CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
COORDENADORIA GERAL DE PÓS-GRADUAÇÃO, EXTENSÃO E PESQUISA
MESTRADO PROFISSIONAL EM GESTÃO E TECNOLOGIA EM
SISTEMAS PRODUTIVOS

RENATO WESSNER DOS SANTOS

RECONHECIMENTO DE SINAIS DE LIBRAS PARA APOIO AO ATENDIMENTO EMERGENCIAL: UMA PROVA DE CONCEITO COM ARQUITETURA *DETR* CUSTOMIZADA

São Paulo
Abril/2026

RENATO WESSNER DOS SANTOS

RECONHECIMENTO DE SINAIS DE LIBRAS PARA APOIO AO ATENDIMENTO EMERGENCIAL: UMA PROVA DE CONCEITO COM ARQUITETURA *DETR* CUSTOMIZADA

Dissertação apresentada como exigência parcial para a obtenção do título de Mestre em Gestão e Tecnologia em Sistemas Produtivos do Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos, sob a orientação do Prof. Dr. Fabrício José Piacente.

Área de Concentração: Sistemas Produtivos
Linha de Pesquisa: Gestão da Produção e Operações
ODS - Objetivos do Desenvolvimento Sustentável relacionados a esta dissertação e seus resultados: Redução das desigualdades (10), Paz, justiça e instituições eficazes (16).

São Paulo
Abril/2026

Santos, Renato Wessner dos

S237r Reconhecimento de sinais de libras para apoio ao atendimento emergencial: uma prova de conceito com arquitetura detr customizada / Renato Wessner dos Santos. – São Paulo: CPS, 2026.

173 f. : il.

Orientador (a): Prof. Dr. Fabrício José Piacente

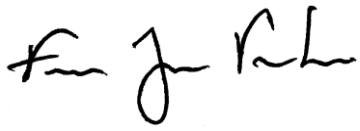
Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos) – Centro Estadual de Educação Tecnológica Paula Souza, 2026.

1. Língua brasileira de sinais. 2. Tecnologias assistivas. 3. Visão computacional. 4. Detection transformer. 5. Serviços de emergência. I. Piacente, Fabrício José . II. Centro Estadual de Educação Tecnológica Paula Souza. III. Título.

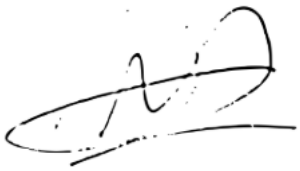
CRB8-8390

RENATO WESSNER DOS SANTOS

RECONHECIMENTO DE SINAIS DE LIBRAS PARA APOIO AO ATENDIMENTO EMERGENCIAL: UMA PROVA DE CONCEITO COM ARQUITETURA *DETR* CUSTOMIZADA



Prof. Dr. Fabrício José Piacente
Orientador – CEETEPS



Prof. Dr. Alexandre de Mello Ferreira
Examinador Externo - UNICAMP

Prof. Dr. Carlos Hideo Arima
Examinador Interno - CEETEPS

São Paulo, 30 de abril de 2026

AGRADECIMENTOS

Dedico este trabalho, antes de tudo, a Deus, que me sustentou com fé, serenidade e direção ao longo de cada etapa, especialmente nos momentos em que a continuidade parecia difícil.

Aos meus pais, minha base, dedico com profunda gratidão. Pelo amor incondicional, pela educação, pelos valores e pelo apoio firme, muitas vezes silencioso, mas sempre presente e que me permitiram seguir adiante com responsabilidade e coragem.

Ao Centro Paula Souza, registro meu reconhecimento pela oportunidade de formação e pelo suporte institucional que possibilitaram o desenvolvimento desta pesquisa. Agradeço aos docentes, orientadores e colaboradores que, direta ou indiretamente, contribuíram para meu crescimento acadêmico e para a consolidação deste trabalho.

À Polícia Militar, expresso minha admiração e gratidão não apenas pelo serviço essencial prestado à sociedade, mas também pelos ensinamentos que reforçaram em mim a disciplina, a constância e, sobretudo, a importância de trabalhar com propósito. Essa visão de missão e compromisso com o bem coletivo foi inspiração permanente e ajudou a orientar o sentido deste estudo.

Por fim, dedico este trabalho a todos que me ajudaram no caminho, familiares, amigos, colegas e profissionais que, com palavras, gestos, apoio prático ou incentivo, contribuíram para que eu chegasse até aqui. A todos, minha gratidão e respeito.

“A comunicação é a ponte que nos conecta,
permitindo o compartilhamento de
pensamentos, sonhos e inspirações.”
Autor desconhecido

RESUMO

SANTOS, R. W. Reconhecimento de sinais em LIBRAS para apoio ao registro de ocorrências de emergência utilizando um *DETR Custom*izado. 173 f. Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2026.

O presente trabalho desenvolveu e avaliou uma prova de conceito para o reconhecimento automático de gestos da Língua Brasileira de Sinais, com foco ao apoio no atendimento emergencial. A proposta fundamenta-se em uma arquitetura *Detection Transformer* customizada, com vocabulário de 20 sinais e eficiência computacional, por meio da redução para uma única camada no *encoder* e no *decoder*, dimensão oculta de 256 e 25 *object queries*. A metodologia incluiu a construção de um *dataset* com 23.968 imagens, proveniente de vídeos do autor e geração de perfis sintéticos, sendo dois empregados no treinamento e um adicional reservado para teste de generalização, com protocolo de divisão para evitar *data leakage*. O modelo foi treinado em fundo controlado e submetido a testes de robustez em fundos ruidosos, sendo comparado a baselines. O *DETR-Custom* alcançou *mAP@0.5* de 0,9114, *FPS* de 104,87 e latência de 9,54 ms. O *F1-Score*, contudo, foi de 0,4702, reflexo do desequilíbrio entre *Precision* e *Recall*, que gerou excesso de falsos positivos. Os resultados confirmam parcialmente a hipótese e demonstram que a customização, notadamente a redução de object queries viabiliza uma solução de baixo custo computacional. Entretanto, o teste de robustez evidenciou degradação do *mAP* para 0,4107 em fundos não controlados, e a assimetria no *transfer learning* em relação aos baselines restringe o escopo atual a ambientes com fundo padronizado. O estudo oferece contribuições tecnológicas e sociais, alinhando-se aos ODS de inclusão e acesso à justiça, e estabelecendo base para futura validação com usuários surdos em cenários reais de emergência.

Palavras-chave: Língua Brasileira de Sinais. Tecnologias assistivas. Visão computacional. *Detection Transformer*. Serviços de emergência.

ABSTRACT

SANTOS, R. W. Reconhecimento de sinais em Libras para apoio ao registro de ocorrências de emergência utilizando um *DETR Custom*izado. 173 f. Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2026.

This study developed and evaluated a proof-of-concept for the automatic recognition of Brazilian Sign Language gestures to support emergency response. The approach leverages a custom Detection Transformer architecture with a 20-sign vocabulary and high computational efficiency, achieved by reducing both the encoder and decoder to a single layer, a hidden dimension of 256, and 25 object queries. The methodology involved building a 23,968-image dataset from the author's videos and synthetic profiles. Two sets were used for training and one for generalization testing, with splitting to prevent data leakage. The model was trained on controlled backgrounds and benchmarked against baselines for robustness on noisy backgrounds. The custom DETR achieved an mAP@0.5 of 0.9114, 104.87 FPS, and 9.54 ms latency. However, the F1-score was 0.4702, reflecting a precision-recall imbalance that caused excessive false positives. Results partially confirm the hypothesis, demonstrating that architectural customization, specifically reducing object queries, enables a computationally efficient solution. Nevertheless, robustness testing revealed a mAP drop to 0.4107 in uncontrolled backgrounds, and asymmetric transfer learning compared to baselines currently restricts the scope to standardized environments. The study provides technological and social contributions, aligning with the UN SDGs for inclusion and access to justice, establishing a foundation for future validation with deaf users in real-world emergency scenarios.

Keywords: Brazilian Sign Language. Assistive technologies. *Computer vision*. *Detection Transformer*. Social inclusion.

LISTA DE FIGURAS

LISTA DE DIAGRAMAS

LISTA DE EQUAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

AP	Average Precision
ASL	American Sign Language
BERT	Bidirectional Encoder Representations from Transformers
BiT	Big Transfer
BSL	British Sign Language
CM	Configuração de mão
CNN	Convolutional Neural Network
CNNs	Redes Neurais Convolucionais
COCO	Common Objects in Context (dataset)
CSLR	Continuous Sign Language Recognition
DETR	Detection Transformer
DL	Deep learning
ENM	Expressões não manuais
FFN	Feed-Forward Network
FN	False Negatives (Falsos Negativos)
FP	False Positives (Falsos Positivos)
FPS	Frames Per Second
GCNs	Redes Convolucionais de Grafos
Grad-CAM	Gradient-weighted Class activation mapping
GRU	Gated Recurrent Unit
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
IoU	Intersection over Union
ISL	Indian Sign Language
KSL	Korean Sign Language
Libras	Língua Brasileira de Sinais
LSTM	Long Short-Term Memory
M	Movimento

mAP	mean Average Precision
MLP	Multi-Layer Perceptron
MHSA	Multi-head Self-attention
NMS	Non-Maximum Suppression
ODS	Objetivos de Desenvolvimento Sustentável
OM	Orientação da palma
PA	Ponto de articulação
PASCAL VOC	Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes
PCD	Pessoas com deficiência
PVT	Pyramid Vision Transformer
R-CNN	Region-based Convolutional Neural Network
RPN	Region Proposal Network
RT-DETR	Real-Time DETR
SSD	Single shot MultiBox Detector
TA	Tecnologias assistivas
ViT	Vision Transformer
ViT-B	ViT-Base
ViT-H	ViT-Huge
ViT-L	ViT-Large
VP	True Positives (Verdadeiros Positivos)
YOLO	You Only Look Once

LISTA DE SÍMBOLOS

Q	Matriz de queries (consultas) do mecanismo de atenção
K	Matriz de keys (chaves) do mecanismo de atenção
V	Matriz de values (valores) do mecanismo de atenção
d_k	Dimensão das keys
d_{model}	Dimensão dos embeddings do modelo Transformer
W_i^Q	Matriz de pesos aprendíveis para projeção de queries
W_i^K	Matriz de pesos aprendíveis para projeção de keys
W_i^V	Matriz de pesos aprendíveis para projeção de values
W^O	Matriz de projeção final do multi-head attention
$head_i$	i-ésima cabeça de atenção
h	Número total de cabeças de atenção
PE	Codificação posicional (positional encoding)
pos	Posição do token na sequência
i	Índice da dimensão do vetor posicional
x	Entrada de uma subcamada do Transformer
N	Número total de predições do modelo
M	Número real de objetos na imagem
q_i	i-ésima object query
σ	Permutação no espaço S_N
$\hat{\sigma}$	Permutação ótima do Hungarian matching
$\mathfrak{S}_N$	Conjunto de todas as permutações possíveis
c_i	Classe verdadeira do objeto i
$\hat{p}_{\hat{\sigma}(i)}\,(c_i)$	Probabilidade prevista para a classe correta
b_i	Bounding box verdadeira do objeto i
$\hat{b}_{\hat{\sigma}}(i)$	Bounding box predita pelo modelo
$\mathbb{1}\{\cdot\}$	Função indicadora
$\emptyset$	Classe "no object" (ausência de objeto*)*
$c_x,\, c_y$	Coordenadas do centro da bounding box
w, h	Largura e altura da bounding box
B_p	Bounding box predita

B_{gt}	Bounding box de referência (*ground truth*)
$\mathcal{L}_{total}$	Função de perda total
$\mathcal{L}_{cls}$	Perda de classificação
$\mathcal{L}_{bbox}$	Perda de bounding box (L1)
$\mathcal{L}_{giou}$	Perda de Generalized IoU
$\mathcal{L}_{match}$	Custo de *matching* por par
$\mathcal{L}_{Hungarian}$	Função de perda do Hungarian *matching*
$\mathcal{L}_{box}$	Perda combinada de bounding box
λ_{cls}	Peso da perda de classificação
λ_{bbox}	Peso da perda de bounding box
λ_{giou}	Peso da perda de *GIoU*
$\bar{t}_{inf}$	Tempo médio de inferência
VP	Verdadeiros positivos
FP	Falsos positivos
FN	Falsos negativos
P	Precision (precisão)
R	Recall (revocação)
$F1$	F1-Score (média harmônica entre P e R)
AP	Average Precision
AP_i	Average Precision da classe i
mAP	mean Average *Precision*
IoU	Intersection over Union
$GIoU$	Generalized Intersection over Union
$\mathbb{R}$	Conjunto dos números reais
H_0, W_0	Altura e largura originais da imagem de entrada
H, W	Altura e largura do mapa de características
C	Número de canais do mapa de características
d	Dimensão reduzida dos canais (256 ou 512)
x_{img}	Imagem de entrada
f	Mapa de ativação do *backbone*
z_0	Mapa de característica após convolução
α	Nível de significância estatística (0,05)

SUMÁRIO

1. INTRODUÇÃO

A comunicação entre pessoas ouvintes e pessoas surdas apresenta desafios relevantes, mesmo diante dos avanços legais e tecnológicos voltados à promoção da inclusão. Esse cenário evidencia a necessidade de soluções que ampliem a acessibilidade comunicacional, especialmente em contextos que demandam respostas rápidas e eficazes, como os atendimentos de emergência.

A Língua Brasileira de Sinais (Libras) é reconhecida oficialmente como meio legal de comunicação e expressão no Brasil. Entretanto, apesar dos avanços normativos, a disponibilidade de intérpretes ainda é limitada, o que dificulta a efetivação do acesso pleno da população surda a serviços essenciais.

Nesse contexto, tecnologias assistivas (TA) baseadas em visão computacional e *deep learning* (*DL*) apresentam-se como alternativas promissoras para o reconhecimento automático de sinais. Tais abordagens possibilitam o desenvolvimento de sistemas capazes de interpretar gestos e auxiliar na mediação comunicacional.

Diante disso, este trabalho propôs o desenvolvimento e a avaliação de modelo de reconhecimento de sinais em Libras, baseado em uma arquitetura *Detection Transformer* (*DETR*) customizada, com foco em eficiência computacional e aplicabilidade em contextos de emergência.

1.1 Contextualização

Segundo Abdul Ameer *et al*. (2024), indivíduos com deficiência auditiva representam mais de 5% da população mundial, o que, em termos absolutos, equivale a aproximadamente 360 milhões de pessoas. Esses indivíduos frequentemente enfrentam dificuldades de comunicação com pessoas oralizadas. González Rodríguez *et al*. (2024) e Mahmud *et al*. (2023) destacam que comunidades de pessoas com deficiência (PCD) auditiva encontram desafios no acesso a serviços fundamentais, como educação, saúde e emprego, devido à barreira linguística que dificulta as interações com a população oralizada. Embora existam intérpretes de língua de

sinais, sua quantidade é insuficiente para promover uma interação social satisfatória entre oralizados e não oralizados.

A língua de sinais constitui um sistema de comunicação visual que utiliza gestos, expressões faciais e movimentos corporais para transmitir ideias e sentimentos, sendo fundamental para garantir a comunicação eficaz de pessoas surdas em seu cotidiano. No Brasil, segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE) (2019), divulgados por meio da plataforma SIDRA - Sistema IBGE de Recuperação Automática, a população com deficiência auditiva somava aproximadamente 2.330.442 pessoas, concentradas majoritariamente na Região Sudeste (40,82%), seguida pelas Regiões Nordeste (29,08%), Sul (16,02%), Norte e Centro-Oeste (6,33%).

No âmbito legal, a Constituição Federal Brasileira (Brasil, 1988), em seu artigo 208, assegura o atendimento educacional especializado às PCD, preferencialmente na rede regular de ensino. Posteriormente, a Lei nº 10.436/2002 reconheceu a Libras como meio legal de comunicação e expressão, regulamentada pelo Decreto nº 5.626/2005, consolidando seu status como uma das línguas oficiais do país. Essas normativas garantem à comunidade não oralizada direitos como educação bilíngue, presença de intérpretes em instituições públicas e privadas e acesso a serviços de tradução e interpretação.

Apesar dos avanços legislativos, a realidade evidencia barreiras comunicacionais significativas entre oralizados e não oralizados, especialmente em contextos que demandam interação imediata, como atendimentos em unidades de saúde, serviços de emergência, órgãos públicos, delegacias e departamentos de segurança pública. No âmbito dos atendimentos de emergência observa-se que essas barreiras têm sido enfrentadas gradualmente por iniciativas integradas das forças de segurança. Conforme Barreto (2023), o Projeto E-SMS da Polícia Militar do Estado de São Paulo registrou, nos últimos cinco anos, 1.131 comunicações à central de emergência 190, distribuídas entre 2018 e 2022. Dados relativos ao registro de boletins de ocorrência também demonstram uso crescente desses canais desde 2019 (Secretaria da Segurança Pública, 2019).

Apesar dos esforços para a criação de mecanismos legais e tecnológicos voltados à inclusão da população não oralizada, persiste um desafio relevante: o desenvolvimento de ferramentas capazes de viabilizar a tradução efetiva de diálogos em Libras para a Língua Portuguesa oral-auditiva e vice-versa. A crescente demanda por *TA* voltadas à inclusão social desse grupo em ambientes educacionais e profissionais evidencia a relevância e o impacto dessas soluções no cotidiano da população não oralizada.

Paralelamente a esse contexto social, os avanços nas áreas de visão computacional e aprendizado de máquina têm ampliado de forma significativa as possibilidades de desenvolvimento de sistemas capazes de reconhecer e interpretar gestos e línguas de sinais. Abordagens convencionais, fundamentadas em algoritmos simples e na extração manual de características, frequentemente apresentam limitações quando aplicadas a ambientes complexos do mundo real, motivando a adoção de técnicas mais avançadas. Nesse cenário, destacam-se métodos de detecção baseados em *Transformers*, que possibilita a captura de relações espaciais e contextuais globais presentes na imagem, promovendo uma compreensão mais abrangente dos objetos analisados (Carion *et al*., 2020).

1.2 Problemática

Apesar dos avanços tecnológicos e do arcabouço legal voltados à inclusão de PCD auditiva, persistem desafios significativos que podem limitar a efetivação plena dos direitos comunicacionais da população não oralizada. Observa-se uma escassez expressiva de intérpretes e tradutores de Libras disponíveis. De acordo com dados do RAIS (2024) que constam no Anexo A (imagem do portal RAIS e método de consulta), existem atualmente 134 intérpretes registrados formalmente no Ministério do Trabalho e Emprego, número que considera apenas profissionais contratados via regime CLT e distribuídos de forma desigual entre os estados brasileiros.

Esse contingente pode ser insuficiente frente à população surda brasileira, existindo um intérprete para cada 17.391 pessoas surdas. Ainda que se estime a existência de aproximadamente 20 mil intérpretes considerando vínculos formais e informais, esse número permanece insuficiente, resultando em uma proporção aproximada de um intérprete para cada 117 pessoas surdas.

No campo tecnológico, embora existam diversos sistemas de reconhecimento automático de língua de sinais desenvolvidos ao longo das últimas décadas, muitos apresentam limitações relevantes para aplicação em contextos reais de emergência. Tais limitações incluem vocabulários restritos, frequentemente associados à datilologia, ausência de especialização contextual e elevada complexidade computacional. Determinadas arquiteturas de *DL*

demandam *hardware* especializado, especialmente *GPUs* de alto desempenho, o que dificulta sua implementação em dispositivos com recursos computacionais moderados.

Diante desse cenário, a questão central desta pesquisa é: de que forma a customização de um modelo de detecção de objetos baseado na arquitetura *DETR* pode equilibrar acurácia (*mAP@0.5* ≥ 0,80; *F1-Score* ≥ 0,80) e custo de processamento (latência ≤ 50 ms; *FPS* ≥ 15), viabilizando o reconhecimento de 20 sinais de Libras em cenários de atendimento e*merge*ncial com recursos computacionais limitados?

1.3 Hipótese

A hipótese desta pesquisa é que um modelo *DETR-Custom*, com arquitetura simplificada, especializado no reconhecimento de um vocabulário controlado de vinte sinais em Libras, é capaz de atingir desempenho de detecção equivalente ou superior a modelos *baseline* de detecção de objetos consolidados, mantendo maior eficiência computacional, tornando-se viável para aplicações em ambientes com recursos limitados.

Essa hipótese é considerada confirmada caso o modelo proposto atinja *mAP@0.5* igual ou superior a 0,80 e *F1-Score* médio igual ou superior a 0,80, além de apresentar tempo de inferência inferior ou igual a 50 ms por imagem e taxa de processamento mínima de 15 *frames per second* (*FPS*). Adicionalmente, o modelo deverá demonstrar desempenho equivalente ou superior no *trade-off* entre acurácia e eficiência quando comparado a, pelo menos, sois modelos *baseline* (*YOLOv8*, *Faster R-CNN* ou *DETR* original), avaliados sob as mesmas condições de *dataset* e *hardware*.

Caso a hipótese seja rejeitada, os dados obtidos poderão fornecer informações sobre as limitações da abordagem proposta e indicar direções para pesquisas futuras, como ajustes adicionais na arquitetura, estratégias de treinamento ou necessidade de ampliação do *dataset*.

A confirmação ou rejeição da hipótese desta dissertação poderá contribuir para o entendimento sobre a customização de arquiteturas *Transformer* em tarefas especializadas de visão computacional. Esta proposição será testada empiricamente por meio dos objetivos específicos detalhados a seguir.

1.4 Objetivos

Diante da problemática apresentada, os objetivos foram estruturados de modo a contemplar tanto os aspectos técnicos do desenvolvimento e avaliação do modelo quanto a análise de sua viabilidade prática como tecnologia assistiva, considerando restrições computacionais.

Este estudo delimita-se ao reconhecimento de vinte sinais da Libras por meio do processamento independente de *frames*, em ambiente controlado, não abrangendo a modelagem de sequências temporais contínuas, a tradução automática de diálogos completos nem a validação em cenários reais de atendimento emergencial.

1.4.1 Objetivo geral

Avaliar o desempenho de uma arquitetura *DETR* customizada para o reconhecimento de 20 sinais de Libras pré-definidos em ambiente controlado, quantificando sua acurácia e eficiência computacional em relação a modelos *baseline* consolidados, para apoio ao atendimento emergencial.

1.4.2 Objetivos específicos

a) Realizar pesquisa bibliográfica exploratória para identificar e comparar arquiteturas de detecção de objetos aplicáveis ao reconhecimento de línguas de sinais;

b) Construir e anotar um *dataset* específico para o reconhecimento de sinais de Libras no contexto de atendimento e*merge*ncial, preservando a integridade entre os subconjuntos de treino, validação e teste;

c) Propor e implementar o *DETR-Custom*, versão da arquitetura *DETR* com simplificações estruturais voltadas à eficiência computacional;

d) Avaliar o desempenho do DETR-*Custom*, comparando os resultados experimentais com os *thresholds* quantitativos da hipótese (*mAP@0.5* ≥ 0,80; *F1-Score* ≥ 0,80; latência ≤ 50 ms; *FPS* ≥ 15), explicitando o grau de confirmação ou refutação de cada parâmetro;

e) Comparar o *DETR-Custom* com modelos *baseline* consolidados (*YOLOv8*, *Faster R-CNN* e *DETR* original) no *trade-off* entre acurácia e eficiência computacional;

f) Analisar a interpretabilidade da arquitetura *DETR-Custom* por meio dos mapas de *cross-attention* do *decoder*.

1.5 Justificativa

A realização deste trabalho justifica-se por razões de natureza social, tecnológica e prática, que convergem para a necessidade de desenvolvimento de uma solução capaz de promover a inclusão efetiva da população não oralizada no acesso a serviços fundamentais de emergência.

Sob a perspectiva social, o desenvolvimento de *TA* para o reconhecimento de Libras relacionou-se diretamente à efetivação de direitos fundamentais, como o acesso à justiça e à comunicação em condições de igualdade, assegurados pelo Decreto nº 6.949/2009, que garante, em seus artigos 9º e 13, o direito à acessibilidade e ao acesso à justiça em igualdade de condições para PCD (Brasil, 2009). A contribuição do estudo nesta esfera social, concretizou-se na proposição de uma ferramenta voltada especificamente ao contexto de atendimento emergencial. Diferentemente de outras pesquisas que abordam a Libras sob a ótica de sequências temporais de vídeo, esta pesquisa contribuiu ao explorar a abordagem de detecção da parte mais significativa do gesto, dessa forma viabilizando a localização e classificação de sinais em vídeo contínuo. Com essa abordagem buscou-se reduzir as barreiras comunicacionais enfrentadas por pessoas surdas, alinhando-se aos Objetivos de Desenvolvimento Sustentável (ODS) 10 e 16.

Do ponto de vista tecnológico, a lacuna na literatura, onde não foram identificados estudos que aplicassem arquiteturas *Transformer*, especificamente o *DETR*, ao reconhecimento da Libras em contextos emergenciais. O trabalho preencheu essa lacuna ao propor uma

adaptação da arquitetura *DETR-Custom*. A contribuição metodológica incluiu a aplicação de um protocolo de avaliação simultânea, combinando métricas de acurácia (*mAP*, *Precision*, *Recall*, *F1-Score*), eficiência computacional (*FPS*, tempo de inferência) e análise qualitativa por meio da visualização da *cross-attention* do *decoder*. Adicionalmente, o estudo contribuiu ao demonstrar a viabilidade de um *pipeline* adaptado para *datasets* de tamanho limitado, utilizando técnicas de *data augmentation* e ajuste de hiperparâmetros, o que ajudou a comprovar a eficácia do modelo mesmo em cenários com restrições de *hardware*.

Já, sob a ótica prática, a eficiência computacional alcançada pelo modelo *DETR-Custom* demonstrou sua aplicabilidade em ambientes que não dispõem de *GPUs* de alto custo, viabilizando sua integração a sistemas de atendimento como ferramenta de apoio em situações nas quais não existem intérpretes disponíveis. A validação da proposta como prova de conceito, utilizando um vocabulário controlado com 20 sinais aliado com a arquitetura modular e escalável do modelo, permitiu estabelecer as bases para a expansão incremental desse vocabulário em trabalhos futuros.

Dessa forma, o produto tecnológico que é resultado dessa pesquisa consiste em um artefato em três camadas interdependentes: (i) o modelo *DETR-Custom*, responsável pelo reconhecimento de sinais de Libras em tempo real a partir do fluxo de vídeo, que foi obtido por meio do treino das imagens; (ii) uma *API* de comunicação, que recebe as predições geradas pelo modelo e as disponibiliza a sistemas externos; e (iii) uma interface de usuário, que foi concebida como prova de conceito, que simula o preenchimento automático dos campos de uma ocorrência dos dados que se imaginam necessários a uma emergência à medida que os sinais são identificados. Essa arquitetura modular garante que a integração entre o modelo treinado e o sistema de apoio ao atendimento emergencial permita, em novas versões, a substituição ou a expansão de qualquer uma dessas camadas sem prejuízo à integridade da solução. Além do esclarecimento de viabilidade técnica da abordagem, o estudo gerou um protocolo de validação e um artefato de software adaptável a diferentes serviços de emergência, contribuindo para a construção de instituições mais inclusivas e eficazes.

1.6 Organização do trabalho

A dissertação está organizada em cinco partes, estruturadas de forma sequencial, de modo a auxiliar o leitor a acompanhar o contexto do estudo, a metodologia de pesquisa, os resultados dos experimentos e as implicações do trabalho de maneira clara e progressiva.

A introdução apresentou o contexto do problema de pesquisa, trazendo dados sobre a comunidade surda brasileira e sua interação com os serviços de emergência, bem como uma breve contextualização sobre os modelos atuais para reconhecimento de imagens. Foram apresentados, a problemática que motivou o trabalho, a justificativa para sua realização, o objetivo geral e os objetivos específicos, a hipótese e a organização da dissertação.

No capítulo dois, a fundamentação teórica reuniu os conhecimentos necessários para o desenvolvimento do projeto, iniciando pela caracterização da Libras e, em seguida, apresentando a evolução dos métodos de visão computacional para reconhecimento de gestos, desde abordagens clássicas até técnicas modernas baseadas em *DL*, com destaque para o *DETR*.

No capítulo três, foi discutida a metodologia empregada no desenvolvimento desta dissertação, incluindo a escolha dos *backbones*, configurações adotadas, construção do *dataset*, técnicas de *data augmentation*, detalhamento da estratégia de treinamento e apresentação das métricas de avaliação utilizadas.

No capítulo quatro, foram descritos os experimentos conduzidos e analisados os resultados obtidos, em confronto com a hipótese formulada, com a identificação das limitações observadas e o estabelecimento de conexões com trabalhos prévios da literatura.

No capítulo cinco, foi apresentado a proposta visual do sistema que usara o modelo treinado para o preenchimento de ocorrência.

A conclusão apresentou propostas para trabalhos futuros, sintetizando as principais contribuições da dissertação, retomando os objetivos estabelecidos no primeiro capítulo e avaliando seu cumprimento.

Os apêndices e anexos reuniram materiais suplementares que apoiam o texto principal, como listagens de código, figuras e tabelas adicionais, bem como descrições detalhadas dos conjuntos de dados e ferramentas utilizados na pesquisa.

Essa organização teve o propósito de proporcionar uma leitura fluida e progressiva, permitindo aos leitores compreender o trabalho desenvolvido, seus fundamentos, metodologia, resultados e conclusões, além de possibilitar que outros pesquisadores possam reproduzir os experimentos e construir sobre as contribuições aqui estabelecidas.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem por objetivo aprofundar a compreensão dos principais conceitos abordados na literatura relacionados à Libras, à detecção de objetos e à arquitetura *DETR*, bem como às suas aplicações. Mais especificamente, a fundamentação teórica enfatiza o desenvolvimento das tecnologias de detecção de gestos, os desafios inerentes a esse campo.

2.1 Língua brasileira de sinais (Libras)

A Libras constitui o principal meio de comunicação da comunidade não oralizada no Brasil e representa um importante marco da diversidade linguística e cultural brasileira. Seu reconhecimento impulsionou o desenvolvimento de pesquisas voltadas à análise de sua estrutura linguística, ao ensino e à difusão em contextos acadêmicos, institucionais e profissionais. Assim como as línguas de sinais utilizadas em outros países, a Libras apresenta organização linguística complexa, com níveis fonológicos, morfológicos e sintáticos próprios, articulados no espaço visual (Quadros; Karnopp, 2004).

2.1.1 Características linguísticas

A Libras é utilizada pela comunidade não oralizada brasileira como meio de comunicação e expressão, sendo dotada de estrutura gramatical própria e independente da Língua Portuguesa. A principal diferença entre a língua utilizada por pessoas oralizadas e a Libras reside no canal de comunicação predominante: enquanto a primeira baseia-se no canal oral-auditivo, a segunda utiliza o canal visual-espacial. Nesse contexto, a Libras é expressa por meio de configurações e movimentos das mãos, associados a expressões faciais e corporais, percebidos visualmente pelo interlocutor.

Maiolini e Maiolini (2023) discutem o reconhecimento legal da Libras, fundamentado na Lei Federal nº 10.436, de 24 de abril de 2002, que a estabelece como meio legal de comunicação e expressão, com sistema linguístico de natureza visual-motora e estrutura gramatical própria. Os autores ressaltam, contudo, que a legislação não é totalmente explícita quanto ao seu reconhecimento formal como língua oficial. Conforme dispõe o Art. 1º da referida lei:

> Art. 1° É reconhecida como meio legal de comunicação e expressão a Língua Brasileira de Sinais - Libras e outros recursos de expressão a ela associados.
>
> Parágrafo único. Entende-se como Língua Brasileira de Sinais – Libras a forma de comunicação e expressão, em que o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constitui um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil (Brasil, 2002).

Segundo Menezes, Faiad e Silva (2024), a Libras possui relevância central por sustentar a comunicação entre pares e favorecer dimensões cognitivas e sociais da comunidade não oralizada. Nesse enquadramento, a língua é caracterizada como um sistema linguístico visuomotor, com estrutura gramatical própria, o que afasta leituras que a tratem apenas como conjunto de gestos. Oliveira Martins e Lopes (2024) reforçam essa compreensão ao atribuir à Libras função essencial de comunicação e expressão nas comunidades surdas. Já Carvalho e Manzini (2017) ampliam o debate ao situar a Libras no contexto brasileiro tanto como língua de atendimento educacional quanto como interface comunicacional entre oralizados e não oralizados, evidenciando sua crescente circulação social mediada, em grande medida, pela atuação de intérpretes. Contudo, a presença ampliada da Libras não implica homogeneidade: as variações regionais, dialetos, revelam diferenças geográficas, culturais e sociolinguísticas que incidem na forma dos sinais e em aspectos gramaticais, configurando um fenômeno comparável ao das línguas orais e exigindo cautela em abordagens que pressuponham padronização única.

2.1.2 Parâmetros dos sinais

A formação dos sinais se dá com base em cinco parâmetros: configuração de mão (CM), orientação da palma (OM), movimento (M), ponto de articulação (PA) e as expressões não manuais (ENM), a permutação desses cinco parâmetros permite a formação de uma ampla

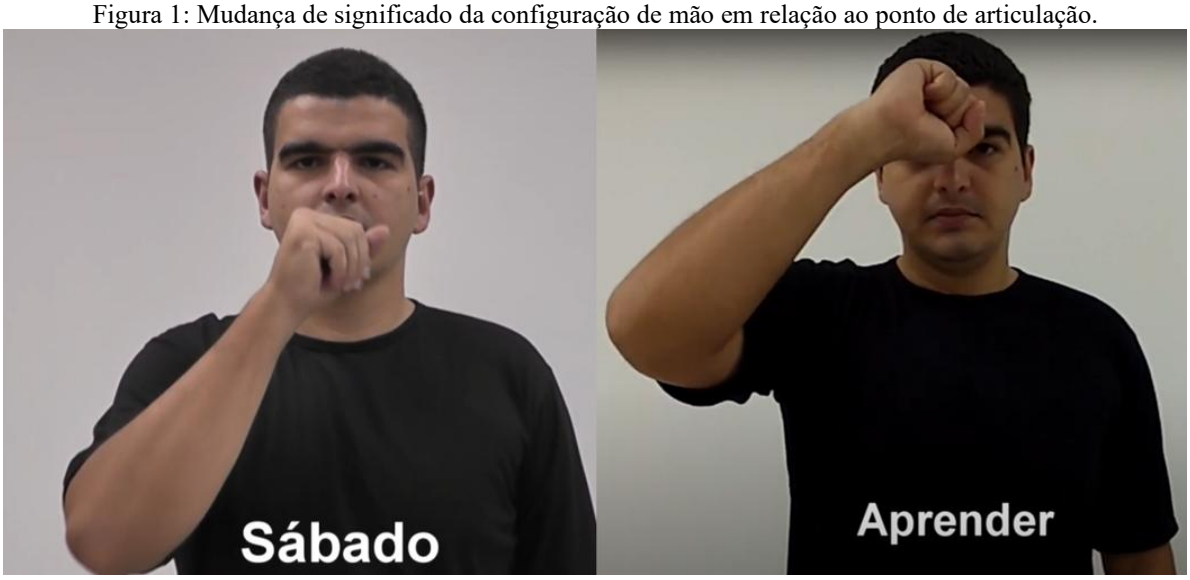
variedade de gestos, e, algumas alterações em determinado parâmetro podem gerar a mudança de significado.

