e literatura sobre o uso de ferramentas similares no contexto brasileiro, e também uma avaliação quantitativa sobre modelos e tipos de glicosímetros tanto no mercado privado quanto no SUS, a partir de uma busca mais aprofundada em registros de ministério, secretarias, licitações e devidas documentações relacionadas a saúde no país.

Como trabalhos futuros, propõe-se: (i) a expansão da base de dados com variações de dispositivos e ambientes (levando em conta também o contexto local), bem como a implementação da detecção das informações que são agregadas aos índices glicêmicos (data, hora); (ii) a integração do modelo a um protótipo funcional de aplicativo; e (iii) a avaliação em cenários reais com pacientes e profissionais de saúde, considerando métricas de usabilidade, acurácia clínica e impacto na adesão ao tratamento.

REFERÊNCIAS

- ASADI, N.; GOUDARZI, M. An Ensemble Mobile-Cloud Computing Method for Affordable and Accurate Glucometer Readout. arXiv. Disponível em: <<http://arxiv.org/abs/2301.01758>>. Acesso em: 2 jun. 2025.
- BRASIL. Ministério da Saúde. Diabetes: complicações. [s.d.]a. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/diabetes/complicacoes/complicacoes>. Acesso em: 11 jul. 2025.
- BRASIL. Ministério da Saúde. Diabetes (diabetes mellitus). [s.d.]b. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/diabetes/diabetes>. Acesso em: 11 jul. 2025.
- BRASIL. Ministério da Saúde. Diabetes: tratamento. [s.d.]c. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/diabetes/tratamento/tratamento>. Acesso em: 11 jul. 2025.
- BONILLA-RIVAS, F. et al. Adapting YOLO to recognition of real numbers in 7-segment digits. Em: 2023 INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING, COMMUNICATION, NETWORKING AND SERVICES (ICCNS), 2023, , [s.l.] . Anais eletrônicos... [S.l.]: [s.n.], 2023. p. 19–25. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/10193592>>. Acesso em: 7 jul. 2025.
- CHARLES, J. et al. Real-time screen reading: reducing domain shift for one-shot learning. 2020.
- Computer vision app allows easier monitoring of diabetes | University of Cambridge. 13 nov. 2020. Disponível em: <<https://www.cam.ac.uk/research/news/computer-vision-app-allows-easier-monitoring-of-diabetes>>. Acesso em: 11 jul. 2025.
- DOCTOR WHO. The Witchfinders. Direção de Sallie Aprahamian. Roteiro de Joy Wilkinson e Chris Chibnall. Temporada 11, episódio 8. [S.l.]: BBC Studios, 2018. 1 episódio (46 min), son., color.
- DUGAS, M. et al. Engagement and Outcomes Associated with Contextual Annotation Features of a Digital Health Solution. Journal of Diabetes Science and Technology, 23 dez. 2020. v. 16, n. 4, p. 804–811.
- FERREIRA, P. et al. Digitization of Medical Device Displays Using Deep Learning Models: A Comparative Study. Applied Sciences, jan. 2025a. v. 15, n. 10, p. 5436.
- FINNEGAN, E. et al. Automated method for detecting and reading seven-segment digits from images of blood glucose metres and blood pressure monitors. Journal of Medical Engineering & Technology, 18 ago. 2019. v. 43, n. 6, p. 341–355.
- GHOSH, S.; SHIT, S. A low cost data acquisition system from digital display instruments employing image processing technique. Em: 2014 INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS (ICACCI), 2014, , [s.l.] . Anais eletrônicos... [S.l.]: [s.n.], 2014. p.

1065–1068. Disponível em:

<<https://ieeexplore.ieee.org/abstract/document/6968229>>. Acesso em: 5 dez. 2024.

GHUGARDARE, R. P. et al. Optical character recognition system for seven segment display images of measuring instruments. Em: TENCON 2009 - 2009 IEEE REGION 10 CONFERENCE, 2009, , [s.l.] . Anais eletrônicos... [S.l.]: [s.n.], 2009. p. 1–6.

Disponível em: <<https://ieeexplore.ieee.org/abstract/document/5395994>>. Acesso em: 5 dez. 2024.

GOODRUM, H.; ROBERTS, K.; BERNSTAM, E. V. Automatic classification of scanned electronic health record documents. International Journal of Medical Informatics, 1 dez. 2020. v. 144, p. 104302.

GROUP, S. M. Computer Vision App for Monitoring Diabetes. 1 mar. 2022.

Disponível em:

<<https://www.techbriefs.com/component/content/article/45378-computer-vision-app-for-monitoring-diabetes>>. Acesso em: 3 dez. 2024.