A estrutura fonológica da Libras é composta por parâmetros que se combinam para formar os gestos, essas unidades foram inicialmente propostas por Stokoe, em 1960, abarcando a CM, PA e M, já os estudos de Quadros e Karnopp (2004) abarcaram outros dois parâmetros: o de ENM e o de OM das mãos. A seguir, apresenta-se a descrição de cada parâmetro.

A CM refere-se à forma assumida pela mão durante a execução do sinal. Segundo Felipe (2005), foram catalogadas 64 configurações distintas, cuja alteração pode modificar completamente o significado do gesto.

Os pontos de articulação indicam o local no corpo ou no espaço onde o sinal é realizado, sendo esses delimitados pela extensão dos braços do gesticulante e podem ocorrer em partes específicas do corpo - como cabeça, tronco ou braços - ou mesmo em espaço neutro à frente do sinalizante. Os gestos como o da palavra “SÁBADO” e “APRENDER”, por exemplo, compartilham a mesma CM, mas distinguem-se pelo PA: o primeiro na testa e o segundo na boca, como observado na Figura 1.

Figura 1: Mudança de significado da configuração de mão em relação ao ponto de articulação.

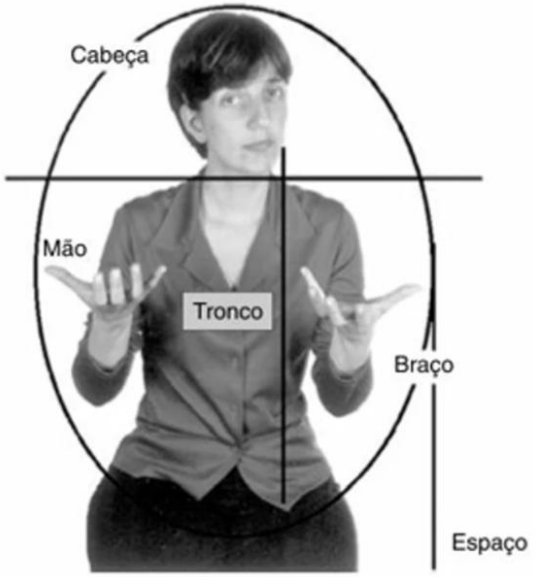


Fonte: Imagem extraída dos vídeos do canal Incluir Tecnologia, (2012) e (2012).

O M caracteriza-se pela forma como ocorre o deslocamento das mãos no espaço durante a execução do gesto. Trata-se de um dos parâmetros mais complexos, pois pode envolver diferentes direções e trajetórias. Alguns sinais são estáticos, enquanto outros apresentam

movimentos lineares, circulares, arqueados ou alternados. A organização do espaço de realização dos sinais pode ser observada na Figura 2.

Figura 2: Espaço de realização dos sinais e as quatro áreas principais de articulação dos sinais.

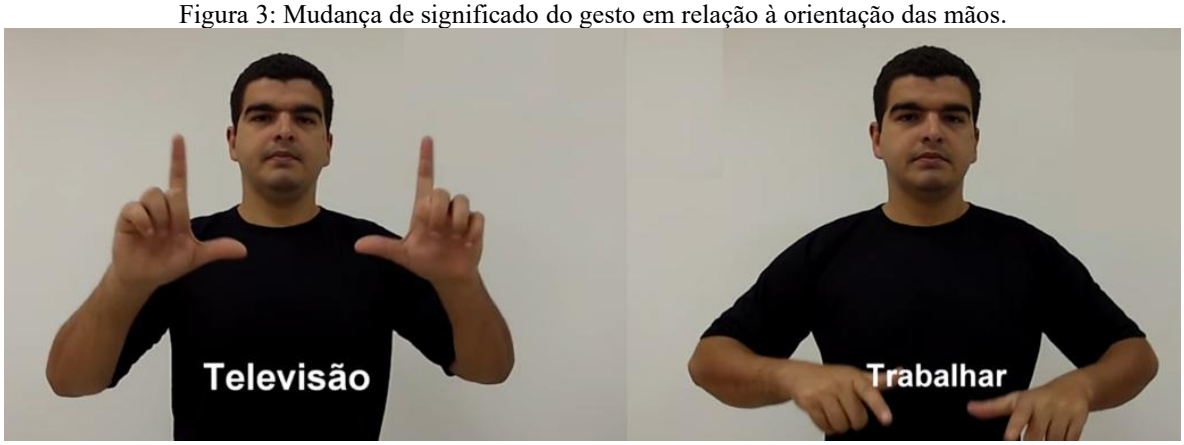


Fonte: Imagem extraída do livro língua de sinais brasileira: estudos linguísticos, (Quadros; Karnopp, 2004).

A orientação das palmas das mãos refere-se à direção para a qual a palma está voltada durante a execução do sinal, podendo estar orientada para cima, para baixo, para frente, para trás ou em direção ao corpo do gesticulante. Esse parâmetro pode ser distintivo, pois sinais com a mesma CM, PA e M podem diferir exclusivamente pela OM, resultando em significados distintos. Os gestos correspondentes às palavras “TRABALHAR” e “TELEVISÃO”, por exemplo, compartilham a mesma CM e o mesmo PA, mas diferenciam-se pela orientação das palmas: no primeiro, as palmas estão voltadas para baixo; no segundo, estão voltadas para frente, com o dorso orientado para o gesticulante, como apresentado na Figura 3.

As ENM correspondem aos movimentos da face, dos olhos, da cabeça e do tronco que acompanham os sinais manuais. De acordo com Quadros e Karnopp (2004), essas expressões desempenham duas funções principais: a construção sintática e a diferenciação lexical. Os estudos pioneiros sobre esse parâmetro na Libras foram realizados por Ferreira-Brito e Langevin, em 1995, que identificaram e sistematizaram as ENM como componentes essenciais da língua de sinais brasileira.

Figura 3: Mudança de significado do gesto em relação à orientação das mãos.



Fonte: Imagem extraída dos vídeos do canal Incluir Tecnologia, (2012) e (2012).

2.1.3 Importância social e acessibilidade

Segundo Preato *et al.* (2020), a análise da importância social da Libras e das questões de acessibilidade no Brasil tem como ponto de partida os marcos legais estabelecidos pela Lei nº 10.436/2002 e pelo Decreto nº 5.626/2005. Esses dispositivos reconhecem a Libras como meio legal de comunicação da comunidade surda e preveem sua inclusão nos currículos educacionais, tanto para a formação de intérpretes quanto para a garantia do direito à educação bilíngue. No entanto, a previsão legal difere significativamente do que se observa na prática educacional e social. Para Silva, Gavaldão e Martins (2020), um dos principais problemas reside no próprio discurso normativo, uma vez que determinadas normas tratam a Libras de forma reduzida, ora como uma adaptação, ora como um recurso educacional ou tecnologia assistiva, o que, segundo os autores, pode desconsiderar seu status de língua plenamente constituída, autores como Silva, Gavaldão e Martins (2020) argumentam que, a comunidade surda é frequentemente tratada sob um viés deficitário, e não como um grupo linguístico e cultural distinto, no qual a inclusão é formalmente prevista, mas o acesso à sua língua própria ainda constitui um desafio significativo.

A implementação das políticas educacionais vigentes é frequentemente discutida na literatura como um processo que enfrenta desafios, observa-se, por exemplo, que mesmo instituições que se autodenominam bilíngues nem sempre adotam a Libras como língua principal de instrução, tratando-a, em alguns contextos, como língua secundária. Em muitos casos, a literatura sugere que o ensino da Libras ocorre de maneira semelhante à abordagem universitária voltada a pessoas oralizadas, com ênfase em aspectos culturais amplos da

comunidade surda e, conforme apontam alguns estudos, um aprofundamento menos sistemático nas dimensões linguísticas e gramaticais. Fernandes (2022) aponta que essa realidade está diretamente associada à carência de professores fluentes em Libras. Ademais, políticas de acessibilidade previstas em lei, como a presença obrigatória de intérpretes, enfrentam desafios de aplicabilidade, uma vez que, na prática, mostram-se mais recorrentes em instituições federais de ensino superior, enquanto redes estaduais, municipais e privadas estão, por vezes, sujeitas a normativas de cumprimento menos abrangente (Silva; Gavaldão; Martins, 2020).

As barreiras linguísticas ultrapassam o campo educacional e alcançam setores críticos, como o da saúde. De acordo com Soares (2024), apontam que a maioria dos cursos da área da saúde no Brasil não costuma oferecer o ensino de Libras em seus currículos, ou o faz apenas como disciplina optativa. Essa lacuna pode comprometer significativamente a qualidade do atendimento à população surda, conforme discutido por Soares (2024) dificultando o processo de diagnóstico e a oferta de tratamentos adequados, uma vez que a comunicação efetiva é imprescindível para a garantia do direito à saúde.

Em contraste com as dificuldades enfrentadas por alunos surdos em processos de oralização, Rodrigues (2020) destaca que a Libras apresenta maior afinidade com linguagens fundamentadas na visualidade e na semiótica. Um exemplo citado pelo autor é a cartografia, que, por utilizar signos, símbolos e leitura visual, mostrou-se uma ferramenta pedagógica acessível e eficaz para alunos surdos. Os resultados do estudo 3.1 que esses estudantes demonstraram facilidade na compreensão da linguagem cartográfica, evidenciando o potencial dessa abordagem como estratégia educacional inclusiva, independentemente da mediação por línguas orais.

Diante dos desafios comunicacionais enfrentados pela população surda no cotidiano, diversas ferramentas vêm sendo desenvolvidas com o objetivo de promover maior autonomia. Cunha e Franco (2021) apresentam, como exemplo, o site Glossário em Libras, voltado ao ambiente educacional, que disponibiliza vídeos com termos e expressões utilizados em espaços como bibliotecas, secretarias e cantinas. A ferramenta possibilita que estudantes surdos realizem solicitações e obtenham informações sem dependência integral de intérpretes. Avaliada quanto à sua eficácia, a iniciativa demonstrou-se capaz de contribuir para a difusão da Libras no ambiente escolar e para a melhoria da interação da comunidade surda no cotidiano acadêmico.

Em síntese, embora os marcos legais tenham promovido avanços relevantes no acesso e na permanência de alunos surdos em determinadas instituições de ensino, a qualidade dessa

inclusão ainda demanda aprimoramento significativo. A distância entre os dispositivos legais e sua aplicação prática permanece como um ponto de atenção na literatura e ainda se mostra um aspecto desafiador, indicando que a questão atual vai além da simples matrícula desses estudantes. Torna-se necessária a criação de interfaces comunicacionais eficazes e a formação de profissionais fluentes em Libras, de modo a garantir que a população surda seja atendida de forma adequada nos diversos setores da sociedade.

2.1.4 Seleção dos sinais para o estudo

Para o presente estudo, foram selecionados vinte sinais da Libras: “ZERO”, “UM”, “DOIS”, “TRÊS”, “QUATRO”, “CINCO”, “SEIS”, “SETE”, “OITO”, “NOVE”, “AVANÇAR”, “VOLTAR”, “CASA”, “EU”, “VER”, “HOMEM”, “ROUBAR”, “SIM”, “NÃO” e “NEUTRO”. A escolha desse conjunto de sinais fundamenta-se em critérios pragmáticos e em sua aplicabilidade direta ao contexto de registro de ocorrências em serviços de emergência, configurando um vocabulário controlado para validação da proposta de reconhecimento automático.

A combinação desses gestos possibilita a construção de enunciados básicos, porém essenciais, para o relato de ocorrências, tais como “homem roubar” e “ver homem roubar”, entre outras variações comunicativas relevantes ao contexto emergencial.

Os sinais escolhidos apresentam diversidade quanto aos cinco parâmetros fonológicos da Libras, permitindo avaliar a capacidade do modelo proposto em distinguir gestos com diferentes CM, pontos de articulação, orientações das mãos, M e ENM. Esses parâmetros compreendem: a forma assumida pela mão durante a produção do sinal; o local do corpo ou do espaço onde o gesto é realizado; o deslocamento das mãos no espaço que pode ser linear, circular, arqueado, ascendente ou descendente, havendo ainda sinais estáticos; a orientação das mãos, que pode variar entre cima, baixo, frente, atrás, lateral ou em direção ao corpo; e as ENM, relacionadas aos movimentos faciais e corporais que acompanham os sinais.

Os parâmetros fonológicos dos sinais selecionados estão descritos no Apêndice A (configuração dos sinais de Libras), sendo que a seleção desse conjunto reduzido de gestos para validação segue os preceitos discutidos por Gupta (2022), que demonstra que sistemas treinados inicialmente com vocabulários restritos podem ser progressivamente expandidos sem

comprometer o desempenho, permitindo a validação rigorosa do *pipeline* proposto antes da ampliação para conjuntos léxicos mais extensos.

Os vinte sinais selecionados constituem elementos fundamentais do léxico necessário para a comunicação básica em contextos de emergência. Segundo Dey *et al.* (2022), a priorização de vocabulários específicos de domínio reduz o ruído linguístico e torna a tradução e o reconhecimento automático estratégias mais viáveis para aplicações práticas de texto e sinais.

Por fim, a limitação às vinte classes nesta fase inicial do estudo possibilita concentrar esforços na otimização da arquitetura proposta e na avaliação criteriosa de seu desempenho, estabelecendo bases sólidas para futuras expansões do vocabulário reconhecido.

2.1.5 Tecnologias assistivas e reconhecimento de sinais

Ao longo dos anos, a tecnologia tem passado por avanços significativos, despertando crescente interesse em sua aplicação como suporte às PCD, de modo a auxiliá-las em atividades cotidianas e ampliar sua autonomia.

Segundo Nascimento *et al.* (2022), as *TA* consistem em estratégias destinadas a promover o aumento da funcionalidade das PCD, possibilitando o desenvolvimento de sua autonomia por meio de equipamentos, recursos e serviços especializados. Dyzel *et al.* (2020) conceituam a tecnologia assistiva como a aplicação de qualquer tecnologia que permita a uma pessoa realizar uma atividade de forma considerada usual, a qual não conseguiria executar de maneira independente. Já Benevides *et al.* (2024) destacam que a adaptação de equipamentos por meio da tecnologia para atender às necessidades individuais deve ser um princípio central nos processos de inclusão das pessoas que demandam o uso de *TA*.

No contexto específico das PCD auditiva, essas tecnologias tornam-se particularmente relevantes. Silveira *et al.* (2022) apresentam dados da Organização Mundial da Saúde (*OMS*), segundo os quais aproximadamente 430 milhões de pessoas no mundo possuem algum grau de perda auditiva incapacitante. De acordo com os autores, as *TA* desempenham papel fundamental na mitigação das barreiras sociais decorrentes dessa condição, sendo aplicadas por meio de

wearables, como luvas capazes de transformar sinais em fala, ou por soluções baseadas em computação gráfica, voltadas à tradução de texto e áudio em língua de sinais em tempo real.

Entretanto, parte significativa das tecnologias existentes ainda apresenta alto custo de aquisição e limitações práticas de uso, o que pode restringir sua adoção em larga escala. Dessa forma, evidencia-se a necessidade do desenvolvimento de *TA* que sejam de fácil utilização, rápida implementação e compatíveis com ambientes reais de atendimento, contribuindo de maneira efetiva para a redução das barreiras comunicacionais enfrentadas por pessoas não oralizadas.

2.2 Visão computacional para reconhecimento de gestos

O reconhecimento de sinais e gestos por meio da visão computacional constitui um campo de pesquisa diretamente relacionado à compreensão da interação humano-computador. Segundo Mohamed, Hassan e Jamil (2024), os métodos de reconhecimento evoluíram desde abordagens tradicionais baseadas em características *handcrafted* até técnicas mais sofisticadas fundamentadas em *DL*. Essas abordagens têm sido aplicadas em diversos domínios, como tradução de línguas de sinais, controle robótico, realidade virtual e aumentada, detecção de objetos e aplicações na área da saúde.

Zhang e Jiang (2024) discutem o uso de modelos avançados, como os *Transformer*s, para lidar com a complexidade do Reconhecimento Contínuo de Língua de Sinais, destacando sua capacidade de modelar dependências de longo alcance por meio de mecanismos de *self-attention*. Os autores também apontam desafios importantes da área, como a necessidade de grandes volumes de dados anotados para o treinamento desses modelos e o desenvolvimento de arquiteturas mais leves e eficientes, adequadas a aplicações móveis e ambientes com restrições computacionais.

Por sua vez, Foteinos *et al.* (2025) apresentam uma análise dos principais avanços recentes em *DL* aplicados à visão computacional para o reconhecimento de gestos, discutindo as arquiteturas mais recorrentes, como Redes Neurais Convolucionais (*CNNs*) 2D e 3D, Redes Convolucionais de Grafos (*GCNs*), utilizadas em dados baseados na captação de esqueletos, e modelos baseados em *Transformer*s.

Esta subseção da dissertação apresenta, portanto, a evolução histórica dos métodos utilizados para reconhecimento de gestos e sinais, abordando desde as técnicas clássicas até os avanços proporcionados pelo *DL*, bem como os principais desafios contemporâneos enfrentados pela área.

2.2.1 Evolução dos métodos de reconhecimento

A evolução dos métodos de reconhecimento de gestos passou por transformações significativas ao longo das últimas décadas. Esses avanços estão diretamente associados ao desenvolvimento tecnológico, tanto em *hardware* quanto em *software*, que ampliaram substancialmente o número de aplicações baseadas em interação humano-computador. Observa-se uma transição progressiva dos sistemas baseados em sensoriamento físico para abordagens fundamentadas em visão computacional, culminando na atual era de *DL*. De modo geral, essa evolução pode ser organizada em três períodos principais: de 1977 a 1990, caracterizado pela era dos sensores; de 1990 a 2010, marcado pela transição para a visão computacional; e, a partir de 2010, a era do *DL*.

O período inicial foi caracterizado pelo uso de dispositivos baseados em sensores físicos. Segundo Dipietro, Sabatini e Dario (2008), um dos primeiros dispositivos desenvolvidos foi a *Sayre Glove*, criada em 1977 por Thomas DeFanti e Daniel Sandin. Esse equipamento utilizava tubos flexíveis contendo luz e uma fotocélula, correlacionando a variação de voltagem à flexão de cada dedo da mão. Posteriormente, em 1983, surgiu a *Digital Entry Data Glove*, desenvolvida por Gary Grimes, que incorporava sensores de toque, sensores de flexão das articulações dos dedos, sensores de inclinação, além de sensores para medir a torção do antebraço e a flexão do pulso. Os sinais capturados eram convertidos em caracteres alfanuméricos, baseados em um subconjunto dos códigos *ASCII* imprimíveis. Outras possibilidades de sensoriamento incluem o uso de giroscópios, sendo que tanto luvas quanto giroscópios podem ser acoplados ao usuário para capturar dados de posição, movimento e flexão (Sarma; Bhuyan, 2021).

Apesar da precisão oferecida na captura das configurações e movimentos das mãos, os dispositivos wearables, especialmente as luvas sensorizadas, apresentam limitações relevantes,

como o alto custo e restrições na interação, que variam de acordo com o modelo e o contexto de uso.

A partir da década de 1990, observa-se o início da transição para métodos baseados em visão computacional. Essa mudança foi impulsionada por avanços no processamento de imagens, pela maior disponibilidade de *hardware* computacional e pela evolução dos algoritmos de detecção. Esse período foi marcado pelo uso de câmeras, tanto simples quanto múltiplas, além de dispositivos mais avançados com captura de profundidade, como o *Kinect*, permitindo a obtenção de imagens e vídeos dos gestos sem necessidade de contato físico com o usuário.

Segundo Sarma e Bhuyan (2021), os estudos dessa fase classificavam o reconhecimento de gestos em categorias como gestos estáticos, associados à ideia de postura, e gestos dinâmicos, baseados em trajetórias, que podiam ser isoladas ou contínuas. Os primeiros sistemas de visão computacional empregavam técnicas como segmentação baseada na cor da pele, detecção de contornos e extração de características geométricas. Os métodos de classificação mais utilizados nesse período incluíam Máquinas de Vetores de Suporte (*Support vector machine*s - *SVM*), *k-Nearest Neighbors* (*k-NN*) e *Hidden Markov Model*s (*HMM*). Entre os principais desafios enfrentados destacavam-se as variações de iluminação, a oclusão, a presença de fundos complexos e ruidosos, bem como dificuldades na segmentação e no rastreamento das mãos.

A terceira e mais recente fase corresponde à era do *DL*, na qual ocorre uma transição da interação humano-computador tradicional para uma interação humano-computador inteligente. Segundo Šumak, Brdnik e Pušnik (2021), os estudos desse período concentram-se majoritariamente no uso de *CNNs*, identificadas como as arquiteturas mais empregadas para o reconhecimento de emoções, expressões faciais e gestos. No entanto, os autores ressaltam a existência de lacunas relacionadas à experiência do usuário e à aceitação dessas tecnologias em contextos fora de ambientes controlados.

Shi *et al.* (2021) destacam as principais arquiteturas utilizadas no *DL* para reconhecimento de gestos, incluindo *CNNs* 3D, redes de dois fluxos (*two-stream networks*), redes recorrentes com memória de curto e longo prazo (*Long Short-Term Memory* - LSTM) e abordagens baseadas em múltiplas modalidades de entrada.

Mais recentemente, arquiteturas baseadas em *Transformers*, originalmente desenvolvidas para o processamento de linguagem natural, passaram a ser adaptadas para

aplicações em visão computacional. Shin *et al.* (2024) apresentam novas arquiteturas representativas dessa fase, como os *Vision Transformers* (*ViT*), *Convolutional Neural Networks with Transformers* (*CMT*) e *GCNs*. Além disso, destaca-se o uso de esqueletos humanos extraídos por meio de *framework*s como *MediaPipe* e *OpenPose*. Os modelos *ViT*, em particular, demonstram capacidade superior para capturar dependências de longo alcance, envolvendo relações espaciais entre diferentes partes do corpo, aspecto fundamental para o reconhecimento de gestos complexos.

2.2.2 Métodos clássicos

Os métodos clássicos de reconhecimento de sinais e gestos baseados em visão computacional podem ser categorizados segundo diferentes estratégias, relacionadas principalmente às etapas de processamento de imagem e de extração de características.

Entre essas abordagens, destaca-se a segmentação por cor da pele, que se baseia na identificação de regiões correspondentes à pele humana. Essa técnica parte da premissa de que *pixel*s relacionados à pele ocupam regiões relativamente compactas em determinados espaços de cor, permitindo sua separação do restante da imagem. Diversos espaços de cor têm sido explorados para esse fim. A utilização do espaço *RGB*, embora intuitiva, apresenta alta sensibilidade a variações de iluminação (Oudah; Al-Naji; Chahl, 2020). Com o objetivo de contornar essa limitação, Qi, Xu e Ding (2022) propõem um modelo de detecção de pele baseado em Mistura Gaussiana (GMM) *CbCr-I*, que combina os espaços de cor *YCbCr* e *YIQ*, associado a um *threshold* adaptativo. Essa abordagem gera uma representação da distribuição da forma da mão, a partir da qual são calculados o contorno e o centróide. As distâncias entre o centróide e os pontos do contorno são então agrupadas segundo ângulo e raio, formando uma matriz descritiva da forma da mão, o que torna o modelo mais robusto às variações de iluminação.

Apesar de sua simplicidade, os métodos baseados em segmentação por cor enfrentam desafios relevantes, como a variação de iluminação incidente sobre a pele, que altera a percepção cromática durante o treinamento, a presença de objetos com cores semelhantes à da pele, que pode gerar falsos positivos (*FP*), e a diversidade de tons de pele associada a diferentes etnias. Além disso, condições de iluminação intensa ou inadequada podem reduzir

significativamente a acurácia do sistema, exigindo o treinamento de modelos adaptados a cenários específicos.

Outra categoria relevante é a dos métodos baseados em contorno e forma, que utilizam a extração da silhueta da mão e a análise de suas propriedades geométricas. Essas técnicas geralmente envolvem a detecção de bordas, seguida da análise do contorno e da extração de descritores de forma. Segundo Qi *et al.* (2024), as características geométricas extraídas incluem o número de dedos estendidos, os ângulos entre os dedos, a razão entre largura e altura da mão, momentos invariantes e descritores de Fourier do contorno. Esses descritores apresentam baixa sensibilidade a transformações como rotação, translação e escala, contribuindo para maior robustez do modelo.

Esses métodos eram comumente combinados com arquiteturas de classificação tradicionais, como *k-NN*, *Support vector machine*s (SVM) e *Random forest*, utilizando características extraídas manualmente. Embora apresentem boa eficiência computacional, tais abordagens demonstram limitações em cenários com oclusão parcial, fundos complexos ou variações significativas na aparência da mão, como diferenças de tamanho ou formato, o que compromete a precisão da detecção (Kareem Murad; H. Hassin Alasadi, 2024).

Os métodos baseados em movimento e fluxo óptico foram amplamente empregados no reconhecimento de gestos dinâmicos, nos quais o movimento ao longo do tempo é um fator determinante. Nessa abordagem, estima-se o padrão de movimento aparente dos objetos *frame* a *frame*, sendo comum o uso de modelos como os *HMM*, que permitem capturar a natureza sequencial dos gestos dinâmicos por meio da modelagem de transições entre estados ocultos que representam diferentes fases do gesto. Contudo, esse tipo de modelo requer cuidadosa definição das características e da estrutura dos estados, além de apresentar sensibilidade à velocidade de execução dos gestos, o que dificulta sua generalização para múltiplos usuários (Qi et al., 2024).

Por fim, o uso de sensores de profundidade, como o *Microsoft Kinect* e o *Intel RealSense*, representou um avanço significativo na representação tridimensional dos gestos utilizados no treinamento dos modelos. Segundo Qi *et al.* (2024), esses dispositivos possibilitam a obtenção de *depth maps* combinados a imagens *RGB*, permitindo uma segmentação mais robusta da mão do gesticulador e reduzindo os efeitos provocados por variações de iluminação e complexidade do fundo. A partir dessa evolução, surgiram também os métodos baseados em esqueletos 3D, que estimam as posições articulares da mão, utilizando

essas informações para a construção de vetores de características empregados na classificação dos gestos (Oudah; Al-Naji; Chahl, 2020).

2.2.3 Deep learning para reconhecimento de gestos

O *DL* transforma o campo de reconhecimento de gestos, substituindo engenharia manual de características por aprendizado *end-to-end* de representações hierárquicas (Wu, 2024; Zhang; Jiang, 2024).

Segundo Zhang e Jiang (2024), as *CNNs* tornaram-se a arquitetura dominante da área de pesquisa para reconhecimento de gestos estáticos, tipos como *AlexNet*, *VGGNet*, *ResNet* e *Inception*, que originalmente foram desenvolvidas para a classificação dos objetos em larga escala sofreram adaptações para serem utilizadas em reconhecimento de sinais. As *CNNs* pré-treinadas em *ImageNet* e fine-tunadas nos *dataset*s em língua de sinais alcançam um desempenho superior aos métodos baseados nas características manuais. O *transfer learning* é tecnologia que permite aproveitar representações visuais aprendidas em grandes *dataset*s mesmo quando os dados de sinais são limitados (Wu, 2024).

Já as arquiteturas recorrentes e temporais utilizadas em gestos dinâmicos e reconhecimento sequenciais de gesto tornaram-se essenciais para o desenvolvimento na área, o *LSTM* e *Gated Recurrent Units* (*GRU*) permitem criar modelos com sequências temporais de vídeo (Jiang et al., 2022) e (Zhang; Jiang, 2024), por outro lado as abordagens híbridas combinando o *CNNs* para extração de características espaciais e o uso de *LSTMs* para modelagem temporal demonstraram resultados superiores ao uso isolado (Ismail; Dawwd; Ali, 2022), pois o *CNN* fica responsável por processar cada *frame* individualmente para extrair representações visuais, que são então alimentadas a uma *LSTM* para capturar dinâmica temporal (Jiang et al., 2022).

As *3D CNNs* têm a vantagem de estender a coleta de informação para a dimensão temporal, ou seja, elas processam a parte de informações de espaço-temporais diretamente, permitindo o aprendizado das características espaciais e temporais juntas (Zhang; Jiang, 2024), eliminando a necessidade de arquiteturas separadas para cada modalidade como é o caso de muitos trabalhos que fazem um *pipeline* conjunto de *CNN* + *LSTM*, esses métodos embora pareçam ter uma vantagem inicial grande, eles demandam alta complexidade computacional,

pois envolvem muitos parâmetros tornando o treinamento lento e exigindo *hardware* especializado, trabalhos mais novos propuseram o uso de *3D CNNs lightweight* combinadas com *Transformers* para balancear o desempenho e eficiência (Zhang; Jiang, 2024).

Já o *ViT*, que é uma adaptação de *Transformer* para a visão computacional, diferentemente das *CNNs* que processam as informações da extração localmente através de filtros convolucionais, *Transformer* utiliza extração por *self-attention* que captura as dependências globais na imagem (Aly; Fathi, 2025), para o uso em reconhecimento, a arquitetura *Transformer* apresenta algumas vantagens, como a capacidade de modelar relações espaciais entre mãos, face e corpo, processamento de sequências de comprimento variável e paralelização durante treinamento (Zhang; Jiang, 2024). Em 2023, foi desenvolvido um modelo baseado em *Transformer* para reconhecimento de *Korean Sign Language* (*KSL*), combinando branches de *CNN* e *Transformer* para aproveitar tanto características locais quanto dependências globais, alcançando 89% de acurácia em um *dataset* de 77 classes (Shin et al., 2023).

Os modelos multimodais e fusão de features, combinam diferentes tipos de entrada, podendo ser *RGB*, profundidade, fluxo óptico, dados de esqueleto, eles demonstram desempenho superior pois a fusão de informações complementares aumenta robustez e precisão (Ismail; Dawwd; Ali, 2022), as estratégias de fusão incluem: fusão em nível de features (concatenação de representações aprendidas), fusão em nível de decisão (combinação de predições de múltiplos modelos), e fusão em nível de arquitetura (redes *multi-stream* que processam diferentes modalidades em paralelo) (Ismail; Dawwd; Ali, 2022).

2.2.4 Desafios no reconhecimento de sinais

Apesar dos avanços significativos na área, o reconhecimento de gestos e sinais em cenários não controlados permanece desafiador, pois existem diversos desafios, no processo de criação de modelo e detecção da imagem (Chakraborty *et al.*, 2018) e (Pisharady; Saerbeck, 2015).

A mudança nas condições de iluminação é um dos problemas que levam a *FP* na detecção, pois a intensidade, direção, cor da fonte luminosa afetam diretamente a captura do gesto, pois para o modelo pode estar se mudando a aparência visual das mãos e dos gestos,

métodos que utilizam a cor de pele são vulneráveis nesse sentido, pois no momento da captura a iluminação não uniforme cria sombras e reflexos especulares que podem ser de forma errada interpretados como parte do gesto, ou podem gerar ocultação da região das mãos, quando pensamos no ambiente externo essa variações são ainda maiores devido a mudanças durante o dia e as condições climáticas (Chakraborty *et al.*, 2018), algumas soluções para este tipo de acontecimento são a normalização de histogramas, uso de espaços de cor que separam luminância de cromaticidade, *data augmentation* com variações de brilho e contraste durante treinamento, o uso de sensores de profundidade que não sofrem variação com à iluminação visível (Pisharady; Saerbeck, 2015) e (Sarowar *et al.*, 2025).

A oclusão e auto-oclusão é um dos desafios mais críticos, pois quando ocorre a cobertura de dedos ou partes da mão se sobrepõem ou parte de objetos a obstruem parcialmente a captura de gesto não é feita de maneira correta (Chakraborty *et al.*, 2018), no caso da auto-oclusão ela muitas vezes ocorre, pois, a mão humana possui mais de 20 graus de liberdade, tendo configurações altamente articuladas onde acontece naturalmente a cobertura (Erol *et al.*, 2007). Para visão monocular, onde existe somente o uso de um sensor, ou seja, uma única câmera para captura e análise de imagens, e estimativa completa da pose da mão permanece um problema ainda não resolvido (Erol *et al.*, 2007). Algumas abordagens que podem ser usadas para mitigar esse problema são o de uso de múltiplas câmeras com diferentes pontos de captura, modelos 3D generativos da mão, e redes neurais com mecanismos de atenção que podem "inferir" partes ocultas baseando-se em contexto visível (Sarowar *et al.*, 2025).

O *cluttered* é outro problema que acontece, e, depende de fatores como a existência da alta densidade de objetos, ou seja, muitos objetos próximos uns dos outros, a própria oclusão, podendo ser voluntaria ou involuntária, escondendo partes importantes dos gestos, o fundo não homogêneo como os do treinamento, podendo haver texturas, cores e formas que se assemelham aos objetos em primeiro plano, e variação de escala e orientação, onde o objeto alvo parece em tamanhos e ângulos muito variados devido à perspectiva (Chakraborty *et al.*, 2018). Todos esses fatores podem confundir algoritmos de detecção.

Os métodos clássicos que fazem a subtração de fundo assumem cenários estáticos, falhando em ambientes dinâmicos onde câmera ou elementos do fundo se movem (Chakraborty *et al.*, 2018), nesse sentido soluções de *DL* demonstram maior adaptação por meio do aprendizado de características discriminativas (Rastgoo; Kiani; Escalera, 2021), mas ainda enfrenta dificuldades dependendo da textura ou cor do fundo são muito similares ao alvo que é a mão do sinalizador.

Estratégias de mitigação incluem uso de fundos artificiais controlados, sensores de profundidade para segmentação 3D (Pisharady; Saerbeck, 2015), e *data augmentation* com variações de fundo durante treinamento para aumentar invariância (Rastgoo; Kiani; Escalera, 2021).

Existem as variabilidades intraclasse e similaridade interclasse, a variabilidade intraclasse acontece quando os gestos apresentam alta variabilidade, seja ele executado pela mesma pessoa ou por um terceiro, acontecem variações de velocidade, amplitude, orientação, precisão, e deve-se considerar os fatores antropométricos relacionados ao tamanho da mão e comprimento dos dedos (Chakraborty *et al.*, 2018). Já os detalhes interclasses ocorrem quando a diferença em detalhes sutis como OM ou número de dedos estendidos durante a repetição de um mesmo gesto (Rastgoo; Kiani; Escalera, 2021). A mitigação é feita utilizando as técnicas de regularização, *data augmentation* extensivo, e uso de *dataset*s grandes e diversos para melhorar generalização e distinguibilidade, pode-se também o usar o *transfer learning* e *domain adaptation* que permite adaptar os modelos treinados em *dataset*s grandes para contextos específicos e com dados limitados (Rastgoo; Kiani; Escalera, 2021).

Reconhecimento em tempo real e eficiência computacional constituem desafios importantes, pois muitas aplicações de reconhecimento de gestos, desde interfaces para PCD, controles de robótica, e realidade aumentada demandam o processamento em tempo real com latência mínima, porém modelos de *DL* e *Transformer*s com muitas camadas, apresentam alta complexidade computacional, nesse momento deve ser pensado no *trade-off* entre a acurácia e eficiência computacional, os modelos mais robustos e com maior profundidade normalmente performam melhor, mas necessitam de *hardware* especializado como *GPUs* de alta performance, porém apresentam problemas no tempo de inferência quando se tenta fazer a aquisição e reconhecimento de dados em dispositivos com recursos limitados (Sarowar *et al.*, 2025).

Pesquisas em *network pruning* que focam em reduzir o tamanho e a complexidade da Rede Neural já treinada, através da remoção de parâmetros considerados redundantes ou pouco importantes para a performance final do modelo, já a quantização é o processo de redução e precisão numérica dos pesos e das ativações da rede neural, uma vez que os modelos normalmente são treinados usando números de ponto flutuante de 32 bits e o processo proposto remapeia esses números para um formato inteiro de 8 bits, consequentemente reduzindo em média 4 vezes o tamanho do modelo, o *knowledge distillation* é uma técnica de compressão de modelos onde um modelo *Student Network* de baixa capacidade adquire o poder de

generalização de um modelo *Teacher Network Oversized*, sendo que o treinamento do *Student Network* é supervisionado, não apenas pelas *hard labels* mas também pelas distribuições de probabilidade suavizadas do *Teacher Network Oversized*, permitindo que o modelo eficiente *Student Network* mantenha uma acurácia comparável à do *Teacher Network Oversized*, reduzindo a latência e o consumo computacional de recursos, arquiteturas eficientes como o *MobileNets* usam a decomposição de uma convolução padrão em duas etapas, a *Depthwise Convolution* feita a aplicação de 1 filtro por canal, seguida de uma *Pointwise Convolution* 1x1, diminuindo os parâmetros e *FLOPs* de 8 a 9 vezes em comparação com *CNNs* densas (Sarowar *et al.*, 2025) como o *VGG* e *ResNet*, já o *EfficientNets* estabelecem um equilíbrio usando o *Compound Scaling* por meio de uma fórmula de escalonamento uniforme para ajuste de três dimensões da rede simultaneamente: a profundidade (α), a largura (β) e resolução (γ) de entrada.

A dependência de dados anotados é um gargalo do *DL*, pois ele depende de um grandes volumes desses dados e os mesmos necessitam ter alta qualidade, sendo um trabalho de significativo esforço manual e demorado que exige conhecimento sobre o que se quer anotar (Rastgoo; Kiani; Escalera, 2021), outro problema é que os *dataset*s públicos de língua de sinais são limitados em tamanho e vocabulário (Chakraborty *et al.*, 2018), os voltados para Libras são ainda mais escassos, um caminho que os autores usam para mitigar esse risco são o uso de técnicas de *augmentation*, aprendizado semi-supervisionado, *self-supervised learning* e síntese de dados através de modelos generativos (Sarowar *et al.*, 2025).

2.3 Detecção de objetos e *Transformer*

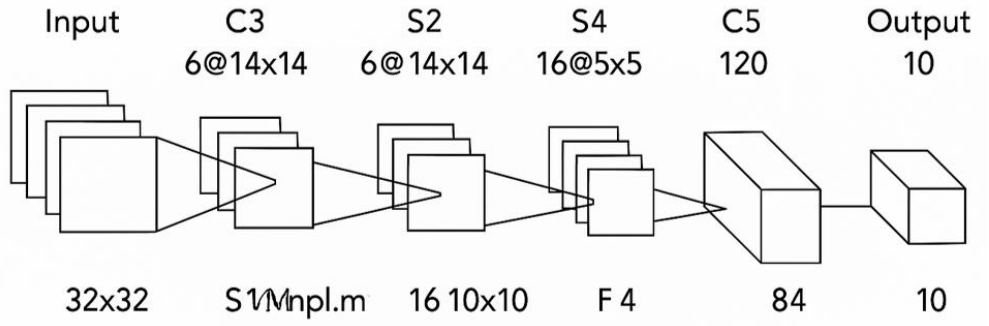
A detecção de objetos constitui uma das tarefas fundamentais da visão computacional, cujo objetivo é identificar categorias de objetos presentes em uma imagem e, simultaneamente, localizar suas posições por meio de *bounding boxes*. Essa tarefa é muito importante para diversas aplicações, como vigilância, veículos autônomos, sistemas de apoio à decisão e reconhecimento de gestos. Esta subseção apresenta a evolução das arquiteturas de detecção de objetos, desde as abordagens baseadas em *CNNs* até as propostas mais recentes fundamentadas em *Transformer*.

2.3.1 Redes Neurais Convolucionais (CNNs) tradicionais

As *CNNs* consolidaram-se na área de visão computacional a partir de 2012, com a utilização da arquitetura *AlexNet* no *ImageNet Large-Scale Visual Recognition Challenge*. A partir desse marco, arquiteturas como *VGGNet*, *ResNet* e *Inception* estabeleceram novas bases para a extração hierárquica de características visuais, impulsionando avanços significativos em tarefas de classificação e detecção de objetos.

As *CNNs* operam por meio de camadas convolucionais que aplicam filtros sobre regiões locais da imagem, seguidas por operações de pooling para redução da dimensionalidade e camadas totalmente conectadas responsáveis pela classificação final. Essa arquitetura se fundamenta em três propriedades-chave: a localidade, viabilizada pelo uso de filtros que processam regiões locais da imagem; o compartilhamento de parâmetros, no qual os mesmos filtros são aplicados em diferentes posições; e a invariância translacional, que permite a detecção de padrões independentemente de sua posição espacial (Simonyan; Zisserman, 2015; He *et al.*, 2016; Redmon *et al.*, 2016).

Figura 4: Processo Convolucional.



Fonte: Dados da pesquisa (2025).

A *VGGNet*, proposta por Simonyan e Zisserman (2015), demonstra que redes profundas compostas por filtros de pequeno tamanho (3×3), dispostos em sequência, podem apresentar desempenho superior em relação a arquiteturas que utilizam filtros maiores. Nesse contexto, as arquiteturas *VGG-16*, com 16 camadas, e *VGG-19*, com 19 camadas, tornaram-se *backbones* amplamente utilizados em diversas tarefas de visão computacional.

A *ResNet*, introduzida por He *et al.* (2016), representou um avanço ao abordar o problema de degradação em redes muito profundas por meio do uso de conexões residuais (*skip connections*). Esse mecanismo possibilita o treinamento efetivo de redes com centenas de

camadas, fazendo com que arquiteturas como a *ResNet-50* e a *ResNet-101* se consolidassem como padrões em aplicações de detecção de objetos.

Apesar da elevada eficácia das *CNNs* na classificação de imagens, sua aplicação à detecção de objetos apresenta desafios importantes. Entre eles, destacam-se a necessidade de processar múltiplas regiões candidatas de forma independente, a dificuldade em capturar dependências globais devido ao campo receptivo limitado das convoluções, e a dependência de componentes manuais adicionais, como a geração de propostas de regiões (*region proposals*) e a supressão de não-máximos (*Non-Maximum Suppression – NMS*), etapas que aumentam a complexidade do *pipeline* de detecção (Ren et al., 2017).

2.3.2 Arquiteturas clássicas de detecção

Nesta sub-subseção, são discutidas duas arquiteturas clássicas de detecção de objetos que serão utilizadas como referência comparativa em relação à proposta apresentada nesta dissertação de mestrado.

2.3.2.1 R-CNN e Faster R-CNN

A arquitetura *Region-based Convolutional Neural Network* (*R-CNN*) é considerada um marco na evolução dos sistemas de detecção de objetos, por ter introduzido avanços significativos em termos de acurácia e eficiência. A arquitetura foi apresentada no trabalho *Rich Feature Hierarchies for Accurate Object detection and Semantic Segmentation*, de Girshick *et al.* (2014). Seu funcionamento baseia-se na geração de aproximadamente 2.000 propostas de regiões por imagem por meio do algoritmo *Selective Search*. Cada proposta é então processada individualmente por uma *CNN* pré-treinada, como *AlexNet* ou *VGGNet*, para extração de características, seguida da classificação por classificadores *SVM*. Posteriormente, é realizado o refinamento das *bounding boxes* por meio de regressão linear.

Essa arquitetura alcançou um *mean average precision* (*mAP*) de 53,3% no *Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes* (*PASCAL*

VOC) 2012, conjunto composto por 11.530 imagens anotadas em 20 classes (Girshick et al., 2014). Apesar do desempenho expressivo para a época, o *R-CNN* apresentava limitações relevantes, como o processamento independente de cada proposta de região, o que resultava em computação redundante significativa, tempo elevado de treinamento e inferência, em torno de 47 segundos por imagem utilizando *GPU*, e grande demanda por armazenamento em disco para os vetores de características extraídos (Girshick, 2015).

Visando superar essas limitações, Girshick (2015) propôs a arquitetura *Fast R-CNN*. As principais inovações introduzidas foram a utilização da camada *Region of Interest Pooling* (*RoI Pooling*), que permite extrair características de tamanho fixo diretamente do mapa de características convolucionais, eliminando a necessidade de processar cada proposta de região individualmente por toda a *CNN*. Além disso, o treinamento passou a ser realizado de forma *multitask*, unificando a classificação das regiões e a regressão das *bounding boxes* em uma única rede treinada *end-to-end*, por meio da combinação das funções de perda de classificação (*classification loss*) e de localização (*localization loss*). Nesse modelo, a imagem é processada apenas uma vez pelas camadas convolucionais, com o compartilhamento das características entre todas as regiões de interesse.

Como resultado dessas alterações, o *Fast R-CNN* apresentou melhorias expressivas em relação ao *R-CNN*, destacando-se:

- treinamento da *VGG-16* aproximadamente nove vezes mais rápido;
- tempo de inferência cerca de 213 vezes menor;
- aumento do valor de *mAP* no conjunto *PASCAL VOC 2012*.

A evolução seguinte foi a arquitetura *Faster R-CNN*, apresentada por Ren *et al.* (2017), que introduziu a *Region proposal network* (*RPN*). Essa rede compartilha as camadas convolucionais com o detector, sendo responsável por gerar automaticamente as propostas de regiões, prevendo simultaneamente *objectness scores*, que indicam a probabilidade de uma região conter um objeto, independentemente da classe, e as coordenadas das *bounding boxes*, representadas pelos parâmetros x, y, w e h. A *RPN* utiliza caixas de referência (*anchors*) com múltiplas escalas e razões de aspecto pré-definidas em cada posição espacial do mapa de características. Na configuração padrão, são utilizados nove *anchors* por localização, resultantes da combinação de três escalas e três razões de aspecto, permitindo a detecção de objetos de diferentes tamanhos e formatos sem a necessidade de pirâmides de imagens.

A arquitetura *Faster R-CNN* possibilita o treinamento unificado da *RPN* e da rede de detecção por meio de *alternating optimization* ou *approximate joint training*, promovendo o compartilhamento eficiente das camadas convolucionais. Em testes experimentais, o *Faster R-CNN* alcançou desempenho de:

- *mAP* de 78,8% no conjunto *PASCAL VOC 2007*;
- tempo de inferência aproximado de 5 *frames* por segundo (*FPS*) utilizando *GPU*;
- desempenho cerca de dez vezes mais rápido que o *Fast R-CNN* baseado em *Selective Search*.

Dessa forma, o *Faster R-CNN* consolidou o paradigma dos *detector*es em dois estágios (*two-stage detector*s), que dominou a área de detecção de objetos por vários anos, servindo de referência para inúmeras arquiteturas subsequentes.

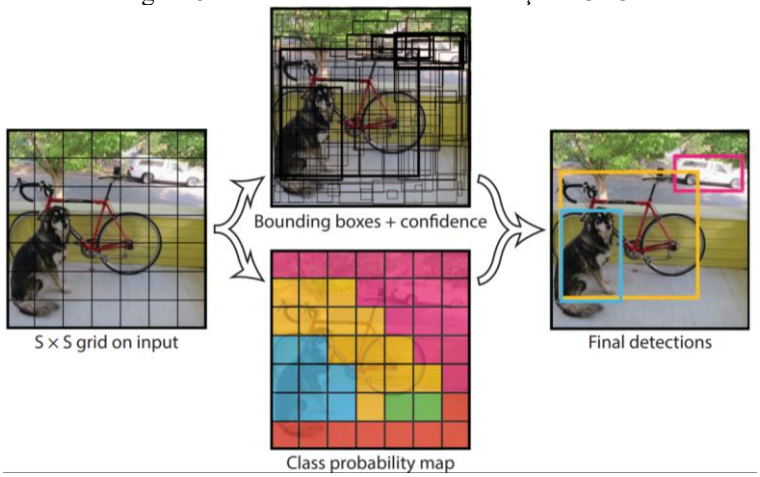
2.3.2.2 YOLO (You Only Look Once)

Enquanto as arquiteturas baseadas em *R-CNN* priorizam a acurácia por meio de múltiplos estágios de processamento, Redmon *et al.* (2016) propuseram uma abordagem alternativa denominada *YOLO* (*You Only Look Once*), que trata a detecção de objetos como um problema de regressão em estágio único (*one-stage*). Nessa abordagem, ao invés de gerar propostas de regiões e classificá-las de forma sequencial, a imagem de entrada é dividida em uma grade $S \times S$, na qual cada célula é responsável por prever diretamente *bounding boxes* e probabilidades de classe em uma única passagem pela rede, conforme ilustrado na Figura 5.

Na sua versão original, a arquitetura *YOLO* é composta por 24 camadas convolucionais seguidas de 2 camadas totalmente conectadas, inspiradas na arquitetura *GoogLeNet*. A imagem de entrada é redimensionada para 448×448 *pixel*s, sendo então dividida em uma grade $S \times S$. Cada célula da grade prediz B *bounding boxes* e C probabilidades de classe, resultando em um tensor de saída de dimensão $S \times S \times (B \times 5 + C)$.

Cada *bounding box* contém as coordenadas do centro (x, y), normalizadas em relação à célula da grade, a largura w e altura h, normalizadas em relação à imagem completa, além de uma *confidence score*, que expressa a probabilidade de a caixa conter um objeto (Redmon et al., 2016).

Figura 5: Sistema do modelo de detecção *YOLO*.



Fonte: Imagem retirada da página 780 do artigo *You Only Look Once*: *Unified*, *Real-Time Object detection* (2016).

De acordo com Redmon *et al.* (2016), uma das principais vantagens da arquitetura *YOLO* está em sua elevada eficiência computacional. Na configuração padrão, o modelo é capaz de processar aproximadamente 45 *FPS*, enquanto a versão *Fast YOLO* atinge até 155 *FPS*, tornando-a adequada para aplicações em tempo real, como veículos autônomos e sistemas de vigilância. Além disso, ao processar a imagem completa, o *YOLO* captura informações contextuais globais, reduzindo a incidência de *FP* em regiões de fundo quando comparado a detectores baseados em propostas de região. A arquitetura também demonstra boa capacidade de generalização, apresentando desempenho consistente no *transfer learning* de imagens naturais para outros domínios, como obras de arte, a exemplo dos conjuntos *Picasso Dataset* e *People-Art Dataset*.

Em contrapartida, a arquitetura *YOLO* apresenta limitações, especialmente em relação à precisão de localização quando comparada a arquiteturas como o *Faster R-CNN*. Essas limitações tornam-se mais evidentes em cenários que envolvem objetos de pequenas dimensões ou objetos muito próximos entre si. Isso ocorre devido à restrição imposta pelo esquema de grade fixa, no qual cada célula é responsável por prever apenas um número limitado de *bounding boxes*, dificultando a detecção de múltiplos objetos pequenos concentrados em uma mesma região espacial.

A arquitetura *YOLO* passou por diversas evoluções ao longo dos anos. A *YOLOv2*, também conhecida como *YOLO9000*, introduziu melhorias como o uso de *batch normalization*, *anchor boxes*, treinamento em múltiplas resoluções e a capacidade de detectar aproximadamente 9.000 classes de objetos (Redmon; Farhadi, 2017). A *YOLOv3* incorporou o

backbone Darknet-53 e passou a realizar predições em múltiplas escalas, aprimorando a detecção de objetos pequenos (Redmon; Farhadi, 2018).

Versões subsequentes, como *YOLOv4*, *YOLOv5*, *YOLOv8* e *YOLOv11*, introduziram avanços significativos, dentre os quais se destaca o uso do *Cross Stage Partial Networks* (*CSPDarknet*), que integra o conceito de particionamento do mapa de características. Nesse esquema, o mapa é dividido em duas partes: uma percorre os blocos convolucionais, enquanto a outra contorna esses blocos e é concatenada diretamente à saída. Esse mecanismo reduz a redundância no cálculo de *gradient*es, tornando a rede mais leve, rápida e eficiente, possibilitando altas taxas de *FPS* mesmo em *hardware* não especializado.

Além disso, essas versões incorporaram técnicas avançadas de *data augmentation*, que vão além de simples transformações geométricas, incluindo estratégias como mistura de informações de múltiplas imagens e alterações na estrutura semântica dos dados, com o objetivo de mitigar o *overfitting*. Outra tendência recente é a adoção de arquiteturas *NMS-free*, que produzem diretamente as detecções finais, eliminando a necessidade do pós-processamento por *NMS*. Enquanto detectores tradicionais geram milhares de *bounding boxes* redundantes que precisam ser filtradas, abordagens mais recentes como o *DETR* e versões avançadas do *YOLO*, a exemplo do *YOLOv10* utilizam mecanismos de atenção ou estratégias de treinamento baseadas em atribuição *one-to-one matching*, mantendo o princípio de detecção em estágio único com foco em velocidade.

2.3.3 Limitações das abordagens tradicionais

As arquiteturas clássicas de detecção de objetos baseadas em *CNNs* apresentam limitações estruturais em seus fundamentos, o que motivou o desenvolvimento de abordagens alternativas na área. Entre os principais fatores que impulsionaram essa evolução destaca-se a forte dependência de componentes manuais e heurísticos. Arquiteturas como *Faster R-CNN* e *YOLO* utilizam *anchor boxes* definidas manualmente, que exigem uma escolha cuidadosa de escalas e razões de aspecto, frequentemente determinadas por meio de técnicas de *clustering* em *dataset*s específicos. Além disso, o processo de *NMS* constitui um pós-processamento heurístico utilizado para eliminar detecções duplicadas. Esses componentes introduzem hiperparâmetros sensíveis ao domínio de aplicação, dificultando a generalização para novos

cenários e demandando conhecimento especializado para o *fine-tuning* dos modelos (Redmon; Farhadi, 2017; Ren *et al.*, 2017).

Outro fator limitante refere-se ao modo como as *CNNs* processam informações espaciais. As características são extraídas predominantemente a partir de informações locais, por meio de filtros convolucionais com campos receptivos restritos. Ainda que arquiteturas profundas ampliem progressivamente esse campo receptivo, a captura de dependências de longo alcance ocorre de forma limitada. Em tarefas que exigem a compreensão de contextos globais ou a detecção de objetos de grande porte, essa característica pode se tornar restritiva. Estratégias como *skip connections*, empregadas em arquiteturas como *ResNet* e *DenseNet*, contribuem para melhorar o fluxo de gradientes durante o treinamento, mas não eliminam completamente as limitações relacionadas à modelagem de dependências globais (He et al., 2016).

No contexto específico do reconhecimento de gestos, a limitação das abordagens tradicionais torna-se ainda mais evidente. Essas tarefas demandam a compreensão das relações espaciais entre múltiplas partes do corpo, como mãos, face e tronco. Entretanto, detectores baseados em *CNNs* tendem a tratar essas partes de forma independente, capturando relações contextuais apenas de maneira indireta, por meio do compartilhamento de características nas camadas convolucionais, o que pode comprometer a interpretação global do gesto.

Além disso, arquiteturas do tipo *two-stage*, como o *Faster R-CNN*, apresentam ineficiências computacionais inerentes ao seu funcionamento. Essas arquiteturas exigem o processamento de centenas de propostas de regiões pela rede de detecção. Embora a *RPN* compartilhe as características convolucionais iniciais, as etapas subsequentes de classificação e refinamento das *bounding boxes* introduzem sobrecarga computacional significativa, o que pode ser um fator limitante em aplicações que demandam respostas em tempo real (Ren et al., 2017).

Outro aspecto relevante diz respeito à rigidez arquitetural das *CNNs* empregadas em sistemas de detecção. Essas arquiteturas são geralmente projetadas com número fixo de *anchors* e estruturas específicas de camadas. Dessa forma, a adaptação do modelo para novas tarefas ou domínios costuma exigir a reestruturação de partes significativas da arquitetura, tornando o processo pouco flexível e custoso em termos de desenvolvimento (Redmon; Farhadi, 2017).

Por fim, o desempenho dos detectores baseados em *CNNs* apresentam elevada sensibilidade às escolhas de hiperparâmetros, como *learning rate*, *weight decay*, configurações

das *anchors*, *thresholds* de *NMS* e estratégias de *data augmentation*. O ajuste desses hiperparâmetros é um processo trabalhoso, dependente do *dataset* e, frequentemente, específico para cada aplicação. Diante dessas limitações, emergiram arquiteturas alternativas, como aquelas baseadas em *Transformer*, que buscam reduzir a dependência de componentes heurísticos e aprimorar a modelagem de relações globais no processo de detecção de objetos.

2.3.4 Vision Transformers

A arquitetura *Transformer*, proposta por Vaswani *et al.* (2017), representou um avanço significativo no campo do processamento de linguagem natural por meio do mecanismo de *self-attention*. Posteriormente, sua adaptação para o domínio da visão computacional abriu novas possibilidades para tarefas de reconhecimento de imagens e detecção de objetos.

Dosovitskiy *et al.* (2021) demonstraram que a aplicação de *Transformer*s puros, operando diretamente sobre sequências de blocos de imagem, é capaz de alcançar desempenho equivalente ou superior às melhores arquiteturas baseadas em *CNNs*, desde que os modelos sejam pré-treinados em *dataset*s suficientemente grandes. Essa proposta resultou no surgimento do *ViT*, marco importante na aproximação entre visão computacional e arquiteturas originalmente desenvolvidas para linguagem natural.

A arquitetura *ViT* processa imagens seguindo um *pipeline* composto por quatro estágios principais. No primeiro estágio, denominado *patch embedding*, a imagem de entrada de dimensão $H \times W \times C$ é dividida em blocos que não se sobrepõem de tamanho $P \times P$, como resultado tem-se $N = \frac{(H \times W)}{P^2}$ blocos, os quais são projetados linearmente para uma dimensão D por meio de uma camada de embedding. Esse processo consiste na transformação dos valores brutos de *pixel*s de cada bloco em vetores numéricos abstratos que representam características visuais.

No segundo estágio, é aplicado o *positional encoding*. Diferentemente das *CNNs*, a arquitetura *Transformer* não incorpora implicitamente informações espaciais, sendo, portanto, incapaz de inferir a posição relativa dos blocos na imagem original. Para suprir essa limitação, são adicionados *embeddings* posicionais aprendíveis aos vetores de cada bloco, fornecendo ao

modelo informações espaciais necessárias para a compreensão da estrutura geométrica da imagem.

No terceiro estágio, a sequência de blocos enriquecida com os *positional encodings* é processada por *L* camadas do *Transformer Encoder* padrão. Cada camada é composta por um bloco de *Multi-head Self-attention* (*MHSA*), seguido por uma *Multi-Layer Perceptron* (*MLP*), com *Layer normalization* aplicada antes de cada bloco e conexões residuais após cada operação. Essa etapa possibilita a modelagem de relações globais entre todos os blocos da imagem.

Por fim, no quarto estágio, é adicionado à sequência um *token* especial denominado (*CLS*), utilizado como representação global da imagem. Esse *token* é processado por uma camada linear para realizar a tarefa de classificação.

Existem diversas variantes do *ViT*, como *ViT-B*, *ViT-L* e *ViT-H*, inspiradas nas configurações do modelo *Bidirectional Encoder Representations from Transformers* (*BERT*), amplamente utilizado em processamento de linguagem natural. Essas variantes diferenciam-se principalmente pela profundidade da rede (número de camadas), dimensão oculta e número de cabeças de atenção. Em algumas notações, como *ViT-L*/16, o sufixo indica o tamanho dos blocos de imagem utilizados, neste caso, *patches* de 16×16 *pixel*s.

Os modelos baseados em mecanismos de atenção apresentam vantagens relevantes. O *self-attention* permite que cada bloco da imagem se relacione diretamente com todos os demais em uma única operação, capturando dependências de longo alcance sem as limitações impostas pelo campo receptivo local das convoluções. Além disso, os *Transformers* não impõem uma estrutura espacial rígida, tornando-os mais flexíveis para adaptação a diferentes tarefas. A única informação explícita de posição é fornecida pelos *positional encoding*s, o que facilita a generalização do modelo.

Outra vantagem relevante está relacionada à escalabilidade. Dosovitskiy *et al.* (2021) demonstraram que o *ViT* pré-treinado no conjunto *JFT-300M*, composto por aproximadamente 300 milhões de imagens, superou arquiteturas baseadas em *CNNs* em diversos *benchmarks*, alcançando, por exemplo, 88,55% no *ImageNet*, 90,72% no *ImageNet-Real* e 94,55% no *CIFAR-100*. Do ponto de vista da eficiência computacional, os *ViTs* podem apresentar menor número de operações (*FLOPs*) para atingir desempenho comparável ao de arquiteturas convolucionais, conforme observado em abordagens como o *Big Transfer* (*BiT*).

Apesar de suas vantagens, os *ViTs* apresentam limitações importantes. Uma delas refere-se à forte dependência de grandes volumes de dados para treinamento. Diferentemente

das *CNNs*, que incorporam vieses indutivos como localidade e invariância translacional, os *ViTs* possuem menor viés estrutural, o que os torna mais suscetíveis ao *overfitting* quando treinados com conjuntos de dados reduzidos, demandando fases extensas de pré-treinamento.

Outra limitação significativa relaciona-se à complexidade computacional do mecanismo de *self-attention*, cuja complexidade é quadrática, $O(N2)$, em função do número de blocos N. Isso ocorre porque cada bloco precisa se relacionar com todos os demais para capturar o contexto global. Assim, o aumento da resolução da imagem implica crescimento quadrático no tempo de processamento e no consumo de memória, inviabilizando o uso de *ViTs* puros em aplicações que demandam imagens de alta resolução, como exames médicos e imagens de satélite.

Para mitigar essa limitação, surgiram variantes como o *Swin Transformer* e o *Pyramid Vision Transformer* (*PVT*). Essas arquiteturas restringem o cálculo da atenção a janelas locais, reduzindo o custo computacional, e, ao mesmo tempo, constroem uma hierarquia de representações ao combinar progressivamente essas janelas em camadas subsequentes. Dessa forma, torna-se possível preservar a compreensão global da imagem sem incorrer nos elevados custos computacionais do *self-attention* global.

O sucesso do *ViT* em tarefas de classificação motivou sua aplicação em problemas de detecção de objetos, culminando no desenvolvimento de arquiteturas como o *DETR*. Essa arquitetura reformula a detecção como um problema de *matching* bipartido direto entre predições e *ground truth*, eliminando componentes manuais como *anchor boxes* e *NMS* por meio do uso de mecanismos de atenção. O *DETR*, por sua relevância ao presente trabalho, será discutido de forma aprofundada no próximo item desta dissertação.

2.4 DETR

O *DETR*, desenvolvido por Carion *et al.* (2020), constitui um marco na área de detecção de objetos ao introduzir uma abordagem conceitualmente distinta das arquiteturas tradicionais. Diferentemente dos métodos clássicos, o *DETR* passa a tratar a detecção de objetos como um problema de predição direta de conjuntos, eliminando a necessidade de diversos componentes heurísticos utilizados em abordagens convencionais. Nesta subseção, são apresentados a

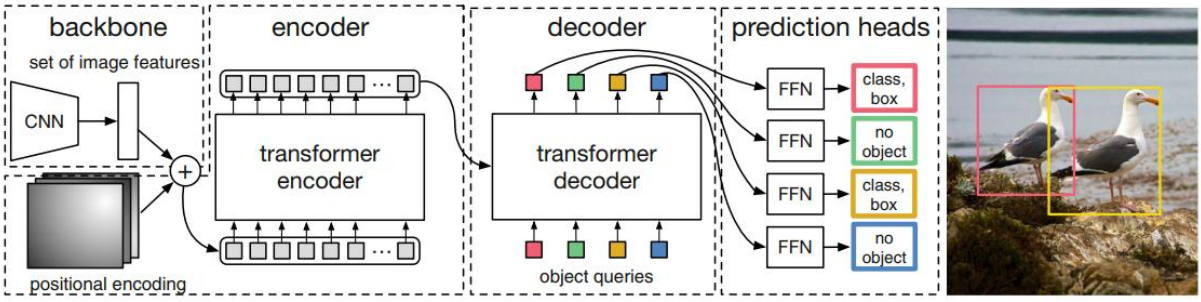
arquitetura original do *DETR*, seus componentes fundamentais e as principais vantagens que o diferenciam dos métodos de detecção de objetos discutidos anteriormente.

2.4.1 Arquitetura original

O *DETR* foi introduzido no artigo seminal "*End-to-end Object detection* with *Transformers*", apresentado na *European Conference on Computer vision* (*ECCV*) em 2020. Sua proposta central consiste em reformular a detecção de objetos como um problema de predição direta de conjuntos, eliminando componentes projetados manualmente que caracterizam os detectores tradicionais, como a geração de *anchors* e a etapa de *NMS* (Carion *et al.*, 2020).

Embora essa proposta represente uma mudança conceitual significativa, a arquitetura do *DETR* é caracterizada por sua simplicidade estrutural, sendo composta por três componentes principais, conforme ilustrado na Figura 2 do artigo original e apresentado na Figura 6. Primeiramente, tem-se o *backbone*, que constitui a espinha dorsal do modelo e corresponde a uma rede neural convolucional, normalmente utilizado o *ResNet-50*, que é responsável por extrair as características visuais da imagem de entrada. Em seguida, utiliza-se a arquitetura *Transformer Encoder-Decoder*, encarregada de processar essas características extraídas e gerar os *embeddings* de objetos. Por fim, as redes *feed forward* totalmente conectadas são responsáveis pela realização das predições finais de classes e coordenadas das *bounding boxes*.

Figura 6: *Pipeline* de detecção de objetos do modelo *DETR*.



Fonte: Carion et al (2020, p. 7).

O *backbone* trabalha a partir da imagem de entrada $x_{img} \in \mathbb{R}^{3\ x\ H_0\ x\ W_0}$ com três canais de cores, lembrando que as de entrada são agrupadas, após é aplicado o preenchimento com

zeros para garantir que todas tenham as mesmas dimensões de H_0 e W_0, que a maior imagem que conte no *batch*. Após o *backbone* cria um mapa de ativação com menor resolução, $f \in \mathbb{R}^{C\ x\ H\ x\ W}$, sendo que os valores típicos utilizados foram de $C = 2048$ e $H = \frac{H_0}{32}$ e $W = \frac{W_0}{32}$. Em seguida é feito primeiro uma camada de convolução reduz o número dos canais C para uma dimensão menor d, com valores de $d = 256$ ou $d = 512$, criando um novo mapa de característica $z_0 \in \mathbb{R}^{d\ x\ H\ x\ W}$. Por fim, o novo mapa é transformado em uma dimensão $d \times HW$, pois é o formato que o *encoder* da arquitetura espera (Carion *et al.*, 2020).

Vaswani *et al.* (2017), explicam que a arquitetura *Transformer* não possui noção intrínseca da posição espacial, dessa forma é necessário colocar uma etiqueta que identifique, esses são os *embeddings* posicionais, dessa forma injetam-se informações sobre a posição relativa ou absoluta dos *tokens* gerados na sequência da entrada na base das pilhas do *encoder* e do *decoder*, o autor ressalta que as codificações posicionais precisam ter a mesma dimensão d_{model}, pois dessa forma pode-se somar as codificações posicionais aos *embeddings*.

Os autores acima propõem utilizar um método baseados em funções seno e cosseno, pois cada dimensão do vetor posicional corresponderá a uma onda de frequência distinta, sendo as funções pares utilizarão a função seno e as dimensões ímpares utilizam a função cosseno, as dimensões dos vetores serão calculadas até o tamanho máximo do d_{model}, essa dimensão calculada será submetida a uma das duas funções, dessa forma o modelo aprende relações relativas entre posições, pois o vetor de *positional encoding* é construído com a mesma dimensão do vetor de embedding, e dessa forma permitindo que o modelo aprenda relações relativas entre posições, pois para qualquer deslocamento fixo k, PE_{pos+k} pode ser representado como uma função linear de PE_{pos}, conforme definidos nas Equações (1) e (2).

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \qquad (1)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \qquad (2)$$

Segundo Vaswani *et al.* (2017), o *encoder* da arquitetura *Transformer* consiste em uma pilha de N camadas iguais, no caso do *DETR* original o número de camadas foi definido como 6, cada uma delas contendo duas subcamadas.

A primeira subcamada é a de *MHSA* que é responsável por permitir que cada posição na sequência de entrada tenha atenção às demais posições, ou seja, aprende relações entre *tokens* na entrada, permitindo que cada *token* possa perceber todos os demais *tokens*, já a segunda camada a de *position wise feed-forward network* (*FFN*) é uma rede totalmente conectada aplicada posição a posição que processa individualmente cada posição de forma não linear, sem misturar informações entre as posições.

Cada uma das subcamadas passa por uma *layer normalization* e a entrada original da subcamada x é somada à sua saída $Sublayer(x)$, conforme apresentado na Equação (3). Segundo He *et al.* (2016) essa estrutura é utilizada das *ResNets* e tem a utilidade de facilitar o fluxo de gradiente no treinamento evitando o *vanishing gradient*, e auxiliar o modelo a aprender transformações mais estáveis.

O *encoder* processa a sequência de características enriquecidas com *positional encoding*, gerando uma representação contextualizada $d \times HW$ (Carion *et al.*,2020), no Diagrama 1, apresenta-se o diagrama anotado do fluxo de dados dentro de uma camada do *encoder* com as duas subcamadas, as setas de *residual connection* e a normalização.

$$Output = LayerNorm(x + Sublayer(x)) \qquad (3)$$

O *decoder* do *DETR* , diferentemente dos *decoder*s autorregressivos em processamento de linguagem natural, opera gerando todas as predições de objetos em paralelo, o *decoder* recebe como entrada um conjunto fixo de N_d *object queries* aprendíveis, normalmente assume-se que $N_d = 6$, sendo que cada camada contém três subcamadas, a primeira é a *multi-head self-attention* sobre as *object queries* que permite a comunicação entre *queries*, fazendo com que o modelo capture as dependências entre potenciais detecções, a segunda é a *multi-head cross-attention*, ou também chamada *encoder-decoder attention* onde as *object queries* interagem com as saídas do *encoder* e aprende seletivamente quais são as mais relevantes da imagem, a terceira FFN, é responsável por transformar individualmente e não linearmente cada *query*, e cada subcamada é seguida por uma conexão residual e uma *layer*

normalization, conforme a estrutura padrão do *Transformer* que também ocorre no *encoder*, no Diagrama 2, apresenta-se o fluxo completo do *decoder* mostrando as *object queries*, as atenções e as cabeças de predição. O conjunto de *object queries* é convencionalmente definido pela Equação (4).