KIRWAN, M. et al. Diabetes Self-Management Smartphone Application for Adults With Type 1 Diabetes: Randomized Controlled Trial. Journal of Medical Internet Research, 13 nov. 2013a. v. 15, n. 11, p. e2588.

KUMAR ANANNYA CHULI, S. Optimizing Pharmaceutical Inventory Management with YoloV7 and Easy OCR on Medicine Strips. International Journal of Science and Research (IJSR), 5 ago. 2023. v. 12, n. 8, p. 1662–1669.

LAKSHMIPATHY, A. et al. MediCaption: Integrating YOLO-Driven Computer Vision and NLP for Advanced Pharmaceutical Package Recognition and Annotation. [s.d.]. Disponível em:

<<https://www.authorea.com/users/759909/articles/735799-medication-integrating-yolo-driven-computer-vision-and-nlp-for-advanced-pharmaceutical-package-recognition-and-annotation>>. Acesso em: 10 mar. 2025.

LIU, Chang. Digits Recognition on Medical Device. Electronic Thesis and Dissertation Repository, 20 abr. 2016. Disponível em: <<https://ir.lib.uwo.ca/etd/3683>>.

LIU, H. et al. DeepSSR: a deep learning system for structured recognition of text images from unstructured paper-based medical reports. Annals of Translational Medicine, jul. 2022. v. 10, n. 13, p. 740.

LOBO, P. et al. Smart scan of medical device displays to integrate with a mHealth application. Heliyon, 1 jun. 2023. v. 9, n. 6. Disponível em:

<[https://www.cell.com/heliyon/abstract/S2405-8440\(23\)03504-1](https://www.cell.com/heliyon/abstract/S2405-8440(23)03504-1)>. Acesso em: 7 jul. 2025.

LOW, L. M.; MOHD SALLEH, F. H.; et al. Detecting and recognizing seven segment digits using a deep learning approach. ITM Web of Conferences, 2024. v. 63, p. 01007.

MOREIRA, L. P. Automated Medical Device Display Reading Using Deep Learning Object Detection. arXiv. Disponível em: <<http://arxiv.org/abs/2210.01325>>. Acesso em: 2 jun. 2025.

OPTICAL CHARACTER RECOGNITION. Em: Wikipedia. [S.l.]: [s.n.], 2024.

Disponível em:

<https://en.wikipedia.org/w/index.php?title=Optical_character_recognition&oldid=1261566996>. Acesso em: 17 dez. 2024.

PRAKRUTHI, M. K.; KALYAN, V. V.; SUHAS, S.; KATTI, M. G.; PAI, S. N. Application of Convolutional Neural Networks in Mobile Devices for Inferring Readings from Medical Apparatus. International Journal of Engineering Research & Technology (IJERT), v. 6, n. 13, p. 49–52, 2018. Disponível em:

<https://rsisinternational.org/IJRSI/Issue41/49-52.pdf>. Acesso em: 2 jun. 2025.

Roboflow: Computer vision tools for developers and enterprises. [s.d.]. Disponível em: <<https://roboflow.com>>. Acesso em: 21 jul. 2025.

SMITH, R. An Overview of the Tesseract OCR Engine. Em: NINTH INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION (ICDAR 2007) VOL 2, 2007, Curitiba, Parana, Brazil. Anais eletrônicos... Curitiba, Parana, Brazil: IEEE, 2007. p. 629–633. Disponível em:

<<http://ieeexplore.ieee.org/document/4376991/>>. Acesso em: 11 jul. 2025.

The 31st British Machine Vision (Virtual) Conference 2020 : Real-time screen reading: reducing domain shift for one-shot learning. [s.d.]. Disponível em:

<https://bmvc2020-conference.com/conference/papers/paper_0512.html>. Acesso em: 3 dez. 2024.

ULTRALYTICS. Treinar dados personalizados. 12 nov. 2023. Disponível em:

<https://docs.ultralytics.com/pt/yolov5/tutorials/train_custom_data>. Acesso em: 25 fev. 2025.

YOLOv8 Architecture; Deep Dive into its Architecture -Yolov8. 15 jan. 2024.

Disponível em: <<https://yolov8.org/yolov8-architecture/>>. Acesso em: 3 jul. 2025.

ZAIDI, S. S. A. et al. A Survey of Modern Deep Learning based Object Detection Models. arXiv. Disponível em: <<http://arxiv.org/abs/2104.11892>>. Acesso em: 19 jul. 2025.

ZHANG, Y. et al. Exploration of Users' Perspectives and Needs and Design of a Type 1 Diabetes Management Mobile App: Mixed-Methods Study. JMIR mHealth and uHealth, 21 set. 2018. v. 6, n. 9, p. e11400.