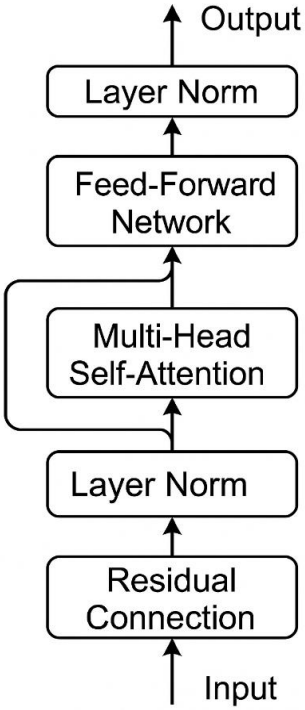
$$Q = \{q_1, q_2, \dots, q_n\} \; onde \; q_i \in \mathbb{R}^d \tag{4}$$

Cada vetor q_i representa uma posição em potencial para a detecção de um objeto, as *object queries* são vetores de embedding aprendidos pelo modelo durante o treinamento, cada vetor assume o papel de um *detector* especializado com foco em diferentes tipos de objetos ou regiões da imagem.

Após o processamento pelo *decoder*, cada um dos N *embeddings* são direcionados de forma independente para duas cabeças de predição, a cabeça de classificação que é uma camada linear que prevê uma distribuição de probabilidade sobre $K + 1$, ou seja a cabeça de classificação é a parte responsável dentro do modelo por informar qual tipo de objeto o modelo encontrou em cada posição (*object query*) ou mesmo se não há objeto, dessa forma transformando todas as saídas do *decoder* em probabilidades para todas as classes possíveis treinadas no modelo.

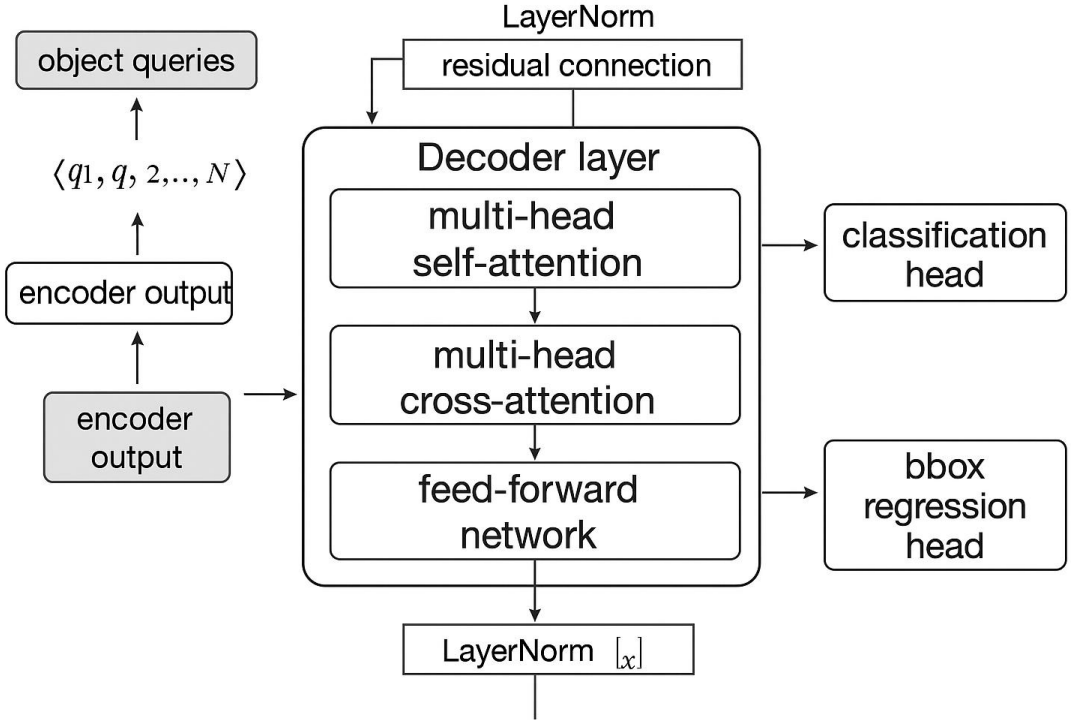
A cabeça de regressão de *bounding box* é uma *MLP* de três camadas que prevê as coordenadas normalizadas da caixa delimitadora, sendo responsável por estimar a posição e o tamanho de cada objeto detectado na imagem. Esse componente é formado por uma rede neural simples, que recebe como entrada o vetor de saída do *decoder*, o qual contém informações sobre o possível objeto, e produz quatro valores que representam as coordenadas da caixa delimitadora ou *bounding box*. Dessa forma, o modelo determina onde o objeto está localizado na imagem e qual é sua dimensão, transformando as informações do *decoder* em coordenadas de uma caixa que envolve o objeto (c_x, c_y, w, h).

Diagrama 1: Fluxo de dados dentro de uma camada do *encoder* com as duas subcamadas.



Fonte: Dados da pesquisa (2025).

Diagrama 2: Fluxo completo do *decoder* mostrando as *object queries*, as atenções e as cabeças de predição.



Fonte: Dados da pesquisa (2025).

Os parâmetros acima do *backbone*, *Positional encoding*, *Transformer Encoder-Decoder*, estão detalhados o que cada um representa na Tabela 1.

Tabela 1: Parâmetros e seus significados.

Símbolo	Significado	Descrição
x_{img}	Imagem de entrada	É a imagem original de entrada para o modelo, com 3 canais (*RGB*)
$\mathbb{R}^{3\ x\ H_0\ x\ W_0}$	Espaço dimensional da imagem	A imagem tem altura H_0, largura W_0 e 3 canais de cor
$f\ \in\ \mathbb{R}^{C\ x\ H\ x\ W}$	Mapa de característica	Saída da *CNN* (*backbone*), que é um mapa de características de menor resolução
C	Número de canais do mapa de característica	Dimensão de canais da saída da *CNN* com valor típico de 2048
H e W	Altura e largura do mapa de ativação	Reduzidos em relação à imagem original $H\ =\ \frac{H_0}{32}$ e $W\ =\ \frac{W_0}{32}$
d	Dimensão de *embedding* do *Transformer*	Redução para uma dimensão menor d, usada pelo *Transformer*
$z_0\ \in\ \mathbb{R}^{d\ x\ H\ x\ W}$	Novo mapa de características	Resultado após a redução de canais usado como entrada do *encoder*
$d \times HW$	Sequência de entrada do *Transformer*	O mapa z_0 é achatado (*flattened*) em uma sequência 1D para o *encoder*
$PE_{(pos,2i)}$, $PE_{(pos,2i+1)}$	*Positional encoding*	Valor do *encoder* posicional para a posição pos na dimensão i. As dimensões pares usam seno e as ímpares usam cosseno
pos	*Position* (posição)	Índice da posição do *token* na sequência (por exemplo, 0 para o primeiro *token*, 1 para o segundo etc.).
i	Dimensão (index da componente)	Indica qual dimensão do vetor de codificação está sendo calculada (entre 0 e $d_{model} - 1$)
d_{model}	Dimensão do modelo (*model* dimension)	Tamanho do vetor de *embeddings* usado pelo *Transformer* $d_{model} = 512$ no modelo base
$10000^{\frac{2i}{d_{model}}}$	Escala de frequência	Controla a frequência das ondas seno e cosseno — cria uma progressão geométrica de comprimentos de onda entre 2πe $10000 \times 2\pi$.
sin, cos	Funções trigonométricas	Geram padrões periódicos usados para representar a posição, seno para dimensões pares, cosseno para ímpares.
c_x	Posição	Posição horizontal do centro da caixa
c_y	Posição	Posição vertical do centro da caixa
w	Dimensão	Largura da caixa
h	Dimensão	Altura da caixa

Fonte: Dados da pesquisa (2025).

2.4.2 Mecanismo de atenção

O mecanismo de atenção constitui o cerne da arquitetura *Transformer* e, consequentemente, do *DETR*. Inicialmente proposto por Vaswani *et al.* (2017), o mecanismo denominado *Scaled Dot-Product Attention* realiza o cálculo de pontuações de similaridade entre

as *queries* (consultas) e as *keys* (chaves), com o objetivo de identificar o grau de relevância entre os diferentes elementos da entrada. Essas pontuações indicam o quanto uma determinada posição do modelo deve "prestar atenção" às demais.

Para evitar que valores excessivamente elevados dessas pontuações prejudiquem o processo de treinamento, os resultados do produto escalar são normalizados pela raiz quadrada da dimensão das *keys*. Em seguida, aplica-se a função *softmax*, responsável por transformar as pontuações normalizadas em pesos de atenção, os quais representam a importância relativa de cada elemento da entrada. Esses pesos são então utilizados para calcular uma média ponderada dos *values* (valores), gerando, assim, uma representação contextualizada da informação.

Em comparação com outros mecanismos de atenção, como o *additive attention*, o *Scaled Dot-Product Attention* apresenta maior eficiência computacional, uma vez que seu cálculo se baseia predominantemente em operações de produto escalar, além de exigir menor custo de memória. Essas características favorecem sua aplicação em arquiteturas baseadas em *Transformers*, como o *DETR*.

O cálculo da atenção é realizado conforme a Equação (5).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (5)$$

Na Equação (5), o termo QK^T representa o cálculo da similaridade entre as *queries* (consultas) e as *keys* (chaves), ou seja, o produto escalar vai medir o quanto cada posição de uma se relaciona com a outra, ou melhor, o quanto um elemento da entrada deve "prestar atenção" a outro, pois quanto maior o valor resultante, maior será a relevância da *key* com relação a sua *query* correspondente.

No entanto, à medida que a dimensão d_k aumenta, os produtos escalares QK^T geram valores mais altos, podendo causar a saturação da função *softmax*, deslocando sua saída para regiões em que os *gradient*es são extremamente pequenos, dificultando o aprendizado do modelo. Dessa forma é necessário normalizar o valor do produto escalar extraindo-se a raiz do valor d_k, gerando pontuações de atenção em uma escala aceitável pelo modelo.

Assim, a função *softmax* transforma os valores obtidos em pesos de atenção normalizados, que indicarão a importância relativa de cada elemento da sequência de entrada,

sendo que esses pesos serão usados para calcular a média ponderada dos valores V (a informação a ser recuperada que contém os dados reais), dessa forma o modelo aprende de forma dinâmica enquanto foca em cada parte da entrada.

O *DETR* usa o mecanismo de *multi-head attention*, o que possibilita o modelo analisar diferentes tipos de relações ao mesmo tempo, ao invés de ter apenas uma forma de atenção, se utiliza várias que funcionam em paralelo.

Cada *head attention* faz o processamento de atenção com as *queries*, *keys* e *values* utilizando parâmetros diferentes para que o modelo aprenda aspectos distintos da informação, ou seja, uma *head attention* aprende sobre o contorno do objeto enquanto outra aprende sobre a textura e outra analisa a posição ou o movimento.

Ao final, o modelo concatena e aplica uma transformação final para combinar tudo que foi aprendido em uma única representação, dessa forma tornando o modelo mais generalizável, pois ele compreende o contexto de vários pontos de vista ao mesmo tempo.

O mecanismo *multi-head attention* é calculado conforme as Equações (6) e (7).

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (6)$$

onde

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (7)$$

Na Equação (6), o termo $MultiHead(Q, K, V)$ representa o mecanismo de atenção com múltiplas *head attention*, as entradas Q (*queries*), K (*keys*) e V (*values*) são usadas para calcular a atenção, cada uma das matrizes é projetada em subespaços diferentes por meio de pesos que são aprendidos durante o treinamento.

Cada *head attention*, $head_i$ realiza uma operação de atenção, $Attention(QW_i^Q, KW_i^K, VW_i^V)$, ou seja, como dito acima cada uma foca em relações diferentes dentro dos dados, sendo que as saídas de todas as *head attention* são concatenadas e transformadas novamente por meio da matriz de projeção W^O, retornando para o espaço de

dimensão original do modelo, onde o h é o número de *head attention* paralelas do modelo a ser treinado.

O *DETR* aplica o uso de *multi-head attention* em dois locais principais do projeto. No *encoder*, a *self-attention* permite que cada posição espacial da imagem capturada atenda a todas as outras posições, dessa forma capturando-se as relações globais e contexto. No *decoder*, o *cross-attention* faz com que as *object queries* atendam às características que foram produzidas pelo *encoder*, focando nas regiões mais relevantes da imagem para cada query, sendo que esse mecanismo é fundamental para a localização dos objetos.

Tabela 2: Ilustração da multi-especialização em *heads* de atenção.

Cabeça	Possível Foco no *DETR*
Head 1	Relações espaciais (proximidade)
Head 2	Características semânticas (classe)
Head 3	Contexto global da cena
Head 4	Contornos e bordas
Head 5	Cores e texturas
Head 6	Relações objeto-objeto

Fonte: Dados da pesquisa (2025).

É necessário enfatizar que a Tabela 2 apresenta um cenário ilustrativo, considerando 6 *heads* de atenção e suas possíveis especializações para fins de compreensão do processo. Na prática, o número de *heads* é determinado pela arquitetura específica do modelo, sendo definido durante a etapa de atribuição de hiperparâmetros. Além disso, a especialização de cada *head* não é pré-programada, mas aprendida de forma orgânica ao longo do treinamento, podendo variar significativamente entre diferentes inicializações do modelo e conjuntos de dados utilizados.

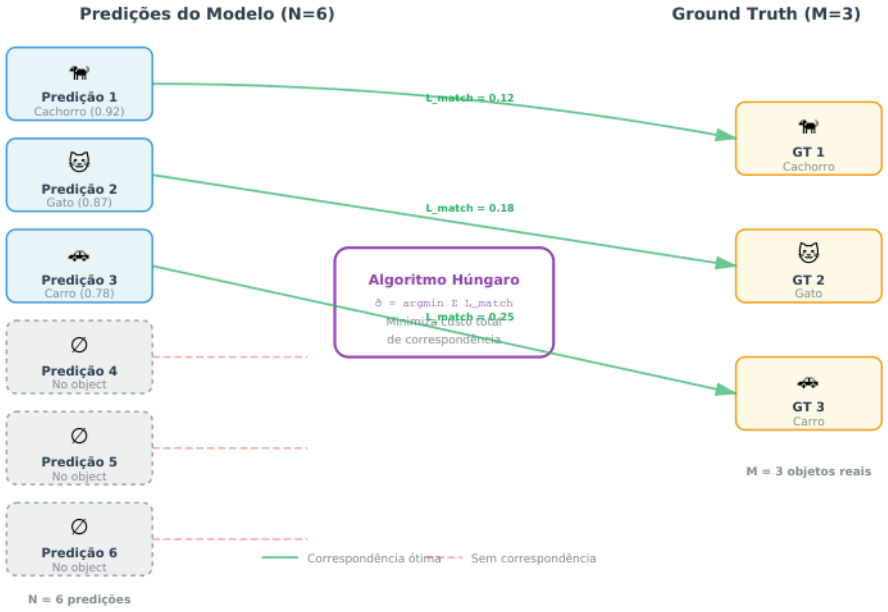
Essa capacidade de capturar relações globais entre as características de uma determinada imagem por meio do mecanismo de atenção distingue o *DETR* das arquiteturas baseadas em *CNN*s, que operam com receptive fields locais e dependem do empilhamento de múltiplas camadas para capturar dependências de longo alcance.

2.4.3 Hungarian matching

Uma das contribuições inovadoras do *DETR* é a utilização do *Hungarian matching* durante o processo de treinamento, pois ele se baseia na função da perda de correspondência bipartida (*bipartite matching loss*), dessa forma permitindo a associação de cada predição do modelo a um objeto real (*ground truth*) sem a necessidade das heurísticas adicionais.

No treinamento, o modelo produz um conjunto fixo de N predições, sendo que esse conjunto normalmente é maior que o número real de objetos M ($M << N$) que estão presentes, assim torna-se necessário estabelecer uma correspondência única (*one-to-one matching*) entre a predição e o que realmente existe no objeto real, nesse modelo as predições que não encontram nenhuma correspondência são associadas à classe *no object* ($\emptyset$), dessa forma garantindo que cada objeto tenha apenas uma predição associada.

Figura 7: Funcionamento do Hungarian *matching*.



Fonte: Dados da pesquisa (2025).

Esse algoritmo, utilizado no *DETR*, é conhecido como *Kuhn-Munkres algorithm*, e é utilizado também em tecnologias de resolução de problemas de associação. Ele trata o problema como uma demanda de otimização combinatória, onde ele encontra a combinação ótima (*optimal matching*) entre predições realizadas e os *ground truth*s, minimizando o custo total de correspondência, por meio da permutação representada por $\hat{\sigma}$ que minimiza o custo global de *matching* na aplicação (Munkres, 1957), conforme expresso na Equação (8).

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{match}\left(yi, \hat{y}_{\sigma\,(i)}\right) \qquad (8)$$

Na Equação (8), $\mathfrak{S}_N$ representa o número total de permutações possíveis dentro do conjunto e a segunda parte $\mathcal{L}_{\text{match}}$é responsável pela parte do cálculo de custo entre pares correspondentes (*pair-wise matching* cost), sendo que o custo é a junção de dois componentes, o primeiro sendo o custo de classificação que é baseado na probabilidade da previsão da classe correta e o segundo é o custo de *bounding box* que é responsável por combinar o erro com a perda generalized *IoU* (*GIoU*), que garante a precisão da localização espacial dos objetos.

Em resumo, o *Hungarian Loss* soma os dois tipos de erro para cada correspondência entre a predição e objeto real, sendo o erro de classificação que mostra se o modelo identificou corretamente o tipo de objeto e o erro de localização que mostra se o modelo desenhou a caixa de forma precisa.

Após, é feita a perda final, que é então computada sobre os pares emparelhados $(yi, \hat{y}_{\sigma\,(i)})$, conforme apresentado na Equação (9).

$$\mathcal{L}_{Hungarian}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}\,(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}_i} \mathcal{L}_{box}\left(b_i, \hat{b}_{\hat{\sigma}}(i)\right) \right] \qquad (9)$$

Em que:

a) N é o número total de predições feitas pelo modelo e cada uma delas será comparada com um objeto real;

b) $\sum_{i=1}^{N}$ é a soma de todos os erros, ou perdas de cada correspondência entre predição realizada e objeto real;

c) $\hat{p}_{\hat{\sigma}(i)}(c_i)$ é a probabilidade prevista para a classe correta c_i depois do emparelhamento feito pelo algoritmo húngaro ($\hat{\sigma}$), ou seja, ele calcula o quanto o modelo acredita que a predição pertence à classe certa;

d) $-\log\, \hat{p}_{\hat{\sigma}(i)}(c_i)$ exprime o *classification loss* penalizando o modelo quando ele dá baixa probabilidade para a classe correta;

e) $\mathbb{1}_{\{c_i \neq \emptyset\}}$ é a função indicadora, atribuindo o valor de 1 se o objeto existe, ou seja, não pertence a classe "*no object*" e atribui 0 se não existe objeto naquela posição, dessa forma o modelo é obrigado somente a calcular a perda de *bounding box* quando há um objeto real.

f) $\mathcal{L}_{box}(b_i, \hat{b}_{\hat{\sigma}(i)})$ expressa o *bounding box loss*, que mede a diferença entre a caixa predita ($\hat{b}_{\hat{\sigma}(i)}$) da caixa real ($b_i$), sendo que normalmente é combinado nesse momento L1 *loss* que é a diferença entre as coordenadas e o *GIoU Loss* que mede a semelhança geométrica entre as caixas.

A utilização do *Hungarian matching* possui diversas vantagens como a não necessidade de *NMS* que tradicionalmente é utilizado em *detector*es baseados em *anchors*, permite o treinamento direto (*end-to-end*), sem a inclusão de heurísticas ou etapas manuais, garante o emparelhamento ótimo, ou seja, associa a cada objeto real à predição mais adequada.

A *GIoU Loss* é empregada na função $\mathcal{L}_{\text{box}}$ por apresentar vantagens sobre a *IoU* tradicional, como invariância à escala, *gradient*es definidos mesmo sem sobreposição entre caixas, e melhor convergência na localização dos objetos (Rezatofighi et al., 2019).

2.4.4 Vantagens do DETR

O *DETR* apresenta diversas vantagens em relação às arquiteturas tradicionais de detecção de objetos. A primeira delas refere-se à simplicidade conceitual e arquitetural. Diferentemente de abordagens como *Faster R-CNN*, *YOLO* ou *RetinaNet*, o *DETR* elimina componentes projetados manualmente, como a geração de *anchors*, a definição de *aspect ratios*, escalas e densidades de *anchors*, bem como a necessidade do processo de *NMS*. No *DETR*, o *Hungarian matching* assegura que existam predições únicas, eliminando a necessidade de pós-processamento adicional. Dessa forma, a detecção é realizada de maneira direta e *end-to-end*, resultando em um *pipeline* mais limpo e de fácil reprodutibilidade. Segundo Carion *et al.* (2020), o código de inferência do modelo pode ser implementado com menos de 50 linhas em *PyTorch*.

Ao formular a detecção de objetos como um problema de predição de *set prediction*, o *DETR* é capaz de capturar relações globais entre os objetos por meio do mecanismo de

self-attention. Essa abordagem evita a geração de predições duplicadas e permite que o modelo aprenda e explore o contexto global da imagem no processo de detecção. Tal característica contrasta diretamente com as arquiteturas baseadas em *anchors*, nas quais as predições são realizadas de forma local e independente.

Outra vantagem relevante do *DETR* está em sua capacidade de adaptação e generalização. O modelo pode ser empregado em diferentes tarefas de visão computacional, uma vez que sua formulação baseada em atenção e predição por conjuntos é independente de heurísticas específicas para cada problema. O *DETR* pode ser estendido, por exemplo, para tarefas de segmentação panóptica, mediante a adição de uma *mask head* responsável pela predição de máscaras de instâncias e regiões sem instância. Além disso, a abordagem de *set prediction* pode ser aplicada a problemas estruturados, como *keypoints detection* e *object tracking*. O modelo também permite a utilização de diferentes *backbones*, podendo ser integrado a arquiteturas modernas, como *Swin Transformer* ou *ConvNeXt*.

O *DETR* alcança desempenho superior na detecção de objetos de grande porte, superando arquiteturas como o *Faster R-CNN* em termos de *AP* no *COCO dataset*. Esse desempenho pode ser atribuído à sua capacidade de capturar o contexto global da imagem e modelar relações de longo alcance entre os objetos.

Por fim, a paralelização e a eficiência na inferência constituem outra vantagem significativa do *DETR*. O *decoder* do modelo gera todas as predições de forma paralela, permitindo que as N predições sejam computadas simultaneamente. Essa característica resulta em melhor aproveitamento dos recursos de *hardware*, como *GPU*s e *TPUs*, contribuindo para uma inferência mais eficiente.

Entretanto, existem limitações do *DETR* original, temos uma convergência lenta, ou seja, ele requer muitas *epochs* de treinamento, baseado no *COCO* como fator de comparação, o *DETR* necessita de 500 épocas, já a *Faster R-CNN* aproximadamente 12 épocas. Segundo Zhu *et al.* (2021), isso ocorre devido à dificuldade de treinar o módulo de atenção no *encoder* para processar características de imagem.

Além disso, em detecções de objetos pequenos, o *DETR* apresenta um *AP* inferior a métodos como *Faster R-CNN + FPN*, pois ao processar mapas de características de alta resolução no *encoder Transformer* tem custo computacional quadrático $O((HW)^2)$, o *self-attention* no *encoder* tem complexidade $O((HW)^2 * d)$, ou seja, a complexidade é proporcional ao quadrado do produto entre a altura (H) e a largura (W) da imagem, multiplicado

pela dimensão do vetor de características (d), tornando impraticável usar características de alta resolução necessárias para detectar objetos pequenos.

Essas limitações apontadas levaram ao desenvolvimento de variantes melhoradas como o *Deformable DETR* (Zhu *et al.*, 2021), o *Efficient DETR* (Yao *et al.*, 2021) e o *DINO* (Zhang *et al.*, 2022), sendo que esses trabalhos abordam especificamente convergência e detecção de objetos pequenos.

Na aplicação ao reconhecimento de língua de sinais, o *DETR* mostra-se particularmente promissor para a captura de relações espaciais envolvidas em sinais que apresentam configurações complexas de mãos, braços e expressões faciais. O mecanismo de atenção permite modelar explicitamente as relações espaciais entre essas diferentes partes do corpo, favorecendo a compreensão global do gesto. Além disso, a capacidade de detecção de múltiplas instâncias em paralelo constitui uma característica valiosa nesse contexto, uma vez que possibilita o processamento simultâneo de diferentes elementos visuais presentes na cena.

O aprendizado *end-to-end* proporcionado pelo *DETR* contribui para a eliminação de componentes manuais, facilitando tanto a otimização do modelo quanto sua adaptação ao domínio específico da língua de sinais. No entanto, a versão original do *DETR* apresenta limitações quando aplicada a conjuntos de dados menores, comuns em cenários de validação de modelos para reconhecimento de Libras e vocabulários restritos. Essa característica motiva a necessidade de customizações da arquitetura, como a redução do número de camadas e da dimensão dos *embeddings*, aspecto que constitui o tema central desta dissertação.

2.5 Trabalhos relacionados

Esta subseção apresenta uma revisão dos trabalhos relevantes na área de reconhecimento de língua de sinais que utilizam técnicas de *DL*, bem como estudos que exploram a aplicação da arquitetura *DETR* em diferentes domínios. O objetivo é contextualizar a presente pesquisa em relação aos avanços mais recentes da literatura, além de identificar lacunas existentes que fundamentam e motivam o desenvolvimento deste trabalho.

2.5.1 Reconhecimento de língua de sinais com deep learning

O reconhecimento automático de língua de sinais constitui um dos campos mais desafiadores da visão computacional, em decorrência da natureza visual-gestual dessas línguas e da necessidade de capturar, de forma simultânea, CM, M, localização espacial e componentes não manuais, como expressões faciais. Durante um longo período, a combinação de redes neurais convolucionais (*CNNs*) para extração de características espaciais com arquiteturas recorrentes, como *LSTMs* e *GRUs*, para modelagem temporal, consolidou-se como o paradigma dominante para o reconhecimento de sinais dinâmicos ou em movimento (Zhang; Jiang, 2024).

Trabalhos como o do pesquisador De Coster *et al.* (2020) adotaram uma arquitetura híbrida, combinando o *OpenPose* para extração de keypoints com *Transformer*s para o reconhecimento de sinais isolados da língua de sinais flamenca, obtendo acurácia de 74,7% em um vocabulário de 100 classes.

Em contraste, Kothadiya *et al.* (2022), no contexto da *Indian Sign Language* (*ISL*), investigaram uma arquitetura combinando *LSTM* e *GRU* aplicada a um *dataset* de 11 classes, alcançando aproximadamente 97% de acurácia. Essa diferença evidencia como o tamanho do vocabulário pode impactar diretamente o desempenho observado.

De forma semelhante, Bhadouria *et al.* (2024) propuseram um sistema para reconhecimento da *ASL* integrando *LSTM* com *MediaPipe* para detecção de mãos em tempo real. Os *landmarks* extraídos de sequências de vídeo são classificados por uma rede *LSTM*, reforçando, junto as ideias de Kothadiya *et al.* (2022) que a combinação de extratores de *keypoints* com modelos recorrentes é eficaz para capturar dependências temporais em sinais dinâmicos. A estratégia do pesquisador De Coster *et al.* (2020), por sua vez, destaca o potencial de *Transformers* quando acoplados a extratores de *keypoints*.

Kumari e Anand (2024) propuseram uma arquitetura híbrida *CNN-LSTM*, utilizando a *MobileNetV2* como *backbone*, em função de sua estrutura leve e adequada para aplicações com restrições computacionais. Segundo os autores, essa escolha reduz a complexidade da arquitetura sem comprometer a eficácia na extração de características espaciais. Além disso, o mecanismo de atenção empregado no modelo contribui para otimizar a *LSTM* na seleção dos *frames* mais relevantes de cada gesto, enfocando características salientes ao longo da sequência temporal.

As *CNNs 3D* configuram-se como uma abordagem alternativa para o reconhecimento de língua de sinais, uma vez que processam diretamente informações espaço-temporais, permitindo o aprendizado conjunto de características espaciais e temporais. No entanto, apesar de promissoras, essas arquiteturas apresentam elevada complexidade computacional em razão do grande número de parâmetros envolvidos. Nesse sentido, Xiangzu, Fei e Guohui (2022) propuseram uma abordagem híbrida que combina *CNNs 3D lightweight* com *Transformer*s, buscando equilibrar desempenho e eficiência. Nessa proposta, as *CNNs* capturam informações espaço-temporais locais, enquanto os *Transformer*s modelam dependências globais de longo alcance.

A adaptação de arquiteturas baseadas em *Transformer*s para o reconhecimento de língua de sinais representa a fronteira atual da pesquisa na área, ao explorar mecanismos de atenção para capturar relações complexas entre diferentes componentes dos sinais. Shin *et al.* (2023) desenvolveram um modelo baseado em *Transformer* para o reconhecimento da *KSL*, combinando *branches* de *CNN* e *Transformer* para explorar simultaneamente características locais e dependências globais. O modelo alcançou 89% de acurácia em um *dataset* composto por 77 classes, superando abordagens fundamentadas exclusivamente em *CNNs* ou *LSTMs*. A arquitetura proposta emprega *Transformer*s tanto na extração de características espaciais quanto na modelagem das sequências temporais, evidenciando a versatilidade dos mecanismos de atenção em diferentes etapas do *pipeline* de reconhecimento. Os autores destacam, ainda, que os *Transformer*s são particularmente adequados para a língua de sinais em virtude de sua capacidade de modelar relações não locais entre mãos, face e corpo.

O reconhecimento contínuo de língua de sinais, no qual múltiplos sinais ocorrem sem segmentação prévia, apresenta complexidade superior ao reconhecimento de sinais isolados. Sistemas voltados para o *Continuous Sign Language Recognition* (*CSLR*) geralmente empregam arquiteturas *encoder-decoder* com mecanismos de atenção, inspiradas em modelos de tradução de sequências (Zhang; Jiang, 2024). Abordagens mais recentes passaram a incorporar o uso de *Transformer*s com conectividade temporal, de modo a modelar transições suaves entre sinais consecutivos.

Por fim, revisões abrangentes da literatura apontam desafios persistentes na área, como a escassez de *dataset*s anotados em larga escala, especialmente para línguas de sinais menos estudadas; a variabilidade entre signatários em termos de velocidade, amplitude e estilo de execução; a necessidade de modelar adequadamente os componentes não manuais, como expressões faciais e movimentos corporais; as dificuldades de segmentação em cenários de

sinais contínuos; e as limitações de generalização dos modelos para signatários não vistos durante o treinamento.

Apesar do crescente interesse em arquiteturas baseadas em *Transformer* para reconhecimento de língua de sinais, observa-se que a maioria dos estudos se concentra em abordagens híbridas ou no uso de *Transformer* para modelagem temporal. A aplicação da arquitetura *DETR*, originalmente proposta para detecção de objetos, ao reconhecimento de língua de sinais permanece pouco explorada na literatura, especialmente no contexto da Libras. Este trabalho busca preencher essa lacuna ao adaptar o *DETR* para o reconhecimento de um vocabulário controlado de vinte sinais em ambiente controlado.

2.5.2 Aplicações de DETR em outros domínios

Uma das principais áreas de aplicação do *DETR* envolve os veículos autônomos, nos quais a detecção precisa e em tempo real de pedestres, veículos e sinalizações constitui um requisito mínimo para a operação segura desses sistemas. Xing *et al.* (2024) propuseram o *MS-DETR* (*MultiSpectral Pedestrian Detection Transformer*) para a detecção de pedestres em condições de baixa luminosidade, integrando imagens *RGB* e térmicas. A arquitetura emprega dois *backbones* e *encoders* específicos por modalidade, seguidos de um *decoder Transformer multimodal*. Para lidar com desalinhamentos comuns entre modalidades visuais e térmicas em *dataset*s reais, os autores introduziram uma estratégia de fusão *loosely coupled*. O *MS-DETR* apresentou desempenho superior nos *benchmarks KAIST*, *CVC-14* e *LLVIP*. De forma complementar, Gao *et al.* (2025) desenvolveram o *RC-DETR* (*Rank-based Contrastive DETR*), voltado à detecção de pedestres em cenas aglomeradas (*crowded scenes*). O modelo introduz aprendizado contrastivo baseado em ranking para melhorar a separação entre verdadeiros positivos (*VP*) e *FP*, alcançando 38,9% de *miss rate* (*MR*) no *dataset CrowdHuman*, abordando uma das limitações críticas do *DETR* original em comparação com detectores baseados em *CNNs*.

Na área médica, o *DETR* também vem sendo explorado em aplicações de diagnóstico por imagem. Bhat *et al.* (2025) abordaram o desafio da detecção de anomalias em mamografias densas, nas quais a sobreposição de tecidos pode obscurecer lesões. Para isso, os autores propuseram um detector contrastivo multimodal que utiliza *embeddings* exemplares intuitivos

e uma estratégia de treinamento iterativo guiada por *cross-attention*. O modelo obteve desempenho expressivo no *dataset VinDR-Mammo*, alcançando um *mAP* de 0,7 para massas e 0,55 para calcificações, representando um aumento de aproximadamente 16 pontos percentuais em relação a abordagens existentes.

Outra aplicação relevante do *DETR* concentra-se na área de vigilância e segurança. Soares (2024), diante do aumento da violência armada, desenvolveu um sistema híbrido de vigilância baseado na arquitetura *DETR*, combinado com múltiplas redes neurais autocoordenadas para a detecção automática de armas de fogo em imagens de sistemas de segurança. O modelo incorpora redes *MLP* auxiliares inseridas de forma sequencial nas camadas conectadas do *DETR*, com o objetivo de aprimorar o aprendizado, especialmente na identificação de objetos pequenos e oclusos. O sistema alcançou 84% de precisão e 99% de *recall* com o uso de quatro redes adicionais, superando limitações de arquiteturas padrão em cenários de baixa qualidade de imagem. De forma similar, Qi *et al.* (2021) propuseram um sistema de detecção de armas em tempo real para vigilância, utilizando um *dataset* composto por 51 mil imagens. A arquitetura híbrida emprega uma versão simplificada do *CenterNet* com *backbone VoVNetv2* executada em câmeras inteligentes para detecção rápida na borda, enquanto um servidor em nuvem realiza a classificação com *ResNet*. Os resultados alcançados indicaram 90,19% de *recall* na detecção na borda e 97,83% de acurácia na classificação em nuvem com *ResNet-50*, equilibrando eficiência computacional e precisão operacional.

Em resposta a essas limitações de diversos estudos, variantes como o *Deformable DETR* (Zhu et al., 2021), *Conditional DETR* (Meng et al., 2023) e *Real-Time DETR* (*RT-DETR*) (Zhao et al., 2024) foram propostas, incorporando atenção deformável, *queries* condicionais e otimizações voltadas à inferência em tempo real.

2.5.3 Lacunas identificadas

Apesar dos avanços evidenciados na literatura tanto no reconhecimento automático de língua de sinais quanto nas aplicações da arquitetura *DETR*, não foram identificados, na literatura consultada, trabalhos que apliquem diretamente essas abordagens ao contexto da Libras. A maior parte das pesquisas na área concentra-se em línguas de sinais mais amplamente estudadas, como a *American Sign Language* (*ASL*) e a *British Sign Language* (*BSL*). Em

contrapartida, estudos voltados à Libras que utilizem arquiteturas modernas de *DL*, especialmente baseadas em *Transformer*s, ainda se encontram em estágio inicial.

Outro fator limitante refere-se à disponibilidade de *dataset*s públicos de Libras. As bases de dados atualmente existentes, como o V-LIBRASIL da Universidade Federal de Pernambuco (UFPE), o *dataset* da Universidade Federal de Viçosa (UFV), o acervo do Instituto Nacional de Educação de Surdos (INES) e o *SignBank* da Universidade Federal de Santa Catarina (UFSC), apresentam restrições significativas. Entre essas limitações estão o reduzido número de vídeos, a baixa diversidade de gestos, a presença de poucos signatários, o que compromete a capacidade de generalização dos modelos, cenários de captura altamente controlados, que não refletem a variabilidade de condições do mundo real, e o foco predominante em sinais isolados, sem contemplar o reconhecimento de sentenças contínuas. A escassez de dados anotados representa, portanto, uma barreira relevante para o treinamento de modelos robustos de *DL* voltados à Libras.

Uma busca na literatura não identificou trabalhos que apliquem a arquitetura *DETR* especificamente ao reconhecimento de língua de sinais brasileira, essa lacuna abre oportunidades para a adaptação do *DETR* a esse contexto.

Adicionalmente, a maioria dos estudos em reconhecimento de língua de sinais concentra-se predominantemente na maximização da acurácia, por vezes sem a devida ênfase na eficiência computacional e da viabilidade de implementação em dispositivos com recursos limitados. Muitos modelos *baseline* utilizam arquiteturas pesadas que, embora apresentem elevado desempenho em *benchmark*s acadêmicos, apresentam desafios significativos para aplicações reais em dispositivos como smartphones, tablets ou computadores convencionais. Nesse sentido, observa-se a necessidade de investigações que explorem o *trade-off* entre acurácia e eficiência computacional, por meio da proposição de modelos customizados e otimizados para domínios específicos, utilizando estratégias como redução do número de camadas, diminuição da dimensionalidade, *pruning*, quantização, entre outras técnicas.

Outra lacuna relevante diz respeito à ausência de aplicações contextualizadas. A maioria dos estudos em reconhecimento de língua de sinais não se concentra em contextos de uso específicos, recorrendo, em geral, a *dataset*s compostos por sinais genéricos, alfabeto manual ou números. Poucos trabalhos consideram vocabulários especializados voltados a domínios particulares, como educação, saúde ou segurança pública. Durante a revisão da literatura, não foram identificados estudos que desenvolvam sistemas de reconhecimento de sinais direcionados especificamente ao contexto de registro de ocorrências de emergência, o qual

exige reconhecimento rápido e preciso em situações de estresse, além de integração com sistemas de gestão de ocorrências já existentes.

Diante desse cenário, a presente dissertação propõe-se a atuar de forma controlada como um trabalho embrionário para validar a aplicação de um *DETR* customizado e otimizado para o reconhecimento de sinais em Libras, utilizando um vocabulário controlado e específico para o registro de ocorrências de emergência.

O estudo busca explorar as vantagens teóricas dos mecanismos de atenção nesse domínio, viabilizar a implementação em dispositivos convencionais e avaliar não apenas a acurácia, mas também a eficiência computacional do modelo, considerando métricas como *FPS* e tempo de inferência. Dessa forma, o trabalho pretende oferecer uma análise holística sobre a viabilidade prática do sistema proposto, contribuindo tanto para o avanço da pesquisa em reconhecimento de língua de sinais quanto para a aplicação prática de arquiteturas *Transformer* em domínios especializados da visão computacional.

2.6 Considerações finais do capítulo

Este capítulo teve como objetivo apresentar a fundamentação teórica necessária para a compreensão da dissertação, abordando desde os conceitos fundamentais da Libras até o estado da arte em reconhecimento de língua de sinais com *DL* e aplicações da arquitetura *DETR*.

Inicialmente, a fundamentação teórica caracterizou a Libras, apresentando seus principais parâmetros linguísticos, tais como configuração das mãos, M, PA, orientação das palmas e ENM. Também foi discutida a justificativa para a seleção dos gestos adotados neste estudo, considerando sua relevância no contexto do registro de ocorrências de emergência.

Em seguida, foi apresentada a evolução histórica dos métodos de reconhecimento de gestos, partindo das abordagens clássicas fundamentadas em segmentação por cor de pele, extração de contornos e modelos baseados em *Hidden Markov Model*s, até as técnicas modernas que utilizam *DL*. Demonstrou-se como as *CNNs* desempenharam papel central no reconhecimento de gestos estáticos, enquanto arquiteturas recorrentes, como *LSTM* e *GRU*, assim como as *CNNs 3D*, possibilitaram a modelagem dinâmica dos gestos ao longo do tempo. A introdução dos *Transformer*s e dos mecanismos de atenção foi apresentada como a fase mais

recente dessa evolução, trazendo capacidade superior para modelar dependências de longo alcance e relações espaciais globais.

Ao longo da fundamentação teórica, também foram discutidos os principais desafios do reconhecimento de gestos, incluindo variações de iluminação, oclusão e auto-oclusão, complexidade de fundos, variabilidade intra-classe, similaridade inter-classe, exigências de processamento em tempo real e a dependência de *dataset*s anotados em grande escala.

O *DETR* foi apresentado como uma arquitetura inovadora para a detecção de objetos, ao tratar o problema como uma tarefa de *set prediction* e eliminar componentes manualmente projetados. A otimização *end-to-end* baseada no *Hungarian matching* confere ao modelo vantagens como simplicidade conceitual, uso de mecanismos de atenção global e paralelização eficiente durante o treinamento. Por outro lado, também foram discutidas suas limitações, como a convergência lenta e o desempenho inferior na detecção de objetos pequenos, preparando o leitor para a compreensão e justificativa das customizações propostas neste trabalho.

A revisão dos trabalhos relacionados evidenciou avanços significativos no reconhecimento de língua de sinais por meio de *DL*, destacando o uso de arquiteturas híbridas *CNN-LSTM*, *CNNs 3D* e *Transformer*s. Também foi demonstrada a versatilidade do *DETR* a partir de aplicações em domínios como direção autônoma, detecção de pedestres, imagens médicas e sistemas de vigilância. No entanto, a revisão identificou lacunas importantes, como a escassez de trabalhos voltados ao reconhecimento de Libras utilizando *Transformer*s, a ausência de aplicações do *DETR* nesse contexto específico, a carência de modelos otimizados com foco em eficiência computacional e a inexistência de sistemas orientados a aplicações práticas para serviços emergenciais.

Essas lacunas identificadas justificam e motivam o desenvolvimento do *DETR-Custom* proposto nesta dissertação: uma adaptação da arquitetura *DETR* para o reconhecimento de sinais em Libras, com vocabulário controlado e foco na eficiência computacional. A proposta considera sua aplicação em computadores convencionais, sem a necessidade de *GPU*s dedicadas, e sua utilização como ferramenta de apoio ao registro de ocorrências em serviços de emergência, representando uma contribuição original que busca preencher lacunas relevantes na literatura.

Por fim, os conceitos, técnicas e referências apresentados neste capítulo fornecem a base teórica sobre a qual a metodologia proposta no capítulo seguinte será construída. Esses conhecimentos são importantes para a compreensão das decisões metodológicas adotadas, a

interpretação dos resultados experimentais e o posicionamento das contribuições desta dissertação no contexto mais amplo da pesquisa em visão computacional e *TA*.

3. METODOLOGIA

Este capítulo traz em detalhes a estrutura metodológica empregada no desenvolvimento e avaliação do modelo *DETR-Custom* proposto. A abordagem realizada segue um *pipeline* de pesquisa aplicada, iniciando pela adaptação da arquitetura do *DETR* para o domínio específico do reconhecimento de sinais em Libras, seguida da construção e preparação de um *dataset* próprio com imagens do próprio autor, do treinamento supervisionado do modelo com técnicas de data augmentation e da avaliação rigorosa do desempenho quantitativo e qualitativo. A estratégia metodológica foi desenhada para validar empiricamente a hipótese central, que verifica a viabilidade de um *Transformer* customizado, mais eficiente computacionalmente, para tarefas especializadas de detecção de sinais em um vocabulário controlado.

3.1 Visão geral da proposta

Inicialmente a pesquisa bibliográfica exploratória foi realizada com o apoio do Consensus (Consensus NLP), ferramenta de busca acadêmica baseada em inteligência artificial, acessada em 2025, empregada exclusivamente para busca de referências relacionadas ao tema com seu respectivo DOI, após listagem foi obtido o artigo original nas bases acessadas por intermédio do CAFe (Comunidade Acadêmica Federada), vinculada ao Portal de Periódicos da CAPES. Foi utilizado a ferramenta de IA generativa Claude da empresa Anthropic, para correção gramatical, ortográfica e de clareza do texto redigido pelo autor e a gestão, a organização e a padronização das referências foram realizadas com o software Zotero, versão 7.0.32 (64 bits).

O projeto parte da premissa de reconhecer imagens coloridas e os sinais realizados por pessoas surdas em um contexto de emergência, possibilitando uma comunicação eficiente entre pessoas surdas e ouvintes. A escolha do algoritmo *Hungarian matching* como base para o treinamento do modelo foi fundamentada na pesquisa bibliográfica exploratória realizada na etapa inicial deste trabalho.

Entre as abordagens identificadas na literatura para associação entre predições e objetos reais em tarefas de detecção, o método húngaro demonstrou-se o mais apropriado para a

arquitetura *DETR*, por permitir correspondência ótima global sem necessidade de heurísticas como *NMS*. Essa escolha justifica a presença de outros métodos no referencial teórico, como *detector*es baseados em *anchors* e region proposals que serviram de base comparativa, enquanto o recorte metodológico concentra-se na abordagem de *set prediction* com *matching* bipartido.

A metodologia foi desenhada para testar a hipótese da subseção 1.3, onde uma versão customizada do sistema *DETR* capaz de realizar as seguintes ações, realizar o reconhecimento de 20 classes de sinais de Libras com *mAP@0.5* ≥ 0,80 e um *F1-Score* médio ≥ 0,80, manter uma eficiência computacional próxima de uma utilização em tempo real, com uma taxa de *frames* de *FPS* ≥ 15 e uma latência ≤ 50 ms para cada imagem, considerando o *hardware* de referência.

Etapas do fluxo metodológico utilizado:

Construção do *Dataset* com imagens com sinais de Libras, com uma definição clara de classes, e um protocolo de coleta que utiliza critérios éticos de coleta, anotação manual de *bounding boxes* e uma divisão de subconjuntos do treino, teste e a validação do *dataset* e teste do mesmo. A definição da arquitetura *DETR Customizada*, a escolha do *backbone* convolucional (*ResNet-50*) (He et al., 2016), a realização da configuração do *encoder*, a configuração do *decoder* e também a configuração do *Transformer*, o número de *object queries* e a realização do desenho das cabeças de predição de classe bem como dos *bounding boxes*.

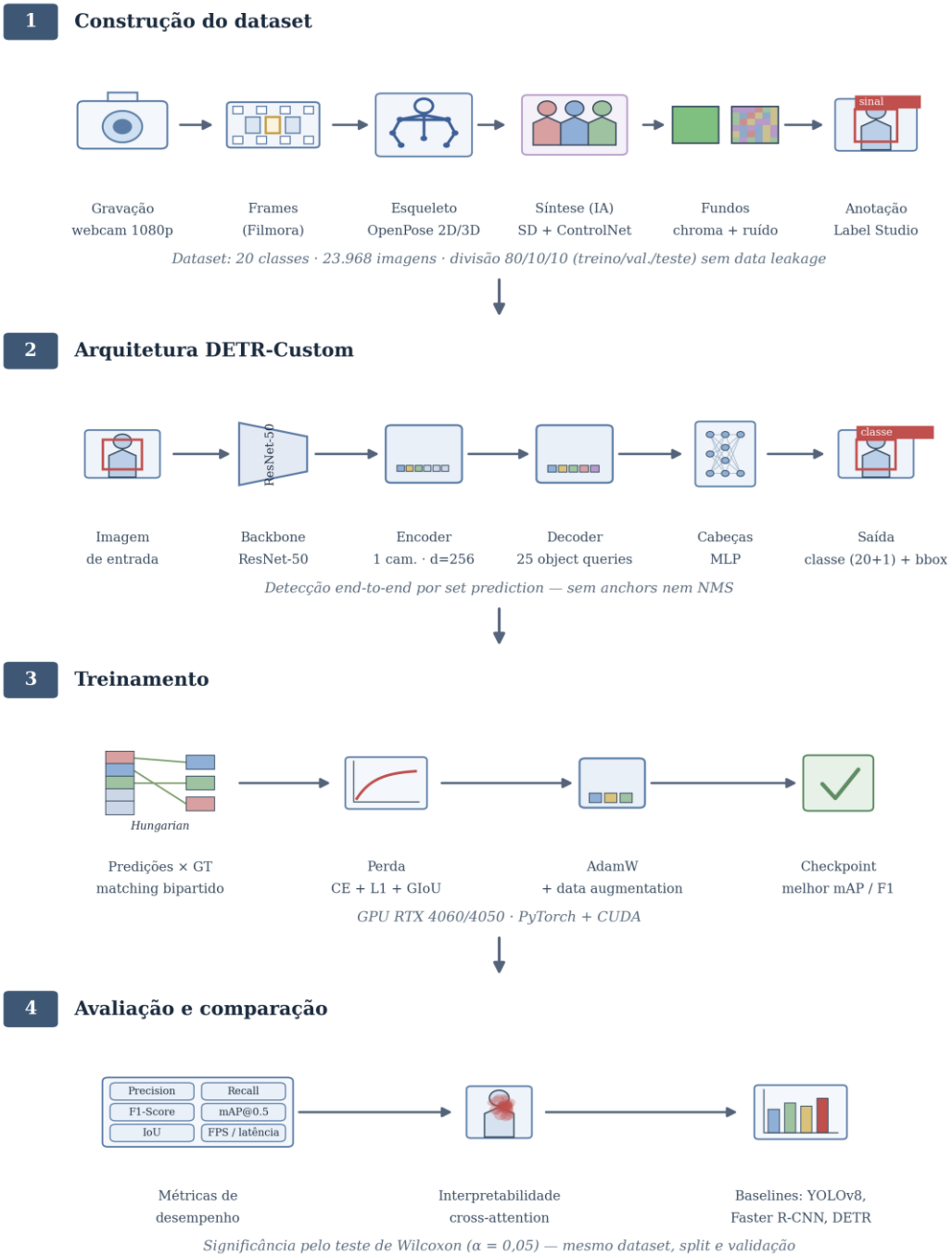
Customização e treinamento do modelo, empregando a especificação de perda multiobjetivo, do algoritmo de *matching bipartite* (Kuhn, 1955), do otimizador, dos hiperparâmetros e da estratégia de treinamento.

Avaliação quantitativa empregando as métricas de detecção e classificação *Precision*, *F1-Score*, *Recall*, *IoU* e *mAP*, além delas, também foram utilizadas de maneira complementar, medidas de desempenho computacional (*FPS* e latência) a comparação com *baselines* de detecção de objetos (*YOLOv8*, *DETR* original e *Faster R-CNN*), com base nos critérios de avaliação e valores reportados por Carion *et al.* (2020) e Ren *et al.* (2017).

A Figura 8 sintetiza o fluxo metodológico completo, do dataset à comparação com os baselines.

As próximas seções detalham todos os componentes desse fluxo, partindo da arquitetura até o protocolo experimental, desta maneira, é possível garantir total transparência no processo de desenvolvimento, reprodutibilidade e alinhamento entre o problema proposto e a pesquisa realizada, a hipótese que foi formulada e os experimentos que foram realizados.

Figura 8: Fluxo da metodologia de pesquisa.



Fonte: Dados da pesquisa (2025).

3.2 *DETR Custom*: arquitetura proposta

Como foi apresentado na subseção 2.4, o *DETR* aborda a detecção de objetos como um problema de *set prediction*, eliminando os componentes heurísticos como as *anchors* e *NMS* por meio de um *pipeline end-to-end* composto por um *backbone* convolucional, *Transformer*

encoder-decoder e *Hungarian matching* (Carion *et al.*, 2020). Nesta subseção são descritas as adaptações introduzidas nessa arquitetura para o trabalho desenvolvido, bem como as razões que fundamentaram a sua escolha.

A adoção do *DETR* como arquitetura para este estudo implica em fazer *trade-offs* em relação a modelos já consolidados.

Arquiteturas que utilizam *CNNs* como o *YOLOv8*, possuem a tendência de convergir mais rápido, e, a alcançar resultados mesmo frente a volumes reduzidos de dados e a apresentar tempos de inferência menores em configurações já padronizadas.

As razões que sustentam a escolha da arquitetura residem em utilizar o mecanismo de atenção global do *Transformer* que captura relações espaciais de longo alcance entre mãos, face e tronco, elementos que conjuntamente define o significado do sinal em Libras, enquanto que as *CNNs* modelam as mesmas relações de modo indireto, por intermédio de campos receptivos locais, conforme apresentado na sub-subseção 2.3.3; a supressão de componentes heurísticos, como *anchors* e *NMS*, reduz a calibração específica para cada domínio conferindo a arquitetura *DETR* maior flexibilidade de adaptação; o levantamento bibliográfico conduzido na sub-subseção 2.5.3 revelou que a aplicação do *DETR* ao reconhecimento de Libras permanece inexplorada.

Dessa forma uma arquitetura já extensivamente validada, como o *YOLO*, embora pudesse gerar métricas mais elevadas, representaria uma contribuição científica com menor alcance.

As customizações descritas nas seções subsequentes, redução a uma única camada de *encoder-decoder*, dimensão oculta de 256 e 25 *object queries*, foram concebidas com o propósito específico de atenuar as limitações reconhecidas do *DETR*.

As subseções abaixo detalham os componentes desta arquitetura.

3.2.1 Backbone: ResNet-50

O *backbone* adotado é a *ResNet-50*, uma rede neural convolucional com 50 camadas, ela também utiliza blocos residuais com o objetivo de contornar o problema do *vanishing gradient*, esses blocos residuais adicionam conexões entre as camadas de atalho, também

conhecidas como *skip connections*, que facilitam o fluxo de *gradient*es em redes muito profundas (He et al., 2016). Essa arquitetura se popularizou em tarefas de classificação e detecção de imagens por combinar uma boa capacidade de representação com um custo computacional relativamente baixo.

Neste projeto, a *ResNet-50* é utilizada como extrator de características de nível intermediário a partir de imagens de entrada que utilizam o padrão *RGB*. As principais decisões de configuração são:

Entrada: imagens que foram redimensionadas para uma resolução padrão (800x600 *pixel*s), preservando a proporção original;

Camadas finais: remoção da camada totalmente conectada (*FC*) originalmente utilizada para classificar em *ImageNet*, de modo a aproveitar apenas o mapa de características convolucionais;

Mapa de características: uso do *feature map* na saída do quarto bloco residual, com dimensão de $2048 \times H\ x\ W$, onde H e W são a dimensão espacial reduzida da imagem;

Redução de canais: aplicando uma convolução 1x1 para reduzir a dimensão de 2048 para 256 canais, para compatibilizar com a saída do *backbone* que possui uma dimensão do *Transformer* de $d_{model} = 256$.

3.2.2 Encoder Transformer

O mapa de características gerado pelo *backbone* passa por uma etapa de transformação chamada de achatamento (*flattening*) do mapa em uma sequência de vetores, denominados *tokens* visuais, aos quais são adicionadas codificações posicionais. Cada coordenada (i, j) do mapa corresponde a um *token* individual, o que permite preservar a disposição espacial da imagem ao longo do processamento.

A arquitetura do *encoder* segue o padrão descrito por Vaswani *et al.* (2017). O primeiro é o mecanismo de *MHSA*, configurado com oito cabeças de atenção, que permite a cada posição do mapa acessar informações das demais posições da imagem, permitindo ao modelo a capacidade de capturar relações entre regiões distintas, como as mãos, braços, rosto e o contexto corporal, dessa forma enxergando como cada parte compõe a imagem.

Na sequência, uma *FFN*, composta por duas camadas lineares separadas, uma com a função não linear (*ReLU*), com dimensão intermediária de 2048, após aplica-se transformações ponto a ponto sobre cada *token*.

Antes de cada sub-bloco, uma operação de *Layer normalization* ajusta a escala e o deslocamento das ativações, estabilizando a distribuição dos valores em cada *token* e prevenindo que as ativações cresçam de modo descontrolado ou se tornem desbalanceadas durante o treinamento.

Por fim, conexões residuais somam a entrada de cada sub-bloco à sua saída, funcionando como *by-pass* para preservar a informação original junto a informação passada, dessa forma com ambas as informações é atenuado o desaparecimento de gradientes (*vanishing gradient*).

A arquitetura adotada neste trabalho corresponde a uma versão simplificada do *DETR*, chamada de *DETR-Custom*, que sofreu três modificações, sendo que a primeira o *encoder* possui uma única camada, em contraste com as seis camadas da versão original, a dimensão oculta foi reduzida $d_{model} = 256$, ao invés $d_{model} = 512$ e o número de cabeças de atenção foi mantido em oito, como ponto de equilíbrio entre custo computacional e capacidade de modelagem.

Essa configuração mais compacta se justifica pelo escopo restrito do projeto, voltado à interpretação de um conjunto limitado de sinais, cenário que dispensa a complexidade exigida por tarefas como detecção em larga escala no *dataset COCO* ou análise de vídeos com alta densidade de objetos.

3.2.3 Decoder Transformer

O *Decoder Transformer* recebe um *input* de dois elementos, o primeiro sendo um conjunto de *embeddings* produzidos pelo *encoder* representado pela imagem, e o segundo, um conjunto fixo de *object queries*, vetores aprendidos que funcionam como *slots* que o modelo irá utilizar para escrever as suas predições de objetos.

As camadas do *decoder Transformer* executam respectivamente o *Masked MHSA* entre as *queries*, possibilitando que consultas sejam feitas entre si para evitar que sejam feitas predições redundantes, a *cross-attention* entre as *queries* e os *tokens* que saem do *encoder*, onde

cada *query* realiza o trabalho de buscar regiões que são relevantes em cada imagem, a *FFN* que é aplicada individualmente em cada *query*, depois a *layer normalization* e conexões residuais, de acordo com o padrão *Transformer*.

Dentro do modelo *DETR-Custom* com escopo reduzido tem-se o número de camadas do *decoder*, 1 camada vs 6 no *DETR* original, dimensão oculta 256, número de cabeças de atenção 8 e número de *object queries* 25, fixos para todas as imagens.

A redução de camadas reduziu consideravelmente o treinamento, preservando a funcionalidade central de *cross-attention* que possibilita que o modelo localize sinais em diferentes posições da imagem.

3.2.4 Cabeças de predição

A cabeça de predição é o módulo final que transforma a representação interna em uma saída concreta para a tarefa específica, transformando *embeddings* em previsões úteis.

Uma configuração de 25 *embeddings*, cada uma delas associada a uma potencial detecção, e sobre cada embedding são aplicadas suas cabeças de predições em paralelo:

A cabeça de classificação segue a estrutura de um perceptron multicamadas (*MLP*) com camadas intermediárias ativadas pela função *ReLU*, a camada final produz um vetor de *logits* com dimensão (20 + 1), correspondendo às 20 classes de sinais mais o rótulo especial "*no object*" (nenhum objeto), indicando a ausência de detecção.

Durante a inferência, a distribuição de probabilidades é obtida via *softmax*.

A cabeça de regressão de *bounding boxes* também adota a arquitetura *MLP*. Sua camada final produz quatro valores contínuos cx, cy, w, h, que representam, respectivamente, as coordenadas do centro, a largura e a altura da caixa delimitadora, todos normalizados no intervalo [0, 1] em relação as direções horizontal e vertical e a largura e altura, também normalizadas da caixa delimitadora relativa à imagem. Essas predições são, em seguida, convertidas em coordenadas absolutas de *pixel*s para o cálculo de métricas de avaliação e para a visualização dos resultados.

A separação em duas camadas especializadas permite otimizar a classificação, mantendo a arquitetura simples e compatível com a formulação de perda multiobjetivo.

3.2.5 Object queries

No início do treinamento, o modelo cria 25 vetores de dimensão, que não carregam qualquer informação semântica relevante, sendo somente ruído aleatório. À medida que o treinamento avança, os gradientes ajustam progressivamente esses vetores, e cada um deles passa a se especializar na detecção de padrões específicos presentes nos dados.

O modelo emprega 25 *object queries*, quantidade superior ao número máximo de sinais esperados na detecção em uma única imagem. Essa escolha proposital garante uma margem de folga e para cenários em que múltiplos sinais coexistem na mesma imagem ou em que surgem ruídos inesperados durante a inferência.

As mesmas *queries* são utilizadas entre todas as imagens do *dataset*, o que favorece a aprendizagem de padrões consistentes e desenvolver uma maior capacidade de generalização do modelo.

O papel semântico de cada query não é atribuído manualmente, mas surge de modo natural ao longo do treinamento, dessa forma não existe nenhuma regra predefinida que associe uma query específica a uma classe determinada, sendo que o próprio modelo organiza essa correspondência.

Essa forma de trabalhar elimina a necessidade de criar *anchors* ou regiões pré-definidas na imagem e segue o princípio de *set prediction* introduzido pelo *DETR*.

3.2.6 Justificativa das customizações

As customizações introduzidas na arquitetura foram motivadas por conta das características específicas do projeto, que possui um escopo reduzido, e das limitações de *hardware*.

3.2.6.1 Redução de camadas (1 camada vs 6 camadas)

A arquitetura original do *DETR* emprega 6 camadas de *encoder* e 6 camadas de *decoder*, o que eleva o custo computacional e prolonga o tempo de convergência do modelo. Neste trabalho, adotou-se a configuração de apenas 1 camada em cada bloco do *Transformer*, decisão motivada por três fatores. Primeiro, o domínio da aplicação envolve um número reduzido de objetos por imagem, limitando-se a um ou dois sinais simultâneos. Segundo, as cenas apresentam características controladas, com enquadramento estável, baixa variabilidade visual e foco na parte superior do corpo. Terceiro, há o interesse em viabilizar o treinamento em *hardware* convencional, próximo de dispositivos de uso doméstico, o que aproxima a proposta da realidade de equipamentos com menor capacidade de processamento.

Com essa redução no número de camadas, esperava-se diminuir o tempo de treinamento de modo expressivo, sem prejuízo à acurácia, que tendia a permanecer em patamares elevados quando comparada aos resultados obtidos pelo modelo tradicional.

3.2.6.2 Dimensão oculta (256 vs 512)

Outra otimização de performance computacional foi a proposta da redução da dimensão oculta do *Transformer* de 512 (*DETR* original) para 256.

Com essa diminuição no número de parâmetros e operações, espera-se um consumo menor de *GPU*, maior velocidade nas etapas de treinamento e inferência, e menor propensão ao *overfitting*, dado que o *dataset* utilizado, embora diversificado, possui escala inferior à de *benchmark*s consolidados como o *COCO* e o *ImageNet*.

Esperava-se que a adoção da dimensão 256 fosse suficiente para capturar os padrões visuais dos sinais em Libras.

3.2.6.3 Número de object queries (25)

O número de *object queries* foi definido em 25, considerando três critérios, à quantidade máxima de sinais de Libras que podem existir de forma simultânea em uma única imagem, a necessidade de reservar predições adicionais para absorção de ruídos e falsas detecções, e o equilíbrio entre a capacidade de representação de múltiplos objetos e o custo computacional do algoritmo Húngaro, cuja complexidade cresce de forma proporcional ao número de predições.

Por se tratar de uma prova de conceito, as imagens contêm apenas um objeto principal, sendo esse o sinal em execução, e alguns elementos ruidoso que pertencem ao fundo artificialmente gerado. As *queries* que não correspondem a nenhum objeto recebem o rótulo "*no object*", tratado de modo explícito na função de perda.

3.2.7 Implementação e código-base

A implementação da arquitetura *DETR-Custom* utilizou como ponto de partida o código do repositório *SignDETR* (Renotte, 2024), disponibilizado sob licença MIT, uma reimplementação da arquitetura *DETR* (Carion *et al.*, 2020) voltada especificamente para detecção de linguagem de sinais americano. Essa referência foi escolhida pela aderência direta ao domínio da pesquisa e pela compatibilidade com os requisitos do projeto.

As modificações introduzidas neste trabalho concentraram-se em três frentes:

(i) adaptação arquitetural, conforme descrito nas sub-subseções 3.2.1 a 3.2.6, contemplando a redução para uma camada de *encoder* e *decoder*, dimensão oculta de 256 e 25 *object queries*;

(ii) adaptação do *script* de treinamento para execução em *GPU* dedicada (*NVIDIA GeForce RTX 4060*), incluindo a migração do modelo, critério de perda e *tensor*es de entrada e saída para dispositivo *CUDA*, além do ajuste do número de épocas de 100 para 1.000, compatível com a escala do *dataset* construído (23.968 imagens), e a remoção do carregamento de pesos pré-treinados para treinamento supervisionado completo a partir do *dataset* próprio para as 20 classes de sinais de Libras, estratégias de *checkpoint* e técnicas específicas de *data augmentation* respeitando a semântica dos sinais;

(iii) desenvolvimento de módulos auxiliares próprios: o *pipeline* de construção do *dataset*, o procedimento de geração de imagens sintéticas a partir de *prompts* (sub-subseção 3.3.1), o *script* de verificação automatizada de *data leakage* e os procedimentos de análise de interpretabilidade por *cross-attention* (sub-subseção 3.5.5), os códigos desenvolvidos foram debugados e refatorados usando o Claude Code.

3.3 Dataset

A qualidade do modelo está fortemente relacionada à qualidade do *dataset* utilizado durante o treinamento. Esta subseção detalha o protocolo de coleta utilizado, a distribuição de classes e a divisão dos dados.

3.3.1 Coleta de dados e geração de sintéticos

A primeira fase do desenvolvimento do *dataset* consistiu-se no levantamento de repositórios públicos de gestos em Libras, foi verificado se bases poderiam ser parcial ou integralmente reaproveitadas.

Durante essa etapa foram examinados os *dataset*s hospedados em plataformas como *Kaggle* e *Universe Roboflow*, bem como acervos de origem acadêmica e institucional, o V-LIBRASIL, mantido pela Universidade Federal de Pernambuco (UFPE); o dicionário de sinais da Universidade Federal de Viçosa (UFV); os vídeos produzidos pelo Instituto Nacional de Educação de Surdos (INES); o *SignBank*, vinculado à Universidade Federal de Santa Catarina (UFSC); *Mendeley Data*, vinculado à *Elsevier*; e *Zenodo* mantido pelo CERN (Organização Europeia para a Investigação Nuclear).

A análise desses repositórios evidenciou algumas restrições que, consideradas em conjunto, tornaram inviável o aproveitamento dos recursos identificados, a primeira é em relação ao tipo de datset, as plataformas como *Kaggle*, *Universe Roboflow* e *Mendeley Data* possuem somente *datset* de datilologia (a representação manual do alfabeto), a segunda restrição refere-se ao escopo dos sinais, pois algumas não possuem os sinais selecionados na

sub-subseção 2.1.4, a terceira é quantitativa, mesmo as bases que contêm sinais selecionados, disponibilizam no máximo até três vídeos por gesto, que é uma quantidade insuficiente para a extração das cerca de 200 imagens por classe.

Em razão dessas constatações, decidiu-se pela construção de um *dataset*.

Dessa forma o *dataset* foi construído utilizando-se uma câmera *Webcam* 1080p C90s pro HD para geração de vídeos de três segundos de cada gesto, o enquadramento dos vídeos focou na parte superior do corpo, de modo a incluir mãos, braços e face, que são os elementos fundamentais para interpretar sinais em Libras; a configuração dos sinais gravados pode ser consultada no Apêndice A (Configurações de sinais em Libras).

Em sequência foi utilizado o programa de edição de vídeos *Filmora* versão 14.5.20 para extração de 30 *frames* por vídeo, na imagem abaixo somente foram selecionados os 15 *frames* para representação do processo do gesto, e, desses *frames* foi extraído o *frame mais representativo* do gesto circulado em vermelho como pode ser observado um exemplo na Figura 9.

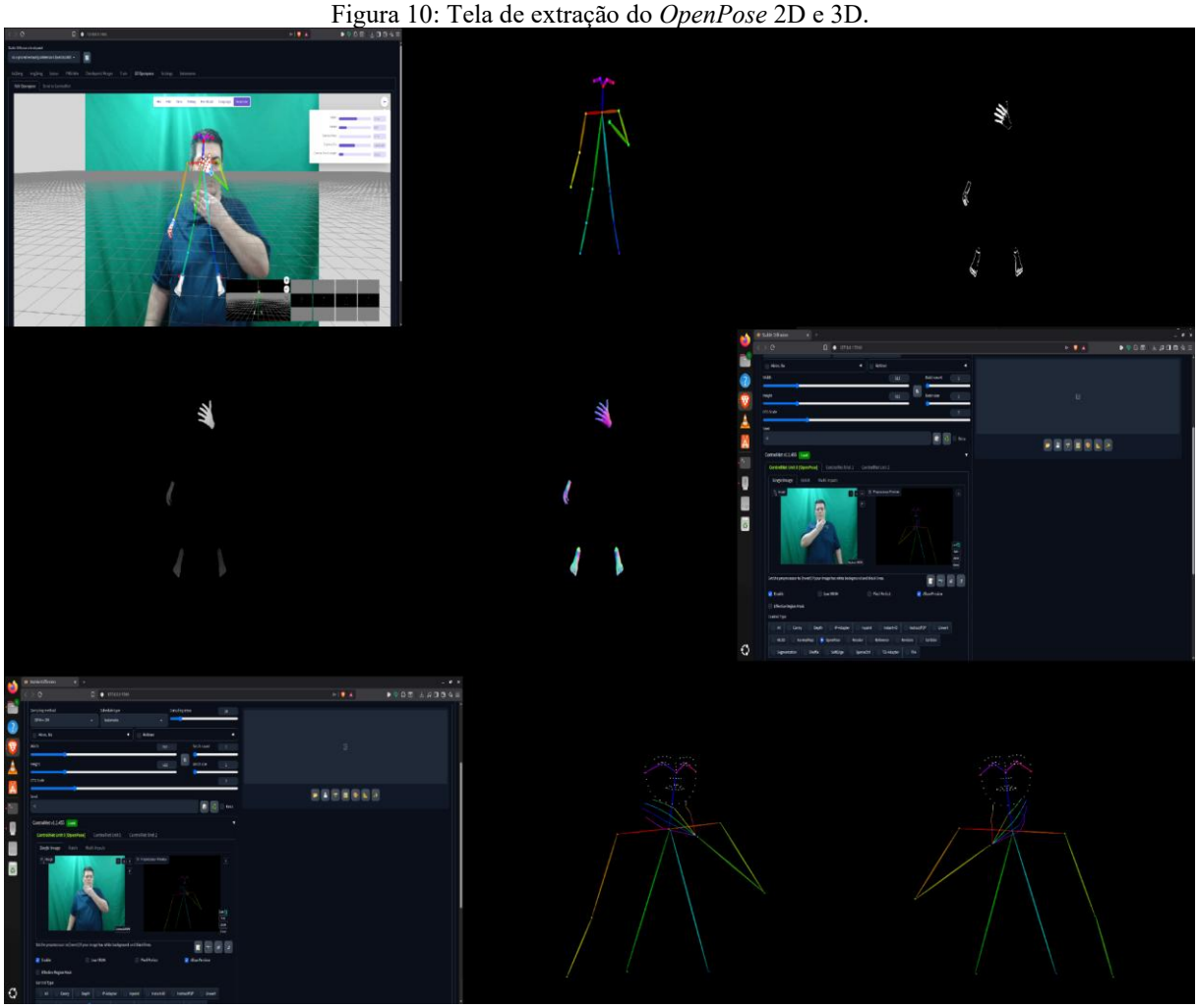
Figura 9: Sinal “EU” com o *frame* mais representativo selecionado.

Fonte: Dados da pesquisa (2025).

Com o *frame* mais significativo escolhido, seguiu-se com a utilização do *Stable Diffusion* A1111 (*AUTOMATIC1111*) com a extensão *ControlNet* (que restringe a geração da imagem de modo a preservar a pose original) que integra o *OpenPose* 2D e 3D (responsável pela extração do esqueleto da pose do *frame*) ao *Stable Diffusion* A1111, foi feita a extração do esqueleto de cada imagem utilizando-se a ferramenta *OpenPose* 3D inicialmente

fornecendo o esqueleto macro do gesto bem como a posição dos dedos e após foi refinado o esqueleto usando o *OpenPose* 2D, como pode ser observado na Figura 9.

Figura 10: Tela de extração do *OpenPose* 2D e 3D.



Fonte: Dados da pesquisa (2025).

Os esqueletos gerados foram utilizados dentro do modelo de inteligência artificial (*IA*) *Qwen3.6-Plus* para geração das imagens, sendo que foram projetadas três imagens sintéticas para aumentar a variabilidade de tons de pele e idade das imagens do *dataset*.

O primeiro *prompt* utilizado como base para a criação da imagem sintética feminina utilizado foi "*Making exactly the same gesture as the skeleton provided by OpenPose, do not show the skeleton provided by OpenPose. Portrait photo of a mature Brazilian woman, age 43-48, medium olive-brown skin tone, oval slightly rounded face, soft cheekbones, gentle jawline without strong angles, dark brown eyes with slightly drooping eyelids showing natural aging, thin natural slightly arched dark eyebrows, medium nose with slightly wide base, medium lips*

with mouth closed in a completely neutral solemn expression with no smile, no visible makeup, natural skin texture with subtle forehead lines and faint crow's feet, slight natural shine on forehead. Black straight to slightly wavy hair, medium length at shoulder level, parted in the middle and pulled back loosely with some strands falling on both sides of the face, visible baby hairs at the hairline, slight natural frizz, no bangs. Medium to full body build, broad rounded shoulders, medium-large bust. Wearing a loose-fitting black peasant blouse (bata style) with wide square bateau neckline gathered with elastic creating small pleats, short puffy sleeves with elastic gathered hem ending at mid-bicep, lightweight cotton or viscose fabric, straight loose silhouette with no waist definition. No visible jewelry or accessories. Standing upright, arms at sides, facing camera directly, neutral expression. Waist-up shot, soft even lighting, solid teal-green fabric backdrop with visible wrinkles. Realistic photography, natural unposed look, DSLR camera style".

O segundo *prompt* utilizado como base para a criação da imagem sintética masculina utilizado foi "*Making exactly the same gesture as the skeleton provided by OpenPose, do not show the skeleton provided by OpenPose. Portrait photo of a young Brazilian mixed-race man, age 24-27, light brown caramel skin tone, oval elongated face, slightly pointed chin, soft cheekbones, moderately defined jawline, dark brown eyes looking directly at camera, natural slightly arched dark eyebrows, medium nose slightly wide at base, thin sparse mustache and light goatee around mouth and chin area, cheeks mostly clean. Short black curly afro-textured hair (type 3C-4A) with more volume on top and shorter on sides, small defined curls, well-groomed. Slim to medium build, medium-width shoulders. Wearing a dark navy blue short-sleeve polo shirt, piquet knit fabric, collar up, button placket with visible light gray inner strip contrast detail, buttons closed, regular relaxed fit, shirt partially tucked into dark gray pants/shorts (only waistband visible). Standing upright with arms relaxed at sides, neutral serene expression with very subtle hint of a smile, mouth closed. Waist-up shot, facing camera directly, soft even lighting. Realistic photography, natural look, DSLR camera style*".

O terceiro *prompt* utilizado como base para a criação da imagem sintética masculina utilizado foi "*Making exactly the same gesture as the skeleton provided by OpenPose, do not show the skeleton provided by OpenPose. Portrait photo of a young Brazilian mixed-race man, age 19-22, medium brown skin tone, oval-rectangular face shape, prominent cheekbones, angular defined jawline, long thin neck, dark brown almond-shaped eyes looking slightly to the side, thick straight dark eyebrows, medium nose slightly wide at base, clean-shaven face with no facial hair, short black tightly coiled afro hair (type 4A-4B) in a low buzz cut with defined*

straight hairline. Slim ectomorph build, narrow shoulders, long thin arms. Wearing a plain light gray heather crew neck short-sleeve cotton t-shirt, regular fit, no print. Standing upright with arms relaxed at sides, neutral serious expression, mouth slightly open. Realistic photography, casual candid look, webcam or smartphone camera style, soft natural indoor lighting".

Lembrando que os *prompt*s para geração de imagem devem ser feitos em inglês, pois são compreendidos melhor pelos modelos de *IA*.

Foi utilizado o modelo de geração de imagem *Nano Banana Pro* para uma geração de um rosto aleatório, após foi feito um *merge* dos rostos utilizando o *FaceSwap* para garantir que os rostos gerados pelos modelos de *IA* não existam no mundo real e sejam somente imagens sintéticas.

Após o *merge* feito nos rostos e pesquisa no buscador do *google* para garantir que as pessoas não existiam como pode ser observado no Anexo B (Pesquisa de imagens sintéticas no *Google* imagens), foi colocado nas imagens um fundo verde *chroma-key* e um fundo ruidoso para uso de treinamento e robustez do modelo, sendo que os fundos foram gerados a partir de *prompt*s e utilizados para geração no *Labs ImageFX* conforme pode ser visto no Anexo C (*Prompts* de geração dos fundos ruidosos). Sendo que os fundos ruidosos foram gerados a partir de prompts textuais elaborados com o auxílio do modelo Gemini (Google) e renderizados no Google Labs ImageFX (modelo Imagem), conforme pode ser visto no Anexo C (Prompts de geração dos fundos ruidosos).

Cada imagem ao final herda o rótulo de classe do *frame* que lhe deu origem, uma vez que a pose é preservada pelo *ControlNet* durante a geração sintética. Essa rastreabilidade entre o *frame* de origem e todas as suas variantes é essencial para o protocolo de divisão dos dados descrito na sub-subseção 3.3.4, pois garante que variantes de um mesmo esqueleto sejam atribuídas ao mesmo subconjunto (treino, validação ou teste), evitando contaminação entre subconjuntos.

A Tabela 3 sintetiza a cadeia de construção do *dataset*, desde a gravação dos vídeos até a composição final das imagens utilizadas no treinamento.

Tabela 3: Cadeia de construção do *dataset*, da gravação ao conjunto final.

Etapa	Descrição	Quantidade
Vídeos gravados	Vídeos de 3 s por sinal, capturados com webcam 1080p	*webcam* 1080p 200 vídeos por classe × 20 classes = 4000 vídeos
Extração de *frames*	30 *frames* extraídos por vídeo via Filmora 14.5.20	6000 *frames* brutos por classe
Seleção do *frame* representativo	1 *frame* selecionado por vídeo, correspondente à CM mais definida do sinal	200 *frames* selecionados por classe
Extração do esqueleto	*OpenPose* 2D e 3D aplicado a cada *frame* selecionado	200 esqueletos por classe
Aplicação de fundos	Cada imagem recebeu 2 fundos (*chroma-key* e/ou ruidoso)	2 imagens por classe
Dataset	Imagens anotadas prontas para uso (*chroma-key*)	11.984 imagens
Dataset	Imagens anotadas prontas para uso (ruidoso)	11.984 imagens
Total do *dataset*	Imagens anotadas prontas para uso	23.968 imagens

Fonte: Dados da pesquisa (2025).

3.3.2 Anotação

Cada imagem do *dataset* foi anotada manualmente utilizando-se o *Label Studio*, utilizando *bounding boxes*, que são responsáveis pela delimitação da região na qual o sinal é executado e por indicar a posição das mãos, e, por um rótulo de classe associado a cada um dos sinais definidos no escopo do projeto. O processo de anotação obedeceu a um protocolo padronizado, o que garantiu a consistência na forma e no dimensionamento das caixas, posicionamento central aproximado sobre a região de interesse e coerência entre diferentes anotadores, por meio da anotação do tipo *YOLO* que é um arquivo de extensão (.txt) (classe x_centro y_centro largura altura), sendo que a primeira posição representa a classe que está associada a imagem, a segunda e a terceira posição são as coordenadas do ponto central da

bounding box, normalizadas de 0 a 1 em relação à imagem inteira, a quarta e a quinta posição são as dimensões da bounding box que são também normalizadas de 0 a 1.

3.3.3 Distribuição das classes

O *dataset* reúne 20 classes de sinais, com gestos de uso cotidiano e gestos relacionados a situações de pedido de ajuda. A distribuição entre as classes seguiu o critério de balanceamento, com número igual de instâncias por sinal. Dessa forma evita-se que o modelo tenda a priorizar classes com maior representatividade, ou seja, maior quantidade de imagens na classe, favorece a leitura das métricas por classe (*Precision*, *Recall* e *F1-Score*), maior confiabilidade ao *F1-Score* médio como indicador do comportamento global do sistema.

O *dataset* totaliza 23.968 imagens anotadas, total esse, resultante da aplicação de dois fundos, *chroma-key* verde e fundo ruidoso colorido nas 11.984 imagens originais. Para o treinamento, teste e validação do modelo nos experimentos descritos no Capítulo 4, utilizou-se o subconjunto de 11.984 imagens com fundo verde, já as 11.984 imagens de fundo ruidoso foram reservadas para o teste de robustez realizados na sub-subseção 4.8.2 junto ao 3 sintético criado e para a análise de interpretabilidade em fundos não controlados, essa escala mostra-se adequada à complexidade da arquitetura de rede adotada.

3.3.4 Divisão dos dados e integridade dos subconjuntos

A divisão do *dataset* fez-se seguindo uma estratégia sequencial em três etapas, para mitigar o risco de contaminação entre os subconjuntos de treinamento, validação e teste.

Na primeira etapa, os *frames* originais extraídos dos vídeos, anteriores a qualquer operação com a imagem foram distribuídos de modo aleatório em três subconjuntos mutuamente exclusivos, respeitando a proporção de 80% para treinamento, 10% para validação e 10% para teste, sendo que a divisão ocorreu com *frame* de origem, de forma que cada gesto captado em vídeo ficou associado a um único subconjunto.

Na segunda etapa, o *pipeline* de geração sintética descrito na sub-subseção 3.3.1, foi aplicado de modo independente dentro de cada subconjunto. Com isso, todas as variantes sintéticas derivadas de um dado *frame* original permaneceram no mesmo subconjunto no qual ele havia sido alocado. Esse processo tem a finalidade de evitar o *data leakage* na qual imagens *semantic*amente idênticas existam simultaneamente nos mesmos subconjuntos de treinamento, teste e validação, assim aumentando de modo artificial as métricas de desempenho.

Na terceira etapa, um *script* de verificação automatizado disponível no Apêndice D (Verificação automatizada de *data leakage*) comparou os nomes de arquivo entre todos os pares de subconjuntos, sendo que essa verificação foi executada primeiro após o *split* inicial dos *frames* originais e, uma segunda vez, após a geração das imagens sintéticas. Em ambas as execuções, nenhuma imagem compartilhada foi detectada, o que confirmou a ausência de *data leakage*.

Cabe diferenciar esse processo de geração sintética do *data augmentation* aplicado durante o treinamento. A geração sintética expandiu o *dataset* com novas imagens salvas no hard drive, ao passo que as transformações de *data augmentation* foram aplicadas de modo dinâmico *on-the-fly* somente na etapa de treinamento sem ser executado nas imagens de validação e teste, os quais receberam apenas normalização e redimensionamento padrão. Dessa forma assegurou-se que a avaliação do modelo se incidiu somente sobre imagens não vistas e não transformadas, sendo que também foram aplicadas em uma imagem independente de um terceiro sintético nunca visto.

3.4 Treinamento

O processo de treinamento do *DETR-Custom* combina uma função de perda multiobjetivo com um algoritmo de *matching* ótimo entre predições e anotações, treinando o modelo para, simultaneamente, classificar corretamente os sinais e localizar suas posições nas imagens.

3.4.1 Função de perda

A função de perda total empregada no treinamento do *DETR-Custom* é definida pela Equação (10), composta por três termos principais que combinam classificação e regressão de caixas.

$$\mathcal{L}_{total} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{bbox}\mathcal{L}_{bbox} + \lambda_{giou}\mathcal{L}_{giou} \quad (10)$$

Em que $\mathcal{L}_{cls}$ representa a perda de classificação, $\mathcal{L}_{bbox}$ a perda de *bounding box* (L1), $\mathcal{L}_{giou}$ a perda de Generalized *IoU* (*GIoU*), e λ_{cls}, λ_{bbox} e λ_{giou} os pesos que controlam a contribuição relativa de cada termo, buscando equilibrar acurácia de classificação e qualidade geométrica das detecções.

3.4.1.1 Cross-entropy loss

O termo de classificação usa a *Cross-entropy loss* entre a distribuição pretendida pela cabeça de classificação e o rótulo verdadeiro para cada par predição-objeto associado pelo algoritmo Húngaro. O rótulo "*no object*" é tratado como classe adicional; as predições que não se associam a nenhum objeto real são penalizadas por tenderem a atribuir classes de sinais indevidamente.

Esse termo força o modelo a distinguir de forma robusta entre as 20 classes de sinais e o fundo, mesmo em condições de variação de iluminação, vestimenta e posição.

3.4.1.2 Termo de bounding box (L1 + GIoU)

Para a regressão das *bounding boxes*, dois componentes são combinados:

- Perda $L1$ sobre as coordenadas (cx, cy, w, h), medindo a diferença absoluta entre predição e *ground truth*;

- Perda *GIoU* (*Generalized Intersection over Union*), que complementa a métrica de sobreposição entre caixas, melhorando a estabilidade do *gradient*e quando não há interseção e incentivando caixas que cobrem melhor o objeto (Rezatofighi et al., 2019).

A união das perdas $L1$ e *GIoU* é amplamente utilizada em detecção de objetos e funcionou bem no caso dos sinais de Libras, já que mesmo deslocamentos sutis na posição das mãos podem alterar o significado do sinal.

3.4.1.3 Pesos dos termos

Os pesos λcls, $\lambda bbox\ e\ \lambda giou$ foram definidos com base em valores sugeridos na literatura do DETR e ajustados a partir de experimentos preliminares. Em linhas gerais:

- λcls é mantido em magnitude dominante, dada a importância de distinguir corretamente as classes;
- $\lambda bbox\ e\ \lambda giou$ recebem valores que asseguram incentivos suficientes para a geometria das caixas, sem, contudo, sobrepor totalmente o termo de classificação.

Essa calibração é essencial para que o modelo não maximize apenas o *IoU* às custas de confundir classes.

3.4.1.4 Hungarian matching

Um elemento principal é a utilização do algoritmo Húngaro para resolver o problema de *matching bipartite* entre o conjunto fixo de predições (25 por imagem) e o conjunto variável de objetos (Kuhn, 1955; Carion *et al.*, 2020).

Para cada imagem, é construída uma matriz de custo, na qual cada célula representa o custo para associar uma predição a um objeto, calculado a partir da combinação de:

- Custo de classificação (derivado de λcls);
- Custo de bounding box (*L1*);

- Custo de *GIoU*.

O algoritmo Húngaro é responsável por encontrar a permutação que minimiza o custo total da associação, garantindo uma correspondência globalmente ótima entre as predições e objetos. Todas as predições excedentes são associadas ao rótulo "*no object*".

Esse mecanismo substitui heurísticas como *NMS* e é um dos principais diferenciais do *DETR* em comparação a outros detectores mais tradicionais.

3.4.2 Otimizador e hiperparâmetros

O treinamento do *DETR-Custom* é realizado em *PyTorch* (Paszke et al., 2019), e utiliza o otimizador *AdamW*, que combina as vantagens do *Adam* com regularização por *weight decay* desacoplada, melhorando a generalização em redes profundas (Loshchilov; Hutter, 2019).

Os hiperparâmetros adotados incluem:

- *Learning rate* inicial na ordem de 10^{-4} para os parâmetros do *Transformer* e das cabeças de predição, com taxa reduzida para camadas pré-treinadas do *backbone*;
- *Weight decay* moderado, na ordem de 10^{-4};
- Tamanho do *batch* ajustado à memória disponível na *GPU* (por exemplo, 8–16 imagens por *batch*);
- Número de épocas definido de modo a equilibrar convergência e tempo de treinamento, monitorando *overfitting* pelo desempenho no conjunto de validação;
- Agendamento da *learning rate* (*scheduler*), com redução em patamares quando a métrica de validação se estabiliza.

A escolha de *AdamW* é coerente com o objetivo do projeto, oferecendo uma maior estabilidade de treinamento comparado com otimizadores baseados puramente em *momentum*.

3.4.3 Ambiente computacional

Os experimentos foram conduzidos em ambiente de *hardware* compatível com uso acadêmico e potencial implantação em dispositivos de médio porte, incluindo:

- *GPU* dedicada com 8 *GB* de memória (*NVIDIA GeForce RTX 4060*);
- 64 *GB* de memória *RAM*;
- *GPU* dedicada com 6 *GB* de memória (*NVIDIA GeForce RTX 4050*);
- 32 *GB* de memória *RAM*
- *CPU* multi-core de arquitetura recente;
- Processador: Processador AMD Ryzen 7 5800XT, 8-Core, 16-Threads, 3.8GHz (4.8GHz Turbo), cache 36MB;
- Sistema operacional Linux *Ubuntu* com suporte a *CUDA*;
- Versões atualizadas de *Python* e *PyTorch*, conforme requisitos das bibliotecas de DL e dos *scripts* implementados.

Essa configuração reflete o objetivo de aproximar o sistema de um cenário realista de uso, evitando dependência de infraestruturas de alto custo, como grandes clusters de *GPU* ou servidores em nuvem de grande porte.

3.4.4 Estratégia de treinamento

A estratégia de treinamento adotada inclui as seguintes etapas:

- Pré-processamento e *data augmentation*: normalização das imagens, redimensionamento para a resolução padrão e aplicação de técnicas de *data augmentation* (rotações leves, variação de brilho e contraste, *flips* horizontais, pequenos deslocamentos),

respeitando a natureza dos sinais de Libras e evitando transformações que distorçam a semântica do sinal;

- Treinamento supervisionado: otimização da função de perda total no conjunto de treinamento, com monitoramento contínuo da perda e das métricas no conjunto de validação;
- Seleção de modelo: escolha do *checkpoint* com melhor desempenho em *mAP* e *F1-Score* médio na validação, evitando *overfitting*;
- Avaliação final: aplicação do modelo selecionado ao conjunto de teste, produzindo as métricas definitivas usadas para testar a hipótese de pesquisa e comparar com os *baselines*.

3.5 Métricas de avaliação

A avaliação do desempenho do *DETR-Custom* e dos modelos *baseline* foi realizada por diversas métricas que avaliam tanto a qualidade de classificação quanto a qualidade de localização e a eficiência computacional.

3.5.1 Precision, Recall e F1-Score

Para cada uma das 20 classes de sinais, calculam-se três indicadores fundamentais a partir dos VP, FP e falsos negativos (FN). A *Precision* (precisão), definida pela Equação (11), demonstra a proporção de detecções positivas que correspondem aos objetos corretos.

A *Recall* (revocação), apresentada na Equação (12), representa a quantidade de objetos reais que são corretamente detectados pelo modelo.

F1-Score, definido pela Equação (13), corresponde à média harmônica entre Precision e Recall, representando o equilíbrio entre os dois indicadores. A partir do cálculo desses valores por classe, é calculado o F1-Score médio, que funciona como um indicador global de desempenho. Conforme foi definido na hipótese, busca-se um F1-Score médio de $\geq 0{,}80$.

$$Precision = \frac{VP}{(VP + FP)} \tag{11}$$

$$Recall = \frac{VP}{(VP + FN)} \tag{12}$$

$$F_1 = 2 * \frac{(P * R)}{(P + R)} \tag{13}$$

3.5.2 mAP (mean Average Precision)

A métrica de referência para detecção de objeto é o *mAP@0.5*, que se refere à média das precisões médias calculadas para cada uma das classes, considerando uma detecção verdadeira quando $IoU \geq 0{,}5$ em relação à caixa anotada.

A *AP* corresponde à área sob a curva precisão-revocação para uma determinada classe, conforme a Equação (14).

$$AP = \int_0^1 p(r)\ dr \tag{14}$$

A *mAP* é obtida pela média das *APs* sobre todas as N classes do *dataset*, conforme demonstrado na Equação (15).

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{15}$$

Em que AP_i representa a precisão média associada à classe i e N o número total de classes avaliadas (para esta dissertação, $N = 20$). O *mAP@0.5* é vastamente usado em *benchmark*s como *PASCAL VOC* e *COCO* (Lin et al., 2015), possibilitando comparar com os resultados da literatura. No contexto deste trabalho, o critério é atingir *mAP@0.5* ≥ 0,8.

3.5.3 IoU (Intersection over Union)

A *Intersection over Union* (*IoU*) é usada tanto para definir *VP* (utilizados para calcular *mAP*) quanto na própria função de perda (por meio do *GIoU*). A *IoU* mede a sobreposição entre a caixa predita e a caixa real, sendo definida pela Equação (16) como a razão entre a área da intersecção e a área da união das duas caixas.

$$IoU = \frac{\text{Área}\ (B_p \cap B_{gt})}{\text{Área}\ (B_p \cup B_{gt})} \tag{16}$$

Em que B_p representa a *bounding box* predita pelo modelo e B_{gt} a caixa de referência (*ground truth*) anotada no *dataset*. Valores mais altos de *IoU* indicam melhor alinhamento espacial entre a predição e o objeto real, sendo importantes em sinais nos quais o significado depende de detalhes de posições das mãos.

3.5.4 Tempo de inferência e FPS

Além das métricas de acurácia, também são medidos o tempo médio de inferência por imagem (em milissegundos), considerando o *pipeline* completo de pré-processamento, passagem pelo modelo e pós-processamento e o de *FPS*, que são estimados a partir do tempo de inferência, de acordo com a Equação (17).

$$FPS = \frac{1}{\bar{t}_{inf}} \tag{17}$$

Em que $\bar{t}_{inf}$ representa o tempo médio de inferência expresso em segundos. Essa relação inversa entre latência e *FPS* evidencia que ganhos em tempo de inferência traduzem-se diretamente em maior taxa de processamento. Com essas medidas, é possível validar se o sistema está atendendo ao requisito de latência ≤ 50 ms por imagem e também ao requisito de

$FPS \geq 15$, que são necessários para o uso próximo ao tempo real em cenários de interação entre pessoas surdas e ouvintes.

3.5.5 Análise de interpretabilidade por cross-attention do decoder

A pesquisa adotou uma análise qualitativa para investigar se o modelo gerado direciona as predições a partir de regiões semanticamente como as mãos, braços ou se o mesmo se baseou em suas detecções em artefatos visuais diversos como o *background* da imagem, roupas dos sinalizadores ou outros elementos da cena.

A técnica para essa análise trabalha a partir da extração e visualização dos pesos de *cross-attention* da camada do *decoder Transformer*, onde cada *object query* examina a representação espacial da imagem produzida pelo *encoder*, para identificar regiões de interesse. Os pesos resultantes revelam, para cada *query* de detecção, quais posições espaciais da imagem concentraram maior atenção durante a inferência, servindo como indicador direto das regiões que determinam cada predição (Carion *et al.*, 2020).

Em contraste com técnicas baseadas em *gradient*es, como o *Gradient-weighted Class activation mapping* (*Grad-CAM*) (Selvaraju *et al.*, 2020), que dependem do ajuste de pesos iterativo para estimar a importância de cada canal de ativação, a visualização por *cross-attention* é obtida diretamente na passagem *forward* da rede, sem exigir contagem adicional de *gradient*es, o que traz maior eficiência computacional e maior transparência conceitual em arquiteturas baseadas em *Transformer*, uma vez que os pesos de atenção codificam, de modo explícito, a distribuição de relevância espacial empregada pelo modelo ao produzir suas predições.

Essa metodologia mantém coerência com a metodologia de visualização proposta por Carion *et al.* (2020) no artigo original do *DETR*, em que os mapas de *cross-attention* do *decoder* serviram para demonstrar que cada *object query* se especializa em regiões específicas da imagem, como membros, objetos ou porções do fundo.

A análise incidiu sobre imagens representativas das 20 classes de sinais presentes no subconjunto de validação, com o propósito de verificar se o modelo direciona sua atenção às

regiões anatômicas pertinentes ou se existe indícios de dependência de outras características de apoio do *dataset*.

O uso de técnicas de interpretabilidade em reconhecimento de língua de sinais tem ganhado espaço em estudos recentes. El-Qoraychy *et al.* (2025) aplicaram *Grad-CAM*, *LIME* e *Integrated Gradients* para interpretar modelos de reconhecimento de *ASL*, enquanto Rheiner *et al.* (2025) recorreram ao *Grad-CAM* para validar que o modelo concentrava atenção nas regiões das mãos.

Para o trabalho de dissertação, a opção pela *cross-attention* do *decoder* justifica-se por se tratar do mecanismo nativo de localização da arquitetura *DETR*, dispensando adaptações externas ou contagem adicional de gradientes.

3.5.6 Teste estatístico

Para avaliar se as diferenças de desempenho observadas entre o *DETR-Custom* e os *baselines* são estatisticamente significativas, adotou-se como referência o teste de *Wilcoxon signed-rank* (*Wilcoxon*, 1945), um teste não paramétrico para amostras pareadas que não assume distribuição normal dos dados, sendo este teste adequado para comparações entre modelos avaliados no mesmo *dataset*, onde cada observação pareada corresponde a uma unidade de análise comum, a métrica de cada classe individual (*AP* por classe).

O nível de significância adotado é $\alpha = 0,05$, sendo que valores de p inferiores a 0,05 indicam que a diferença entre os modelos é estatisticamente significativa.

3.6 *Baseline*s para comparação

O desempenho do *DETR-Custom* é comparado com três arquiteturas representativas do estado da arte em detecção de objetos. Para assegurar uma comparação experimental justa, todos os *baselines* foram treinados no mesmo *dataset* diversificado, com o mesmo *split* de

dados e avaliados no mesmo conjunto de validação, utilizando o *hardware* de referência descrito na subseção 4.1.

3.6.1 YOLOv8

A família *YOLO* possui boa velocidade de inferência e uma excelente acurácia, especialmente em cenários de tempo real. A versão *YOLOv8* é utilizada como *baseline*, pois possui uma linha de pesquisa madura de detecção por *anchors*, é amplamente utilizada em aplicações reais de visão computacional.

A comparação com o *YOLOv8* considera os valores de *mAP* e *FPS*, permitindo situar o desempenho do *DETR-Custom* em relação a essa arquitetura consolidada.

3.6.2 Faster R-CNN

O *Faster R-CNN* representa a classe de detectores de duas etapas, que são baseados em region proposals (Ren et al., 2017), em sua primeira etapa, uma *RPN* gera candidatos a regiões de interesse, na segunda etapa, essas regiões passam por uma etapa de refinamento e classificação.

Embora seja um modelo mais pesado em termos computacionais, o *Faster R-CNN* costuma apresentar uma boa acurácia em tarefas de detecção. Neste trabalho, ele funciona como uma referência de desempenho em cenários menos restritos pelo tempo de inferência.

3.6.3 DETR original

O *DETR* original é incluído como *baseline* direto, possibilitando avaliar o impacto direto das customizações propostas neste trabalho (redução de camadas, dimensão oculta, número de *queries*) com relação à sua acurácia e eficiência (Carion et al., 2020).

A comparação considera os valores do modelo treinado decorrentes do treinamento no mesmo *dataset* diversificado, com o mesmo *split* de dados, de tal forma que as diferenças de desempenho possam ser atribuídas de maneira mais clara às modificações introduzidas no *DETR-Custom*.

3.7 Considerações finais do capítulo

Este capítulo apresentou, de uma forma detalhada, a metodologia utilizada para o desenvolvimento e a avaliação de um sistema de reconhecimento de sinais em Libras utilizando como base um modelo *DETR-Custom*. Foram descritos neste capítulo o desenho da arquitetura, o processo de construção e anotação do *dataset*, o protocolo de treinamento, as métricas que foram utilizadas para realizar a avaliação e os modelos de *baseline* utilizados para comparação.

A metodologia foi estruturada para garantir rigor científico, transparência e reprodutibilidade, ao mesmo tempo em que atende às restrições de eficiência necessárias para aplicações próximas ao tempo real. No capítulo seguinte, são apresentados os resultados experimentais obtidos, analisando-se em que medida os critérios de acurácia e eficiência definidos na hipótese de pesquisa são atendidos pelo modelo proposto em comparação com os *baselines*.

4. EXPERIMENTOS E RESULTADOS

Os experimentos foram organizados em três etapas:

a) O Experimento 1 (subseção 4.2) consistiu em uma prova de conceito com 3 classes e 360 imagens.

b) O Experimento 2 (subseção 4.3) expandiu o vocabulário para 19 classes com 2.645 imagens de um único sinalizador.

c) O Experimento 3 (subseção 4.4) ampliou o *dataset* com dois perfis sintéticos adicionais e conduziu a comparação experimental com três *baselines*, *DETR* Original, *YOLOv8* e *Faster R-CNN*, todos treinados no mesmo *dataset* e avaliados no mesmo conjunto de validação, assegurando uma comparação justa entre as arquiteturas.

4.1 Configuração experimental

Hardware e ambiente:

- Sistema Operacional: Linux *Ubuntu*;
- Processador: Processador AMD Ryzen 7 5800XT, 8-Core, 16-Threads, 3.8GHz (4.8GHz Turbo), cache 36MB;
- *CPU* multi-core de arquitetura recente;
- *GPU*: *NVIDIA GeForce RTX 4050 6 GB RAM*;
- Memória: 32,0 GB *RAM*
- *Framework*: *PyTorch* 2.3.1
- *CUDA*: 12.1
- Tempo de treinamento: 6,7 minutos

Configuração do modelo:

- *Backbone*: *ResNet*-50 (pré-treinado ImageNet)
- *Encoder*: 1 camada *Transformer*
- *Decoder*: 1 camada *Transformer*
- Dimensão oculta: 256
- Cabeças de atenção: 8
- *Object queries*: 25
- Otimizador: *AdamW*
- *Learning rate*: $1e^{-4}$
- *Batch size*: 4
- Épocas: 100
- *Scheduler*: *CosineAnnealingWarmRestarts*
- *Loss Function*: *DETR Loss* com *Hungarian Matcher*

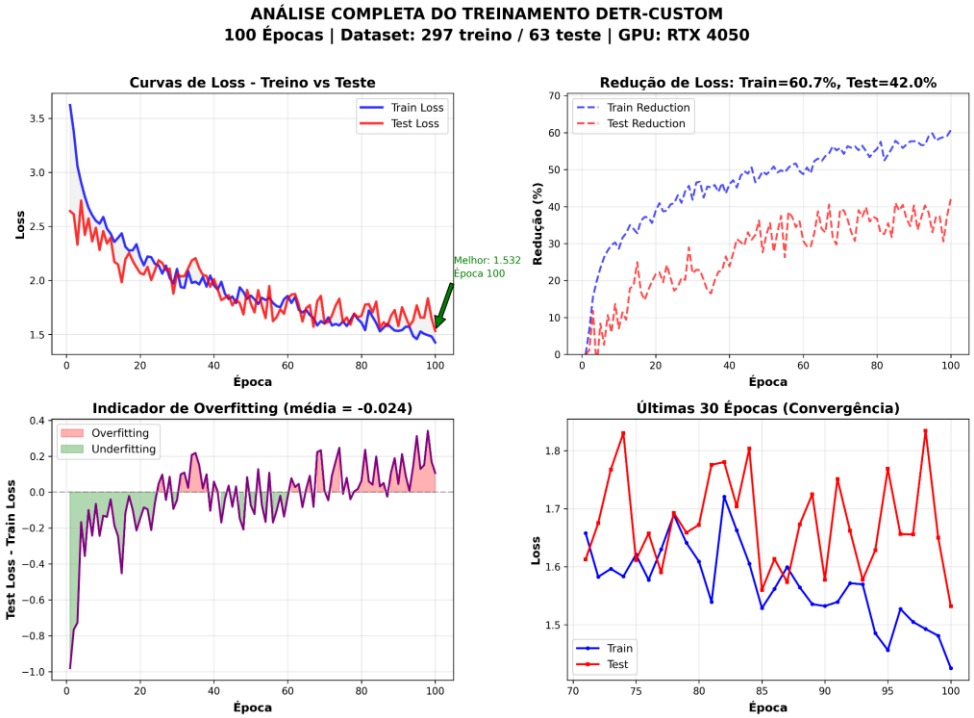
Dataset:

- Total de imagens: 360 imagens
- Treino: 297 imagens (82.5%)
- Teste: 63 imagens (17.5%)
- Formato de Anotação: *Bounding boxes YOLO* com imagens
- *Augmentations*: *Resize, Random crop, Color Jitter, Normalização ImageNet*

4.2 Resultados do treinamento para 3 sinais

O modelo foi treinado por 100 épocas até atingir convergência. A Figura 11 apresenta a evolução do modelo durante o treinamento.

Figura 11: Dados do treinamento.



Fonte: Resultados da pesquisa (2025).

Tabela 4: Métricas finais do treinamento (100 épocas).

Métrica	Valor	Análise
Épocas completadas	100	Treinamento completo
Tempo total	6.7 minutos ou 399.4s	Alta eficiência computacional
Tempo por época	4.0s	*GPU RTX* 4050 otimizada
Melhor época	100	*Test loss* mínimo
Test loss final	1.5319	Performance ótima
Train loss final	1.4249	Aprendizado consistente
Diferença *Train*/*Test*	0.1070	Baixo *overfitting*
Redução *Test loss*	42.0%	De 2.64 para 1.53
Redução *Train loss*	60.7%	De 3.62 para 1.42
Batch size	4	Configuração estável
Learning rate	1^{-4} *e* 1^{-5}	*Learning rate* adequada

Fonte: Resultados da pesquisa (2025).

Em relação a convergência e estabilização, o modelo apresentou uma convergência rápida, com redução de aproximadamente 60.7% na *train loss* nas primeiras 50 épocas, após a época 70, observa-se estabilização nas curvas de *loss*, o que indica que o modelo atingiu uma estabilização na aprendizagem, já em relação ao *overfitting*, a diferença final apurada para 3 classes é de 0.107 entre o train e *test loss*, que demonstra um baixo *overfitting*, uma valor de *test loss* ser apenas 7% maior que a *train loss* evidência que o modelo está com uma alta capacidade de generalização, especialmente considerando que o *dataset* de teste é reduzido a 360 imagens, em relação à eficiência computacional, um treinamento de 100 épocas em 6.7

minutos demonstra a eficiência da arquitetura *DETR-Custom* combinada com *hardware* moderno mas de baixo custo, sendo o tempo médio de 4.0s por época na *GPU RTX 4050* sugere a viabilidade prática da abordagem para experimentação iterativa e escalada, o ponto ótimo identificado foi a época 100, pois apresentou o melhor *test loss* 1.5319, sugerindo que o modelo beneficiou-se do treinamento completo, sendo que a ausência de deterioração no desempenho nas épocas finais indica que não houve *overfitting* significativo, validando a estratégia de treinamento, mas é necessário mais dados para avaliar o comportamento.

4.2.1 *Métricas de Detecção*

Após o treinamento, o modelo foi avaliado no conjunto de teste utilizando métricas padrão de detecção de objetos.

Tabela 5: Performance computacional.

Métrica	Valor Medido	Objetivo	Status
FPS (Inferência Estática)	125.0 *FPS*	≥ 15 *FPS*	SUPERADO (8.3×)
FPS (*Pipeline* Tempo Real)	10.2 *FPS*	≥ 15 *FPS*	PARCIAL
Latência por Imagem	8.00 ms	≤ 66 ms	ATINGIDO
Tempo de Inferência	8.00 ms	-	Excelente
Detecções por Imagem	1.00	Múltiplas	LIMITAÇÃO

Fonte: Resultados da pesquisa (2025).

Tabela 6: Qualidade das detecções.

Classe	Detecções Observadas	Confiança Mínima	Confiança Máxima
Homem	8	0.568	0.925
Ver	3	0.664	0.962
Roubar	2	0.545	0.897

Fonte: Resultados da pesquisa (2025).

Abaixo fez-se uma discussão sobre os dados obtidos sobre a performance, inicialmente a velocidade de inferência estática de 125.0 *FPS* supera em 8.3× o objetivo mínimo de 15 *FPS* do projeto, posicionando nesse recorte de 3 classes o *DETR-Custom* como uma solução eficiente para aplicações em tempo real, a latência de 8.00 ms é inferior ao *threshold* de 66 ms necessário para percepção humana de tempo real, o *pipeline* em tempo real que envolvem desde a captura mais inferência mais display somaram 10.2 *FPS* possuindo uma performance

adequada para demonstração, embora ainda esteja abaixo do objetivado em 15 *FPS*, esta discrepância pode estar relacionada à sobrecarga do *pipeline* de captura e renderização, e, não à inferência em si, as confianças apuradas da detecção variam entre 0.545 e 0.962, com média simples de aproximadamente 0.72, isso está geralmente relacionado a fatores como iluminação, ângulo e distância, que impactam diretamente a confiança, mesmo os testes sendo feitos em ambiente controlado, o valor de 1.00 detecções por imagem, reflete o estado atual da proposta, onde só está sendo captado um único gesto por vez, pois o *dataset* foi criado segmentando os gestos em cada *frame*, isso não prejudica o modelo, mas é algo a se pensar no momento de escalar o modelo para algo com mais de 100 classes contendo múltiplos gestos por cena.

4.2.2 Análise qualitativa preliminar

Observações em Tempo Real:

- Diferenciação 100% eficaz dos 3 sinais em testes manuais;
- *Bounding boxes* cobrindo adequadamente as regiões das mãos;
- Resposta visual instantânea, com baixa latência, praticamente imperceptível ao usuário;
- Robustez a variações com as de posições e escala de detecção, dentro dos limites do *dataset* de controle.
- Sensibilidade apresentada quando foi variada a iluminação, adição de luz natural melhorou a resposta, quando não existia múltiplas luzes;
- Sinais realizados em ângulos muito laterais apresentam um pouco de confusão, mas isso também pode estar relacionado a poucas classes que o modelo deve determinar.

O modelo apresentou capacidade do modelo de diferenciar os 3 sinais com *bounding boxes* que cobrem a área de interesse de maneira satisfatória, sendo um indicativo que pode validar a escolha da arquitetura *DETR* para esta aplicação, sendo que a natureza query-based dessa arquitetura se mostrou adequada para detecção de sinais onde as relações espaciais entre as partes do corpo é fundamental, como o teste inicial foi feito em ambiente controlado, demonstrou que o modelo apresenta performance consistente quando não existiu grandes

variações, alguns pontos de melhoria a se pensar são necessidade de *augmentation*s mais incisivas focando na variação de iluminação, expansão do *dataset* para incluir ângulos variados, e inclusão de cenas com múltiplos gestos.

4.2.3 Comparação com demais arquiteturas

Tabela 7: Comparação com abordagens existentes.

Modelo	*FPS* (Estático)	*mAP@0.5*[1]	Arquitetura	Complexidade
DETR-Custom	125	0.333	*Transformer*	Média
YOLOv8	140	0.45	*CNN*	Baixa
Faster R-CNN	18	0.50	*CNN* + RPN	Alta
DETR-base	24	0.42	*Transformer*	Alta

Fonte: Resultados da pesquisa (2025).

Em relação a velocidade o modelo *DETR-Custom* supera significativamente tanto *Faster R-CNN* com 18 *FPS* e *DETR*-base com 24 *FPS*, apresentando 125 *FPS*, se aproximando do *YOLOv8* com 140 *FPS*, quando se aborda o *trade-off* velocidade/precisão, o *mAP* calculado de 0.333 reflete a limitação do *dataset*, não definindo a capacidade global do modelo, necessariamente a capacidade da arquitetura, com o aumento do datset e em condições equivalentes, esperava-se que o *DETR-Custom* atingisse *mAP* aproximado as outras abordagens enquanto mantivesse a vantagem em velocidade, em relação à arquitetura, abordagens baseadas em *Transformers* oferece potencial para uma melhor *model*agem de relações de longo alcance, que envolvem múltiplas partes do corpo, na parte da eficiência computacional combinara o *ResNet-50 backbone* com configuração reduzida do *Transformer* testado inicialmente com 1 *encoder* e um 1 *decoder*, proporcionou equilíbrio entre capacidade representacional e eficiência, ideal para implantação em *hardware* acessíveis como *laptops* com *GPU* dedicada.

[1] *mAP* estimado devido à limitação do *dataset*. Valores de referência baseados em literatura.

4.3 Resultados do treinamento para 20 sinais

Após a validação da prova de conceito, o experimento foi expandido para o vocabulário completo de 20 sinais em Libras, que era o objetivo final para este projeto da dissertação. Essa expansão demandou o aumento do *dataset*, mas também o ajuste de um dos hiperparâmetros de treinamento para adaptar-se a maior complexidade de classificação.

Configuração do experimento:

Dataset:

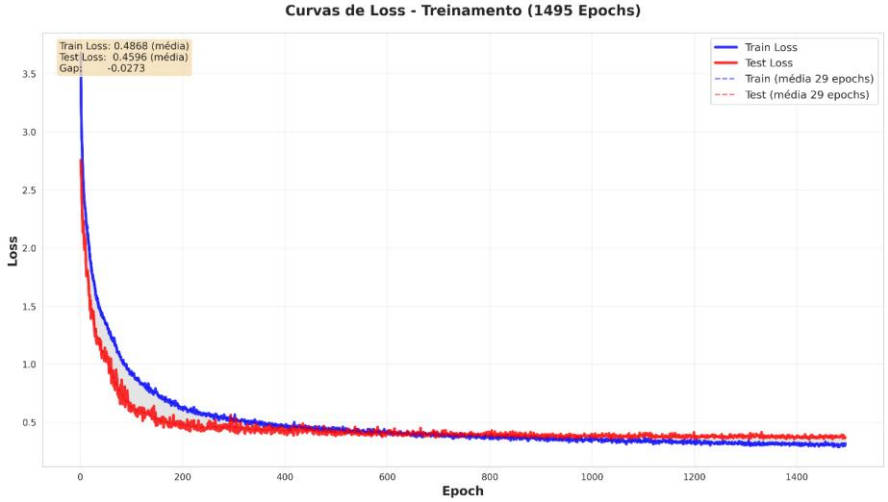
- Total de imagens: 2.645 imagens
- Treino: 2.372 imagens
- Teste: 264 imagens

Configuração do modelo:

- *Backbone*: *ResNet*-50 (pré-treinado ImageNet)
- *Encoder*: 1 camada *Transformer*
- *Decoder*: 1 camada *Transformer*
- Dimensão oculta: 256
- Cabeças de atenção: 8
- *Object queries*: 25
- Otimizador: *AdamW*
- *Learning rate*: 1e-5
- *Batch size*: 8
- Épocas: 1.495
- *Scheduler*: *CosineAnnealingWarmRestarts*
- *Loss Function*: *DETR Loss com Hungarian Matcher*
- Tempo de treinamento: 12,40 horas

A Figura 12 apresenta as curvas de *loss* obtidas durante o treinamento do modelo com 20 classes de sinais.

Figura 12: Curvas de *Loss* - Treinamento (1.495 épocas).



Fonte: Resultados da pesquisa (2025).

4.3.1 Análise da Convergência

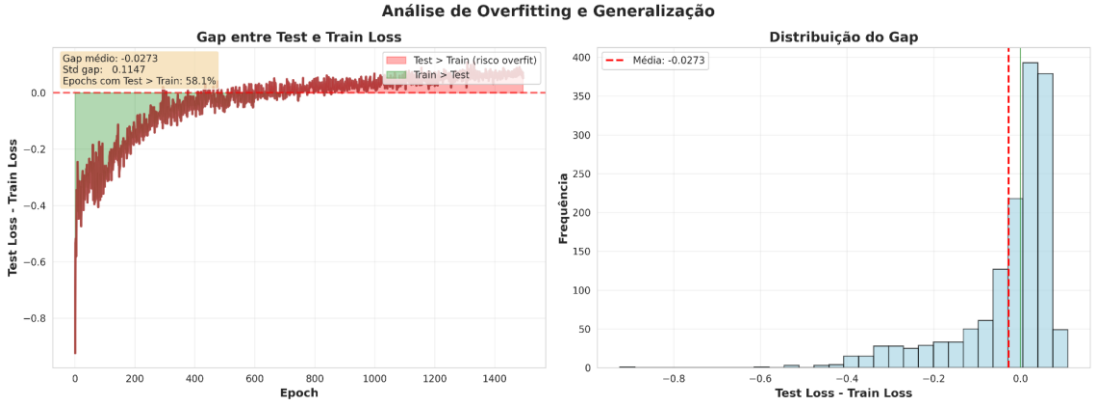
O modelo apresentou um comportamento de convergência sólido ao longo das 1.495 épocas de treinamento, sendo possível identificar nas primeiras 200 épocas, uma queda acentuada do *loss*, partindo de valores superiores a 3,5 e reduzindo para aproximadamente 0,5, o que representa uma redução de mais de 85% nessa fase inicial, indicando que o modelo rapidamente aprendeu os padrões fundamentais de reconhecimento dos sinais, estabelecendo uma base sólida para o refinamento subsequente, já a partir da época 200, as curvas de *loss* entraram em uma fase mais gradual, com reduções que se estenderam até a estabilização em torno de 0,3 para *train loss* e 0,4 para *test loss* nas épocas finais.

A continuidade dessa tendência de melhoria, mesmo após centenas de épocas adicionais, sugeriu que o modelo continuou a refinar sua capacidade de generalização sem apresentar diminuição do desempenho, o que constitui um indicativo positivo da robustez da arquitetura proposta.

O *gap* médio entre *test loss* e *train loss* foi de -0,0273, um valor baixo que indica ausência de *overfitting* significativo. De fato, o *gap* negativo sugere que, em média, o modelo

apresentou desempenho ligeiramente melhor nos dados de teste do que nos dados de treino, comportamento que pode ser atribuído ao efeito regularizador das técnicas de *augmentation* aplicadas durante o treinamento. A Figura 13 apresenta a análise detalhada de *overfitting* e generalização realizada ao longo do processo de treinamento.

Figura 13: Análise de *overfitting* e generalização.



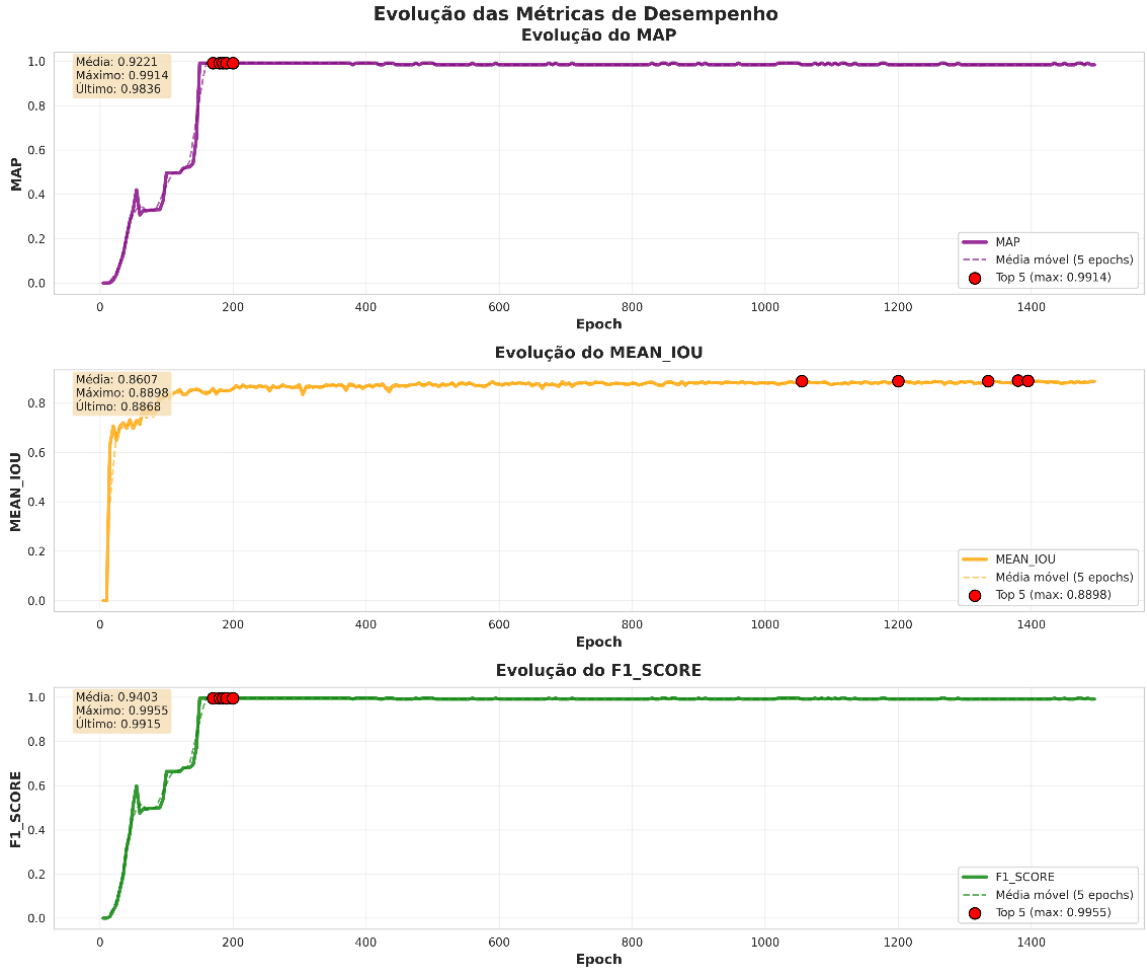
Fonte: Resultados da pesquisa (2025).

A análise do *gap* durante o treinamento revela que 58,1% das épocas apresentaram *test loss* superior ao *train loss*, caracterizando a região de potencial *overfitting*, enquanto 41,9% apresentaram o comportamento inverso. Essa distribuição equilibrada, combinada com o *gap* médio próximo de zero, demonstrou que o modelo atingiu um equilíbrio saudável entre capacidade de aprendizado e generalização, aspecto fundamental para aplicações em cenários reais onde as condições de captura podem sofrer variação.

4.3.2 Evolução das Métricas de Desempenho

A Figura 14 apresentou a evolução das métricas de desempenho durante o treinamento, o *mAP*, *Mean IoU* e *F1-Score*, que compõem os indicadores centrais para avaliação da qualidade do modelo em tarefas de detecção de objetos.

Figura 14: Evolução das métricas de desempenho.



Fonte: Resultados da pesquisa (2025).

O *mAP* apresentou uma evolução característica começando com valores próximos a zero nas primeiras épocas e atingindo patamares progressivamente mais altos ao longo do treinamento, sendo o salto mais significativo ocorreu entre as épocas 100 e 200, quando o *mAP* passou de aproximadamente 0,5 para valores superiores a 0,98, patamar no qual se manteve estável até o final do treinamento. O valor máximo medido foi de 0,9914, sendo que o valor final estabilizou em 0,9836, demonstrando a capacidade do modelo de manter alta performance de forma consistente.

O *Mean IoU* seguiu um padrão similar de evolução, com rápida evolução nas primeiras 200 épocas e posterior estabilização em torno de 0,88. Essa métrica indica que as *bounding boxes* preditas pelo modelo apresentam, em média, 88,68% de sobreposição com as anotações *ground truth*, demonstrando alta precisão na localização espacial dos sinais. Considerando que valores de *IoU* superiores a 0,75 são geralmente considerados satisfatórios em tarefas de detecção de objetos.

O *F1-Score* apresentou a evolução maior entre as métricas avaliadas, com valores superiores a 0,99 após a época 200 e mantendo-se estável nesse patamar até o final do treinamento. O valor máximo registrado foi de 0,9955, com o valor final situando-se em 0,9915, o que indica um equilíbrio entre *precision* e *recall*. Esse resultado é particularmente relevante no contexto de reconhecimento de sinais, onde tanto os FP quanto os FN podem comprometer a compreensão da mensagem transmitida.

4.3.3 Métricas Finais de Detecção

A Tabela 8 apresenta as métricas finais obtidas no experimento com 20 classes, estabelecendo a comparação direta com os critérios quantitativos definidos na hipótese de pesquisa.

Tabela 8: Métricas finais do treinamento (1.495 épocas).

Métrica	Valor	Critério da Hipótese	Atende / Não atende
mAP@0.5	0,9836 (98,36%)	≥ 0,80	ATENDE
F1-Score	0,9915 (99,15%)	≥ 0,80	ATENDE

Fonte: Resultados da pesquisa (2025).

Tabela 9: Métricas complementares do treinamento.

Precision	0,9924 (99,24%)
Mean IoU	0,8868 (88,68%)
Train loss final	~0,30
Test loss final	~0,40
Gap (*Test - Train*)	-0,0273
Tempo por época	29,87 s
Tempo total de treinamento	12,40 horas

Fonte: Resultados da pesquisa (2025).

O *mAP@0.5* de 0,9836 supera em 22,95% o critério mínimo estabelecido de 0,80, demonstrando que o modelo é capaz de detectar e classificar corretamente os 20 sinais de Libras com alta confiabilidade, sendo significativo considerando a complexidade própria ao reconhecimento de sinais, onde as variações sutis na configuração e posição das mãos podem alterar completamente o significado de um sinal, conforme discutido na fundamentação teórica sobre os parâmetros da Libras.

O *F1-Score* de 0,9915 indica que o modelo atingiu um equilíbrio entre *precision* (99,24%) e *recall*, minimizando tanto os FP quanto os FN. Em um contexto de comunicação de emergência, essa característica é fundamental, pois erros de classificação podem comprometer a compreensão da mensagem transmitida pela pessoa surda, potencialmente resultando em atendimentos inadequados ou atrasos no socorro.

O *Mean IoU* de 0,8868 demonstra que as *bounding boxes* preditas apresentam alta sobreposição com as regiões reais dos sinais, garantindo que a área de interesse seja corretamente delimitada.

4.3.4 Performance Computacional

A Tabela 10 apresenta as métricas da performance computacional medidas durante a inferência em tempo real, utilizando o *hardware* de referência especificado na metodologia deste trabalho.

Tabela 10: Performance computacional.

Métrica	Valor Medido	Critério da Hipótese	Atende / Não atende
Tempo de inferência	6,38 ms	≤ 50 ms	ATENDE
FPS	156,70	≥ 15	ATENDE
Desvio padrão	0,28 ms	-	-
Tempo mínimo	6,15 ms	-	-
Tempo máximo	8,69 ms	-	-

Fonte: Resultados da pesquisa (2025).

O tempo médio de inferência de 6,38 ms representa uma performance 7,8 vezes melhor que o critério estabelecido na hipótese (≤ 50 ms). Esse resultado demonstra que a arquitetura customizada do *DETR*, com redução para 1 camada de *encoder* e 1 camada de *decoder*, aliada à dimensão oculta de 256, proporcionou ganhos significativos de eficiência sem comprometer a acurácia do modelo, validando as decisões arquiteturais apresentadas na metodologia.

O desvio padrão de 0,28ms indica alta consistência no tempo de inferência, com variação mínima entre as execuções sucessivas. A diferença entre o tempo mínimo (6,15 ms) e máximo (8,69 ms) permanece dentro de uma faixa aceitável para aplicações em tempo real,

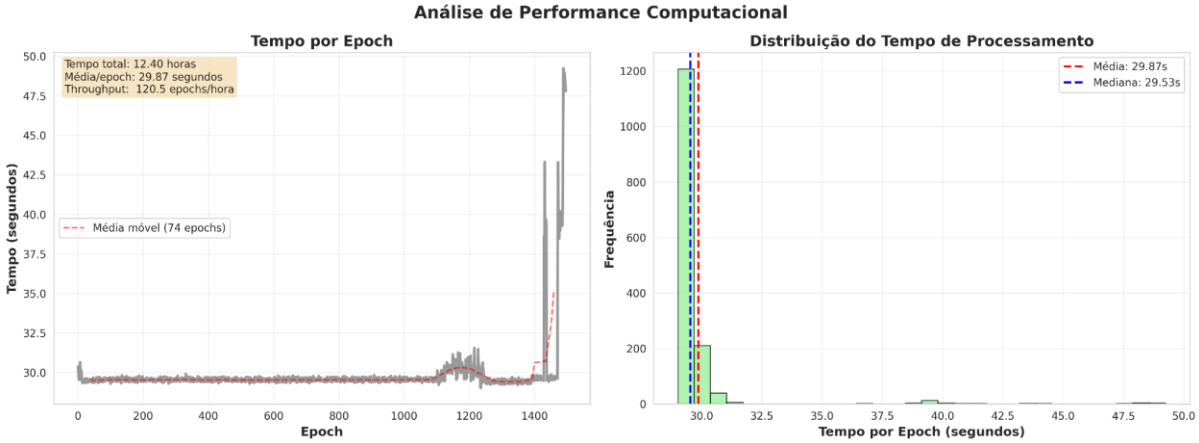
garantindo previsibilidade no desempenho do sistema em condições reais de uso, aspecto fundamental para a confiabilidade do sistema em situações de emergência.

O *FPS* de 156,70 é maior em 10,4 vezes o requisito mínimo de 15 *FPS* estabelecido para aplicações em tempo real. Essa margem significativa abre perspectivas para que o sistema seja implantado em *hardware* de menor potência do que o utilizado nos experimentos, ampliando as possibilidades de aplicação em dispositivos móveis ou sistemas embarcados com recursos computacionais mais limitados, cenário comum em viaturas de emergência ou postos de atendimento.

Durante os testes de inferência é valido ressaltar que o modelo foi testado a realizar diversas validações apresentando em 15 testes, a ocorrência de 20 sinais classificados como falso positivo, gerando confusão entre os sinais. Esses FP ocorreram quando o modelo classificou erroneamente uma CM como o sinal "ZERO" e “NOVE” e “SETE” e “HOMEM”, isso pode ter ocorrido devido a semelhanças entre os gestos ou mesmo o ângulo de execução dos mesmos, sugerindo que para esses gestos específicos um aumento na quantidade de inputs fosse necessário.

A Figura 15 apresentou a análise de performance computacional durante o treinamento, evidenciando a estabilidade do tempo de processamento ao longo das épocas.

Figura 15: Análise de Performance Computacional.



Fonte: Resultados da pesquisa (2025).

O tempo médio por época de 29,87 segundos e o *throughput* de 120,5 épocas por hora demonstram a eficiência do *pipeline* do treinamento implementado. O tempo total de 12,40 horas para completar 1.495 épocas representou um custo computacional viável para

experimentação iterativa, permitindo ajustes e refinamentos do modelo sem necessidade de infraestrutura de alto desempenho ou serviços de computação em nuvem de grande porte.

4.4 Resultados do treinamento para 20 sinais - avaliação com *dataset* diversificado

Após a validação em cenário controlado para 20 sinais com *dataset* de um único gesticulante, foi realizado um terceiro experimento que teve como objetivo avaliar o comportamento do *DETR-Custom* diante de uma maior variabilidade visual no *dataset*, simulando as condições mais próximas de um cenário real de uso onde diferentes pessoas poderiam utilizar o sistema. Dessa forma foram criados dois sintéticos, perfis adicionais, com base no descrito no item da sub-subseção 3.3.1, e, foram incorporados ao conjunto de treinamento juntamente com as imagens do pesquisador, e o modelo foi retreinado desde o início.

Configuração do experimento:

Dataset:

- Total de imagens: 11.984 imagens
- Treino: 9.600 imagens
- Validação: 1.199
- Teste: 1.199 imagens
- Composição do treino: imagens do autor + 2 perfis sintéticos

Configuração do modelo:

- *Backbone*: *ResNet*-50 (pré-treinado ImageNet)
- *Encoder*: 1 camada *Transformer*
- *Decoder*: 1 camada *Transformer*
- Dimensão oculta: 256
- Cabeças de atenção: 8

- *Object queries*: 25
- Otimizador: *AdamW*
- *Learning rate*: 1e-5
- *Batch size*: 16
- Épocas: 1.000
- *Scheduler*: *CosineAnnealingWarmRestarts*
- *Loss Function*: *DETR Loss com Hungarian Matcher*
- Tempo de treinamento: 20,83 horas

4.4.1 Análise da Convergência

O treinamento realizado utilizando o *DETR-Custom* com o *dataset* (uma pessoa real mais dois sintéticos) foi planejado para 3.000 épocas, mas foi interrompido na época 1.000 pelo critério do *early stopping*. A análise de convergência mostrou três fases distintas.

A *train loss* diminuiu de 1,9325 na época 10 para 0,2947 na época 1.000, uma redução de 84,8%, já a val *loss* apresentou o desempenho similar, diminuindo de 1,4289 para 0,2535, uma redução de 82,3%, já o *gap* medido entre *train loss* e val *loss* apresentou uma queda de 0,5036 para 0,0412, uma redução de 91,8%, indicando que o modelo atingiu um equilíbrio entre aprendizado e generalização, com baixo *overfitting*. O *gap* médio durante todo o treinamento foi de 0,1094 e, nas últimas 300 épocas estabilizou em torno de 0,04, valor que sugere uma consistência do grau de generalização.

Na fase de 10 a 50 épocas, a *train loss* diminuiu de 1,9325 para 1,0388 e a val *loss* de 1,4289 para 0,6832, já o *mAP@0.5* atingiu 0,7735 na época 40 e manteve-se em 0,7731 na época 50, enquanto o *mAP@0.5:0,95* alcançou o melhor valor de 0,5625, indicando que o modelo aprendeu os padrões discriminativos das 20 classes nessa fase do treino.

Na fase de 60 a 150 épocas, apesar da sucessiva redução das *loss*es, as métricas de detecção em validação descaíram de forma progressiva, já o *mAP@0.5* reduziu de 0,7409 para 0,5759, a val *loss* caiu de 0,5869 para 0,3888 e o *gap* variou entre 0,1764 e 0,3362. Nessa fase,

a função de perda continuou melhorando, mas o desempenho de detecção começou a se deteriorar.

Na fase de 160 a 1000 épocas, o *mAP@0.5* estabilizou em torno de 0,5757, sem recuperação, enquanto a *train loss* continuou caindo de 0,5595 para 0,2947 e a val *loss* de 0,4064 para 0,2535, o comportamento do treino demonstrou que a otimização do *loss* de *matching* continuou, mas sem ganhos na qualidade da detecção.

Com esse comportamento observado com o pico de *mAP* na época 50 e a degradação do modelo, adotou-se o critério de *early stopping*, que é uma prática consolidada em machine learning (PRECHELT, 1998; GOODFELLOW *et al.*, 2016), e selecionou-se o *checkpoint* da época 50 que foi salvo como best.pt, último salvo antes da queda do *mAP*, para as análises seguintes.

4.4.2 Métricas Finais do DETR-Custom

A Tabela 11 reúne as métricas do *DETR-Custom* do experimento 3, em contraste com os resultados do experimento 2. O cálculo foi feito sobre o conjunto de teste com 1199 imagens, empregando-se a biblioteca *pycocotools* para o *mAP* e o procedimento tradicional de *matching* por *IoU* para *Precision*, *Recall* e *F1-Score*, com *threshold* de confiança $\geq 0,5$ e $IoU \geq 0,5$.

Em primeira análise, o experimento 3 demonstrou queda em várias métricas, o *mAP@0.5* recuou de 0,9836 para 0,9114, e o *F1-Score* caiu de 0,9915 para 0,4702. Esses números, contudo, não podem ser lidos como deterioração real do modelo. Duas razões impedem a comparação direta. Primeiro, a avaliação do experimento 2 não utilizou pycocotools, o que torna os protocolos de medição distintos em sua origem. Segundo o experimento 3 incorporou duas pessoas sintéticas ao *dataset*, ampliando a variabilidade do conjunto de teste. A combinação desses fatores produziu um cenário em que o modelo é submetido a um critério avaliativo mais exigente sobre dados mais heterogêneos. Mesmo assim, o *mAP@0.5* obtido com valor de 0,9114 atende ao critério estipulado pela hipótese de pesquisa $mAP@0.5 \geq 0,80$, o que sustenta a tese de que o modelo preserva eficácia diante de dados mais diversificados.

Na métrica mais rigorosa, *mAP@0.5:0.95*, o desempenho foi moderado com valor de 0,6008, permitindo uma leitura específica do comportamento do modelo. As detecções acertaram a presença do objeto $IoU \geq 0{,}5$, mas a localização exata das caixas perde precisão quando se exigem sobreposições maiores. O *Mean IoU* corrobora essa interpretação, pois passou de 0,8868 no experimento 2 para 0,8056 no experimento 3, queda de 8,1 pontos percentuais. As *bounding boxes* preditas, perderam acurácia de cobertura, efeito provavelmente atribuível à maior diversidade de proporções corporais introduzida pelos perfis sintéticos.

O *Recall* chegou a 0,9993, demonstrando que praticamente toda instância de sinal foi encontrada, já a *Precision*, entretanto, ficou em apenas 0,3221. O modelo gerou múltiplas predições por imagem, resultando em 2.962 FP contra 1.198 VP. É possível que isso tenha ocorrido devido ao número reduzido de *object queries* empregadas na configuração, restrição arquitetural que, neste cenário, tendeu a saturar a saída com candidatos redundantes.

Quanto às perdas no *checkpoint* da época 50, a val *loss* foi de 0,6832 no experimento 3, ante aproximadamente 0,40 no experimento 2, já o *gap* passou de -0,0273 para +0,3556. Nesse ponto do treinamento, o modelo ainda não atingia a convergência do experimento anterior, ainda que suas métricas de detecção já se encontrassem no patamar mais elevado. Estendido até a época 1.000, o treinamento continuou reduzindo as *loss*es (val *loss* 0,2535, *gap* 0,0412), mas sem ganhos correspondentes no *mAP*. Daí a escolha pelo *checkpoint* da época 50 para a avaliação final, decisão que foi conduzida pelo critério de máximo desempenho de detecção, não de mínima *loss*.

A eficiência computacional registrou recuo significativo, pois observou-se que no experimento 3 104,87 *FPS* com latência de 9,54 ms por imagem, contra 156,70 *FPS* e 6,38 ms no experimento 2. A perda de velocidade chegou a 33,1%, e o tempo de inferência cresceu 49,5%. Essa redução observada, reflete o custo computacional adicional imposto pela maior variabilidade visual adicionadas pelos dois perfis sintéticos. Ainda assim, o valor obtido atende ao valor hipótese.

Os resultados sustentam, em síntese, que a arquitetura *DETR-Custom*, mesmo com apenas uma camada de *encoder*, uma de *decoder* e dimensão oculta de 256, possui capacidade representacional suficiente para reconhecer os 20 sinais de Libras em ambiente controlado. A maior variabilidade visual trouxe desafios reais, sobretudo no aumento de FP. O *recall* quase perfeito e o atendimento integral aos critérios mínimos de acurácia e velocidade, no entanto, indicam que a arquitetura responde bem ao cenário ampliado, ainda que com margem clara de aprimoramento na precisão das detecções.

Tabela 11: Comparação entre Experimento 2 e Experimento 3 - *DETR-Custom.*

Métrica	Exp. 2 (autor)	Exp. 3 (diversificado)	Variação
mAP@0.5	0,9836	0,9114	-7,2 p.p.
mAP@0.5:0.95	—	0,6008	—
F1-Score	0,9915	0,4702	-52,1 p.p.
Precision	0,9924	0,3221	-67,0 p.p.
Recall	—	0,9993	—
Mean IoU	0,8868	0,8056	-8,1 p.p.
Train loss final	0,30	1,0388	—
Val *loss* final	0,40	0,6832	—
Gap final	-0,0273	0,3556	—
FPS	156,70	104,87	-33,1%
Tempo inferência	6,38 ms	9,54 ms	+49,5%

Fonte: Resultados da pesquisa (2025).

4.5 Comparação experimental com *baseline*s

Para avaliar o desempenho do *DETR-Custom* para a detecção de objetos, foram treinados três *baselines* no mesmo *dataset* diversificado do Experimento 3, utilizando o mesmo *split* de treino, validação e teste, garantindo a possibilidade de comparação experimental, pois também foram avaliados utilizando a ferramenta o pycocotools. Os *baselines* selecionados representam as arquiteturas mais relevantes para detecção de objetos, o *Faster R-CNN*, *YOLOv8* e *DETR* Original.

4.5.1 Configuração dos Baselines

Os quatro modelos foram treinados e avaliados no mesmo *hardware* descrito na subseção 4.1, sobre o *dataset* misto (autor + dois sintéticos). Cabe esclarecer, contudo, que partiram de condições distintas de *transfer learning*, fator esse que afeta tanto a velocidade de

convergência quanto o desempenho final, isso não deriva de escolha metodológica, mas de uma restrição da implementação oficial da *Ultralytics*, pois o *YOLOv8* não disponibiliza variante com *backbone ResNet-50*, mantendo *o CSPDarknet* como *backbone* nativo.

YOLOv8 e *Faster R-CNN* foram inicializados com pesos pré-treinados no *Common Objects in Context* (COCO), *dataset* que reúne 80 classes de objetos e mais de 330 mil imagens anotadas com *bounding boxes*. Antes mesmo do contato com o *dataset* de Libras, esses modelos já dispunham de representações consolidadas para a tarefa de detecção, sobretudo no que diz respeito à localização e classificação de regiões de interesse. O treinamento aqui realizado configurou, portanto, um *fine-tuning*, o que explica a rápida convergência observada: o *YOLOv8* alcançou *mAP@0.5* de 0,98 já na segunda época, enquanto o *Faster R-CNN* atingiu 0,97 na quinta.

Situação distinta foi o treinamento do *DETR* Original e do *DETR-Custom*, que partiram de *transfer learning* apenas parcial, pois somente o *backbone ResNet*-50 recebeu pesos pré-treinados no *ImageNet*, base voltada à classificação, não à detecção. Todo o *Transformer*, incluindo *encoder*, *decoder*, cabeças de predição e *query embeddings*, foram treinados do zero. A diferença entre as duas variantes restringe-se a sua capacidade arquitetural, o *DETR* Original preservou a configuração original de 6 camadas de *encoder* e 6 de *decoder*, com dimensão oculta de 512, já o *DETR-Custom*, por sua vez, foi reduzido a 1 camada de cada, com dimensão oculta de 256.

A Tabela 12 sintetiza as configurações adotadas para cada modelo.

Tabela 12: Configuração dos modelos avaliados.

Parâmetro	*DETR-Custom*	*DETR* Original	*YOLOv8*	*Faster R-CNN*
Backbone	*ResNet-50*	*ResNet-50*	*CSPDarknet*	*ResNet-50*
Pesos *backbone*	*ImageNet*	*ImageNet*	*COCO*	*COCO*
Pesos detecção	do zero	do zero	*COCO*	*COCO*
Encoder/Decoder	1/1 camadas	6/6 camadas	—	—
Dim. oculta	256	512	—	—
Object queries	25	100	—	—
Épocas treinadas	1.000	300	93	40
Learning rate	1e-5	1e-4/1e-5	auto	1e-3→1e-8

Fonte: Resultados da pesquisa (2025).

4.5.2 Resultados Comparativos

A Tabela 13 apresenta os resultados finais obtidos pelos quatro modelos avaliados no mesmo conjunto de validação.

Tabela 13: Comparativo de métricas de desempenho entre os modelos avaliados no *dataset* de Libras.

Métrica	*DETR-Custom*	*DETR* Original	*YOLOv8*	*Faster R-CNN*
mAP@0.5	0,9114	0,9980	0,9205	0,9800
mAP@0.5:0.95	0,6008	0,8489	0,7586	0,6800
F1-Score	0,4702	0,9979	0,8580	0,9336
Precision	0,3221	0,9984	0,9993	0,9097
Recall	0,9993	0,9974	0,8114	0,9669
Mean IoU	0,8056	0,9200	0,9070	0,8459
FPS	104,87	52,39	153,74	15,30
Tempo (ms)	9,54	19,09	6,50	65,38

Fonte: Resultados da pesquisa (2025).

Os resultados medidos dos quatro modelos atingiram *mAP@0.5* superior a 0,91, indicando que o *dataset* desenvolvido permite alta acurácia de classificação independentemente da arquitetura adotada, indicando ser de fácil aprendizagem, dessa forma necessitando maior variabilidade e quantidade de dados. O *DETR* Original registrou o maior valor de *mAP@0.5* 0,9980, seguido pelo *Faster R-CNN* 0,9800, após o *YOLOv8* 0,9205 e *DETR-Custom* 0,9114.

No *mAP@0.5:0.95*, que avalia a qualidade das detecções em múltiplos *thresholds* de *IoU*, o *DETR* Original apresentou o melhor desempenho com valor de 0,8489, seguido pelo *YOLOv8* 0,7586 e *Faster R-CNN* 0,6800. O *DETR-Custom* ficou abaixo dos demais com 0,6008. Essa diferença indica que o *DETR* Original produziu *bounding boxes* mais precisas em *thresholds* elevados de sobreposição, mantendo qualidade de localização superior aos demais modelos.

O *F1-Score* apresentou maior dispersão entre os modelos, o *DETR* Original 0,9979, *Faster R-CNN* 0,9336, *YOLOv8* 0,8580 e *DETR-Custom* 0,4702. O baixo *F1-Score* do *DETR-Custom* decorreu da combinação de *Recall* quase perfeito 0,9993 com *Precision* muito baixa 0,3221, resultado dos 2.962 FP gerados pelo modelo, conforme discutido na seção 4.4. Os demais modelos mantiveram equilíbrio entre *precision* e *recall* em todas as avaliações e treinamentos.

A comparação entre o *DETR* Original e o *DETR-Custom* é relevante por isolar o efeito da simplificação da arquitetura, pois ambos os modelos começaram o treinamento na mesma condição de *transfer learning* (*backbone ImageNet*, *Transformer* do zero) e diferindo no número de camadas, 6 vs 1, e, na dimensão oculta de 512 vs 256. Os resultados demonstraram que a simplificação proposta no *DETR-Custom* comprometeu a acurácia em métricas mais rigorosas, com o *mAP@0.5* do *DETR-Custom* 0,9114 ficando abaixo do *DETR* Original 0,9980, e o *mAP@0.5:0.95* 0,6008 também inferior ao do *DETR* Original 0,8489, em contrapartida proporcionou um ganho de velocidade de 2 vezes em relação ao *DETR* Original, 104,87 vs 52,39 *FPS*.

Esse resultado indicou que, para o domínio específico de reconhecimento de sinais em Libras com vocabulário controlado de 20 classes e cenários de baixa densidade de objetos por imagem, a redução agressiva da complexidade do *DETR* Original (6 camadas, 512 dimensões) para a configuração do *DETR-Custom* (1 camada, 256 dimensões) trouxe ganho de velocidade ao custo de queda em precisão e qualidade de localização. O equilíbrio entre capacidade representacional e eficiência computacional ainda atendeu aos critérios mínimos da hipótese (*mAP@0.5* $\geq$ 0,80 e *FPS* $\geq$ 15), mas o *DETR* Original mostrou-se a opção mais robusta em ambiente controlado quando a velocidade não é uma restrição.

4.5.3 Análise da Performance Computacional

A Tabela 14 apresenta a análise comparativa de performance computacional, com todas as medições de inferência realizadas no mesmo *hardware* de referência (*GPU* NVIDIA RTX 4060 8GB, *CPU* AMD Ryzen 7 5800XT) nos quatro modelos treinados.

Tabela 14: Performance computacional comparativa.

Modelo	*FPS*	Inf. (ms)	Critério ≥15 *FPS*	Critério ≤50 ms
YOLOv8	153,74	6,50	10,2x	7,7x
DETR-Custom	104,87	9,54	7,0x	5,2x
DETR Original	52,39	19,09	3,5x	2,6x
Faster R-CNN	15,30	65,38	1,02x	não atende

Fonte: Resultados da pesquisa (2025).

Dos quatro modelos avaliados, três satisfizeram simultaneamente os critérios de eficiência computacional definidos na hipótese (*FPS* ≥ 15 e latência ≤ 50 ms), embora com margens distintas entre si, já o *Faster R-CNN* não atendeu a todos os critérios.

O *YOLOv8* figurou como o modelo mais rápido do conjunto: 1,5 vezes mais veloz que o *DETR-Custom*, 2,9 vezes mais que o *DETR* Original e 10,1 vezes mais que o *Faster R-CNN*. Já o *DETR-Custom*, com latência de 9,54 ms, apresenta-se como alternativa para aplicações em tempo real, sobretudo em cenários de recursos computacionais limitados, uma restrição recorrente no contexto brasileiro de implantação em dispositivos móveis ou embarcados.

A diferença de 2,0 vezes entre o *DETR-Custom* e o *DETR* Original demonstra o ganho de performance obtidos pelas modificações de arquitetura propostas como a redução de seis para uma camada de *encoder* e *decoder* e diminuição da dimensão oculta de 512 para 256. Embora o *DETR* Original processe cada imagem em 19,09 ms, valor ainda compatível com o critério de 50 ms, o *DETR-Custom* executa a mesma inferência em 9,54 ms, preservando margem operacional considerável para *hardware* de menor potência. O *Faster R-CNN*, em contraste, registrou a maior latência entre os modelos testados 65,38 ms, excedendo o limite estipulado.

Esse ganho de velocidade do *DETR-Custom* em relação ao *DETR* Original confirma a hipótese de que as customizações propostas para a arquitetura reduzem o custo computacional da arquitetura *DETR* sem prejuízo da acurácia, ao menos nas condições do teste de detecção em ambiente controlado.

4.5.4 Análise Estatística

A Tabela 15 compila os resultados do teste de *Wilcoxon* pareado aplicado às 20 classes do *dataset*, comparando os quatro modelos par a par com base na métrica *mAP@0.5:0.95* por classe. O teste foi conduzido sobre amostras pareadas (n = 20), e os p-valores foram corrigidos pelo método de Holm-Bonferroni para múltiplas comparações. Adotou-se o nível de significância $\alpha = 0,05$.

Tabela 15: Resultados do teste de *Wilcoxon* pareado entre os quatro modelos, com base na métrica *mAP@0.5:0.95* por classe (n = 20 classes), p-valores corrigidos por Holm-Bonferroni (α = 0,05).

Comparação (A vs B)	Média A	Média B	Diferença	Wins A	Wins B	W	p-valor (Holm)	Significativo
YOLOv8 vs *Faster R-CNN*	0,7586	0,6800	+0,0787	18	2	31,0	0,0127	Sim
YOLOv8 vs *DETR* Original	0,7586	0,8489	-0,0902	6	14	50,0	0,0400	Sim
YOLOv8 vs *DETR-Custom*	0,7586	0,6008	+0,1578	18	2	19,0	0,0023	Sim
Faster R-CNN vs *DETR* Original	0,6800	0,8489	-0,1689	0	20	0,0	0,00001	Sim
Faster R-CNN vs *DETR-Custom*	0,6800	0,6008	+0,0792	15	5	38,0	0,0214	Sim
DETR Original vs *DETR-Custom*	0,8489	0,6008	+0,2480	20	0	0,0	0,00001	Sim

Fonte: Resultados da pesquisa (2025).

Analisando todas as seis comparações pareadas, todas apresentaram diferenças significativas após a correção de Holm-Bonferroni ($p < 0,05$), indicando que cada modelo se diferenciou dos demais, de forma consistente quando avaliado classe a classe. A ordenação obtida após o teste foi: *DETR* Original 0,8489, *YOLOv8* 0,7586, *Faster R-CNN* 0,6800 e *DETR-Custom* 0,6008, sendo essa ordenação suportada por significância estatística em todas as comparações.

O *DETR* Original apresentou o maior valor médio de *AP* por classe 0,8489, superando os demais modelos com significância estatística, com destaque para as comparações entre ele e o *Faster R-CNN* e o *DETR-Custom*, nos quais teve o melhor desempenho em todas as 20 classes (*Wins* = 20/20, p = 0,00001). Esse resultado refletiu o benefício da arquitetura *DETR* completa para a tarefa de localização precisa em *IoUs* mais exigentes, ainda que ao custo de uma latência mais alta 19,09 ms.

O *YOLOv8* performou melhor que o *Faster R-CNN* em 18 das 20 classes (p = 0,0127) e o *DETR-Custom* em 18 das 20 classes (p = 0,0023), porém ficou estatisticamente abaixo do *DETR* Original, performando melhor em apenas 6 das 20 classes (p = 0,0400). Esse resultado evidenciou o equilíbrio característico do modelo entre velocidade e precisão, posicionando-o como uma alternativa em cenários em que o tempo de inferência é fator crítico.

O *Faster R-CNN* performou melhor que o *DETR-Custom* em 15 das 20 classes (p = 0,0214), confirmando que, mesmo com uma latência substancialmente maior, sua qualidade de detecção em *IoUs* mais exigentes foi superior à do modelo proposto.

O *DETR-Custom* apresentou o menor valor médio de *AP* por classe 0,6008, ficando estatisticamente com menor desempenho dos três modelos de referência. Esse dado deve ser interpretado em conjunto com o objetivo central deste trabalho que não foi maximizar a precisão de localização em *IoUs* estritos, mas sim avaliar o impacto da simplificação arquitetural de uma arquitetura *Transformer* (1 camada de *encoder*, 1 de *decoder*, dimensão oculta de 256), no *trade-off* entre acurácia e custo computacional. Quando avaliado pelo critério da hipótese da pesquisa (*mAP@0.5* $\geq$ 0,80), o modelo alcançou 0,9114, atendendo ao requisito mínimo estabelecido, e demonstrou desempenho competitivo em velocidade frente às arquiteturas *Transformer* e *CNN* de maior complexidade, conforme detalhado na seção 4.5.5.

Em síntese, a análise estatística de *Wilcoxon* confirmou que as diferenças observadas entre os quatro modelos não foram resultado de variabilidade aleatória, mas sim de diferenças reais de capacidade de detecção em *IoUs* mais complexas.

4.5.5 Análise do Trade-off Velocidade versus Acurácia

A Tabela 16 sintetiza o *trade-off* entre velocidade e acurácia, utilizando a razão entre *mAP@0.5* e tempo de inferência como indicador de eficiência global.

Tabela 16: Eficiência computacional dos modelos avaliados.

Modelo	*mAP@0.5*	Inf. (ms)	Eficiência (*mAP*/ms)
YOLOv8	0,9205	6,50	0,1415
DETR-Custom	0,9114	9,54	0,0956
DETR Original	0,9980	19,09	0,0523
Faster R-CNN	0,9800	65,38	0,0150

Fonte: Resultados da pesquisa (2025).

O *YOLOv8* apresentou eficiência integral, sendo 1,5 vezes mais eficiente que o *DETR-Custom*, 2,7 vezes mais eficiente que o *DETR* Original e 9,4 vezes mais eficiente que o *Faster R-CNN*. O *DETR-Custom* foi 1,8 vezes mais eficiente que o *DETR* Original e 6,4 vezes mais eficiente que o *Faster R-CNN*, demonstrando que a customização de arquitetura proposta neste trabalho alcançou o objetivo de aproximar uma arquitetura *Transformer* da faixa de eficiência dos *detector*es baseados em *CNN*.

Os resultados obtidos com o *DETR-Custom* em velocidade e sua eficiência computacional, demonstra um resultado significativo, pois ele, o modelo proposto evidenciou que, para domínios especializados com vocabulário controlado, a simplificação arquitetural pode acelerar significativamente a inferência de uma arquitetura *Transformer*, aproximando-a da velocidade dos *detector*es *CNN*, ainda que ao custo de uma menor precisão de localização (*mAP@0.5:0.95* = 0,6008) quando comparada ao *DETR* Original (0,8489).

Essa constatação reforçou o norte central deste trabalho que a customização de uma arquitetura *Transformer* para um domínio específico e restrito, como o reconhecimento de um vocabulário controlado de sinais em Libras, permite obter desempenho competitivo em velocidade frente a arquiteturas *Transformer* genéricas de maior complexidade em ambiente controlado, atendendo aos requisitos mínimos de acurácia (*mAP@0.5* ≥ 0,80) e velocidade (≥ 15 *FPS*) definidos na hipótese de pesquisa.

4.6 Análise de interpretabilidade por *cross-attention*

4.6.1 Configuração experimental

A capacidade de interpretação do modelo de aprendizado profunda tem papel fundamental na validação de sistemas de classificação de sinais da Libras.

Nesta pesquisa o mecanismo de *cross-attention* foi utilizado para investigar a correspondência entre as regiões identificadas pelo mapa de calor e as regiões anatomicamente relevantes para cada gesto manual avaliado. O experimento foi organizado em três cenários complementares:

(i) um conjunto controlado de 1.199 imagens com fundo verde padronizado onde existem imagens do autor e de dois sintéticos criados artificialmente;

(ii) um conjunto de 1.192 imagens com fundos coloridos variados, que simulam condições reais de uso, denominado fundo ruidoso onde existem imagens do autor e de dois sintéticos criados artificialmente;

(iii) um conjunto de 978 imagens com fundos coloridos variados, que simulam condições reais de uso, denominado fundo ruidoso onde existem imagens somente de um sintético criados artificialmente, porém essa não foi nunca utilizada no treinamento do modelo.

Os cenários (i) e (ii) possuem 20 classes de sinais, ao passo que o cenário (iii) reúne 19 classes. Todos foram avaliados sob quatro threshold de confiança: 0,30, 0,50, 0,70 e 0,90.

A comparação entre os três cenários permite observar o efeito de diferentes fontes de variação visual sobre o comportamento do modelo treinado, distinguindo o que pode indicar ser atribuível à capacidade discriminativa sobre gestos daquilo que procede do contexto visual e das características próprias de imagens sintéticas. O fundo verde funciona como controle do modelo, já o fundo ruidoso simula o desafio do modelo em detecção em fundo não controlado, e o terceiro sintético testa a capacidade de generalização do modelo para representações virtuais de sinais criados artificialmente e com fundos ruidosos.

4.6.2 Fundo Verde

A Tabela 17 apresenta os resultados quantitativos no cenário de fundo com fundo verde controlado. O grau de confiança 0,30 configura o cenário de máxima cobertura, pois esse *threshold* expressou o valor mínimo de probabilidade que o modelo exige de si mesmo antes de emitir uma predição, dessa forma todas as 1.199 imagens receberam uma predição, com taxa de detecção de 100%, ausência total de não detectados (0) e 57 erros de classificação. À medida que o *threshold* foi aumentado, cresceram as não detecções, com 8 em 0,50 até 94 em 0,90, e consequentemente se reduziu os erros de classificação de 54 para 36, o que demonstrou o *trade-off* entre cobertura e precisão.

A redução dos erros de classificação com a elevação do *threshold* não pode ser considerada como um aprimoramento do modelo treinado, mas sim a supressão das predições mais incertas aceitas em *thresholds* menores, já os erros remanescentes em *thresholds* elevados correspondem aos casos de maior confiança equivocada.

Os erros de classificação são as predições realizadas que não coincidiram com a classe esperada, já os não detectados são iguais as imagens nas quais a confiança máxima obtida ficou

abaixo do *threshold* e a taxa de erro em porcentagem foi calculado como a proporção de não detecções sobre o total.

Tabela 17: Taxa de detecção e não detecção por *threshold* – fundo verde.

Threshold	Detectados	% Detectados	Não Detectados	Erros de Classificação	(%) Não Detectados
0,30	1.199	100,00%	0	57	0,00%
0,50	1.191	99,33%	8	54	0,67%
0,70	1.172	97,75%	27	47	2,25%
0,90	1.105	92,16%	94	36	7,84%

Fonte: Resultados da pesquisa (2025).

A Tabela 18 apresenta a confiança média e o desvio padrão calculado para as predições corretas e incorretas em cada *threshold*, incluindo a variação da confiança (calculada como a diferença absoluta entre a média de acertos e a média de erros) calculada sobre imagens detectadas). O delta de confiança de 0,104 (*threshold* 0,30) para 0,018 (*threshold* 0,90) indicou que os erros estão diminuindo, mas persistentes em *threshold* elevados, o que pode indicar que são erros de natureza estrutural, ou seja o modelo aprendeu um padrão que para ele significa "X", mas na realidade pode ser "Y", nesse caso utilizar *threshold* maiores pode fazer com que o modelo só detecte os erros com alta confiança.

Tabela 18: Confiança média (± desvio padrão) nas predições corretas e incorretas — fundo verde.

Threshold	Acertos (média ± dp)	Erros (média ± dp)	Δ Confiança
0,30	0,970 ± 0,065	0,866 ± 0,164	0,104
0,50	0,972 ± 0,056	0,891 ± 0,129	0,081
0,70	0,976 ± 0,041	0,932 ± 0,075	0,044
0,90	0,984 ± 0,018	0,967 ± 0,024	0,018

Fonte: Resultados da pesquisa (2025).

A Tabela 19 demonstra as principais confusões que ocorreram no cenário com fundo verde (chroma key), sendo que os mesmos pares de confusão entre sinais persistiram em todos os *threshold*s, o que indica a natureza estrutural dos erros. Os pares voltar→avançar (22 ocorrências) e avançar→voltar (21 ocorrências) totalizaram 43 erros, ou 75,4% dos 57 erros de classificação no *threshold* 0,30, isso correu devido à impossibilidade de distinguir a orientação do movimento em imagens estáticas, pois ambos os movimentos possuem a mesma configuração e a mesma quantidade de dedos, mudando somente a direção no qual apontam.

Tabela 19: Principais confusões no cenário de fundo verde.

Classe real	Predito como	Ocorrências	Observação
voltar	avancar	22	Ambiguidade gestual direcional
avancar	voltar	21	Ambiguidade gestual direcional
quatro	três	7	Número de dedos similar
seis	roubar	2	Gestos parecidos

Fonte: Resultados da pesquisa (2025).

4.6.3 Fundo Ruidoso

O experimento foi refeito com a mesma quantidade de imagens 1.192, porém nessas imagens foram colocados fundos coloridos variados para simular ambientes reais não controlados.

A Tabela 20 demostra o comportamento do modelo em diferentes *thresholds* em fundos ruidoso, com o *threshold* 0,30, 97,73% das amostras receberam uma classificação, o que fez com esse *threshold* tivesses a menor taxa de não detecção em 2,27%, porem essa amplitude de detecção gerou 303 erros de classificação. À medida que o *threshold* foi se tornando mais restritivo, a relação se inverteu, no *threshold* 0,90, a porcentagem de detecção alcança 57,89%, com 502 imagens não detectadas por não alcançarem a *threshold* mínimo de confiança exigido, porem os erros de classificação foram reduzidos para 61 ocorrências. Os dados demostram o *trade-off* clássico entre cobertura e acurácia aparente, onde o aumento do *threshold* não aprimora o aprendizado do modelo, mas sim, silencia as predições duvidosas, que passam a integrar a categoria "não detectado".

Tabela 20: Taxa de detecção e não detecção por *threshold* – fundo ruidoso.

Threshold	Detectados	(%) Detectados	Não Detectados	Erros de Classificação	(%) Não Detectados
0,30	1.165	97,73%	27	303	2,27%
0,50	1.067	89,51%	125	233	10,49%
0,70	912	76,51%	280	138	23,49%
0,90	690	57,89%	502	61	42,11%

Fonte: Resultados da pesquisa (2025).

A Tabela 21 demonstra a confiança média e o desvio padrão calculado para as predições corretas e incorretas em cada *threshold*, a confiança média dos acertos demonstrou crescimento conforme foi se elevando o *threshold*, saindo de 0,909 em 0,30 para 0,979 em 0,90,

acompanhado de uma diminuição do desvio padrão a cada aumento do *threshold*. A confiança média dos erros também apresentou crescimento à medida que o *threshold* se elevou, saindo de 0,679 em 0,30 para 0,955 em 0,90, o que indicou que *thresholds* mais altos não eliminaram as predições incorretas, apenas filtraram aquelas com menor convicção, deixando passar erros cada vez mais confiantes. A variação na confiança entre acertos e erros alcançou seu valor máximo de 0,230 no *threshold* 0,30, ponto no qual o modelo demonstrou a maior separabilidade entre predições corretas e incorretas, e diminuiu progressivamente para 0,166 em 0,50, 0,078 em 0,70 e apenas 0,023 em 0,90, evidenciando que a separabilidade entre acertos e erros se reduziu conforme o *threshold* foi elevado.

O comportamento observado demonstrou que os erros remanescentes sob *thresholds* altos ainda possuem a natureza estrutural, ou seja, algumas das predições realizadas com maior grau de certeza estavam baseadas em associações visuais equivocadas assimiladas durante o treinamento. No *threshold* 0,90, a diferença de apenas 0,023 entre a confiança média de acertos e erros indicou que, a confiança da predição deixou de ser um indicador útil para distinguir corretas de incorretas, pois os erros remanescentes alcançaram níveis de convicção quase idênticos aos dos acertos.

Tabela 21: Confiança média (± desvio padrão) nas predições corretas e incorretas — fundo ruidoso.

Threshold	Acertos (média ± dp)	Erros (média ± dp)	Δ Confiança
0,30	0,909 ± 0,140	0,679 ± 0,203	0,230
0,50	0,925 ± 0,111	0,759 ± 0,156	0,166
0,70	0,950 ± 0,068	0,872 ± 0,090	0,078
0,90	0,979 ± 0,024	0,955 ± 0,031	0,023

Fonte: Autor (2025).

A Tabela 22 exibiu a ambiguidade gestual que se concentrou na maior parcela das falhas, sendo a maior ocorrência encontrada no sinal "voltar" sendo classificado como "avançar", isso se deveu a ser a mesma CM somente mudando a sua direção de sinalização, após o sinal "homem" classificado como o sinal "ver", ambos os sinais compartilham o formato similar a letra "v", porem dependendo do ângulo da sinalização o modelo pode não ter captado a diferença da quantidade de dedos.

A recorrência de confusões entre numerais como "um" sendo detectado como "dois" e "quatro" sendo detectado como "três", evidencia que o modelo em algumas detecções não

conseguiu dissociar signos que compartilham quase a mesma quantidade de dedos ou a morfologia básica da mão.

A persistência desses mesmos pares de erro ao longo dos diferentes *threshold*s testados pode indicar limitações intrínsecas ao reconhecimento de padrões estáticos, e não ruído associado à baixa confiança.

Tabela 22: Principais confusões no cenário de fundo ruidoso.

Par (classe real → predita)	*Threshold* 0,30	*Threshold* 0,50	*Threshold* 0,70	*Threshold* 0,90
voltar → avancar	32	30	23	17
homem → ver	26	24	14	5
um → dois	18	13	8	2
neutro → dois	17	11	6	1
oito → dois	14	12	12	5
quatro → três	12	12	6	4

Fonte: Resultados da pesquisa (2025).

4.6.4 Fundo Ruidoso mais novo sintético

O terceiro cenário avaliado compreende 978 imagens geradas por *avatar* sintético, com total de 20 classes de sinais. Esse cenário representou o teste de generalização mais exigente, pois o modelo foi submetido a representação sintética nunca vista antes contendo iluminação e proporções corporais distintas das observadas durante o treinamento. Os resultados da Tabela 23 revelam degradação severa em todos os indicadores, situando este cenário em patamar de desempenho substancialmente inferior aos dois anteriores.

A Tabela 23 apresenta dois fenômenos, a taxa de não-detecção e a taxa de FP, com o *threshold* 0,30, o sistema detectou 930 das 978 amostras, mas 632 dessas detecções foram FP, ou seja, o modelo emitiu uma predição de classe com confiança acima do *threshold*, porém a classe real era outra, a taxa de 68,0% auferida de FP entre os itens detectados é crítica, pois representou uma resposta afirmativa e errada, não uma abstenção a classificação do sinal.

Conforme foi analisado outros *thresholds* como o de 0,90, a proporção de FP caiu para 51,6%, porem o sistema passou a não detectar um total de 656 *frames* de sinais.

Esse comportamento observado ilustrou o *trade-off* entre detecção e confiabilidade, em *threshold*s baixos como já discutidos temos a maior quantidade de detecção ao custo de introduzir um volume elevado de FP, enquanto *threshold*s altos reduzem esses erros, mas tornam o sistema de certa forma omisso.

Tabela 23: Cobertura e falsos positivos por *threshold.*

Threshold	Detectados (%)	Detectados	Não detectados	Falsos positivos (%)	Falsos positivos
0,30	95,1%	930	48	68,0%	632
0,50	77,9%	762	216	65,4%	498
0,70	56,9%	556	422	60,6%	337
0,90	32,9%	322	656	51,6%	166

Fonte: Resultados da pesquisa (2025).

A Tabela 24 apresenta que quando se analisa a confiança associada aos FP, não é apenas o qual o modelo erra a classe, mas ao erro associado a alta convicção. No *threshold* 0,30, os FP apresentaram confiança média de 0,7087, valor inferior ao dos acertos 0,8187, e uma variação de confiança de 0,11 que indicou a existência da separabilidade entre as duas distribuições, sendo uma possível solução o uso de um *threshold* mais elevado. No entanto, o aumento no *threshold* para 0,90, causou quase o total desaparecimento dessa separabilidade, pois os FP apresentaram confiança média de 0,9661, apenas 0,0087 abaixo dos acertos 0,9748, o que indicou que utilizando-se alto *threshold*, o modelo produziu FP que são indistinguíveis dos acertos, dessa forma a gradação de confiança deixa de ser um indicador confiável de correção.

Tabela 24: Confiança média de acertos versus falsos positivos (média ± dp).

Threshold	Acertos (média ± dp)	Falsos positivos (média ± dp)	Δ Confiança
0,30	0,8187 ± 0,2011	0,7087 ± 0,2074	+0,1100
0,50	0,8701 ± 0,1487	0,7861 ± 0,1601	+0,0841
0,70	0,9264 ± 0,0849	0,8791 ± 0,0974	+0,0473
0,90	0,9748 ± 0,0278	0,9661 ± 0,0287	+0,0087

Fonte: Resultados da pesquisa (2025).

A Tabela 25 demonstra quais pares de classe são sistematicamente confundidos e em que grau essa confusão persiste ao longo dos *threshold*s. O par de identificação quatro identificado como avancar é o falso positivo mais resistente do sistema, no *threshold* 0,30 ocorreu 31 vezes e, mesmo no *threshold* 0,90 ocorreu 21 vezes, ambos os sinais possuem configuração semelhante, sendo diferido somente pela adição do dedo polegar. O mesmo padrão foi identificado para o gesto voltar sendo identificado como avancar, no *threshold* 0,30 ocorreu 28 vezes e, mesmo no *threshold* 0,90 ocorreu 19 vezes, ambos os sinais possuem a mesma CM mudando somente a direção, e outro caso de maior atenção foi encontrado em a predição do numeral sete que foi predito como dois, no *threshold* 0,30 ocorreu 27 vezes e, mesmo no *threshold* 0,90 ocorreu 12 vezes, ambos os gestos possuem a mesma CM, somente mudando a direção, podendo ocorrer a confusão durante o *data augmentation* quando faz a rotação dos gestos durante o treino, sugerindo que existiu uma sobreposição estrutural no espaço de features entre esses pares.

Em contrates as classes que apresentaram maior quantidade de FP, pares como a classe homem que no começo foi detectado como dois, no *threshold* 0,30 ocorreu 12 vezes e, mesmo no *threshold* 0,90 ocorreu 0 vezes, similar temos a classe não que foi predita como oito, que no *threshold* 0,30 ocorreu 15 vezes e, mesmo no *threshold* 0,90 ocorreu 1 vezes, indicando que esses FP ocorrem apenas em regiões de baixa confiança e podem ser controlados por filtragem.

Tabela 25: Pares de falsos positivos mais frequentes (classe real → classe predita).

Par (real → predito)	*Threshold* 0,30	*Threshold* 0,50	*Threshold* 0,70	*Threshold* 0,90
quatro → avancar	31	30	26	21
voltar → avancar	28	27	23	19
nove → dois	28	19	10	5
sete → dois	27	23	18	12
sim → ver	24	21	17	9
três → avancar	24	18	14	7
um → dois	23	22	15	9
eu → dois	22	20	17	9
seis → avancar	20	16	12	5
zero → dois	18	13	11	5
não → oito	15	15	10	1
zero → avancar	15	11	6	5
três → dois	16	14	9	2
voltar → dois	14	12	9	4
roubar → dois	13	12	9	3
quatro → dois	14	13	5	1
homem → dois	12	9	3	0
eu → avancar	10	9	6	5
nove → avancar	8	7	6	3
homem → avancar	7	6	6	4

Fonte: Resultados da pesquisa (2025).

4.6.5 Mapas Cross-attention

Os mapas de *cross-attention* produzidos pelo *decoder* do modelo *DETR-Custom* treinado foram agrupados em quatro padrões observados e definidos pelo autor, a partir da observação dos mapas de ativação, se criou quatro divisões o *unimodal* (um único *blob* de alta intensidade bem delimitado espacialmente), *bimodal* (apresenta dois centros de atenção distintos e separados espacialmente, *semantic*amente coerente com gestos), irregular (mapas cujo *blob* não é simétrico ou circular, mas se estende em uma direção específica ou se fragmenta em lobos de intensidades diferentes) e difuso extenso (corresponde a manchas de calor espalhadas por grande área da imagem sem um acumulo de intensidade claramente dominante), a categorização teve como fundamentos a percepção de qualidade de localização de mapas de ativação apresentada no artigo *Grad-CAM*: Visual Explanations from Deep Networks via *Gradient*-based *Localization*, onde os mapas se concentram sobre a região de identificação do objeto, dessa forma revelando maior coerência entre a decisão do modelo e a característica visual alvo, ao passo que os mapas dispersos apontam menor especificidade na representação aprendida.

Considerando a inviabilidade de se examinar a totalidade do conjunto de imagens empregadas na etapa de validação, procedeu-se à seleção de um subconjunto para a subsequente análise dos mapas, um para as imagens de fundo controlado, um para as imagens de fundos ruidoso e um para as imagens de fundo ruidoso com sintético nunca visto pelo modelo.

4.6.5.1 Mapas Cross-attention – Fundo Verde

O conjunto de 20 classes analisados revelou um modelo com desempenho geral elevado, atingindo acurácia média de 97,4% sobre um conjunto selecionado com fundo controlado. As confianças individuais variaram de 89,5%, registrado para a classe voltar até 100,0%, observado na classe neutro. Analisando-se os *frames* selecionados, eles se distribuíram em quatro padrões distintos, no *unimodal* foram categorizadas classes como ver, um, sete e cinco, no *bimodal* foram categorizadas classes como casa e oito, no irregular forma categorizadas as classes zero e não e como difuso extenso, predominante estão as classes voltar, avancar, quatro e nove. A qualidade dos mapas de calor gerados apresentou correlação direta com a confiança, as três classes classificados com menor são voltar (89,5%), avancar (94,4%), quatro (94,1%),

concentraram as menores confianças do *dataset*, enquanto as demais classes sustentaram confianças acima de 95%.

A análise dos padrões de atenção demonstra que o modelo aprendeu representações espacialmente coerentes com a fonologia da Libras em grande parte dos casos, o exemplo mais significativo de acerto semântico é a classe casa, cujas manchas bimodais correspondem aos dois pontos anatômicos do gesto, os dois focos de calor representam as mãos direita e esquerda formando o telhado. Da mesma forma, a atenção ao rosto observada na classe ver (mão em V próxima ao olho) e a classe homem (mão no queixo) é linguisticamente válida, pois esses sinais utilizam regiões faciais como articuladores não-manuais. Em contrapartida, os padrões difusos das classes voltar e classe avançar, nas quais as manchas se espalharam desde o rosto, ombros e torso sem delimitar o gesto manual, indicam que o modelo não conseguiu identificar uma feature gestual mais precisa e categórica para essas classes, provavelmente porque ambos dependem da direção do movimento que um único *frame* não foi capaz de capturar adequadamente.

A análise das *queries* do *decoder*, revelou que uma das *queries* ativas identificadas é a Query 13, foi ativada por 18 dos 20 sinais analisados, enquanto o sinal neutro utilizou a Query 9 e o sinal casa utilizou a Query 24. Essa dominância da Query 13 representa um gargalo semântico real, pois os sinais com morfologias gestuais opostas, como avancar e voltar dentre outros competiram pelo mesmo slot de detecção no *decoder*. Por outro lado, a existência das Query 9 e Query 24 como slots especializados foi um sinal positivo, pois o modelo reservou representações dedicadas para classe neutro (classe sem gesto manual definido) e classe casa (gesto bimanuais complexo), indicando que o *decoder* aprendeu, ao menos parcialmente, a diferenciar categorias de alto afastamento semântico. Ainda assim, a questão crítica permaneceu em aberto é como o modelo diferencia os 18 sinais que passaram pela Query 13, sendo o mais provável é que diferenciação ocorreu na etapa de classificação pós-detecção e talvez não no mecanismo de atenção em si, sendo necessário outros métodos de investigação futuro, ou mesmo retreinar o modelo com maior quantidade de objects query para forçar o modelo a se especializar mais por tipo de gesto, posição corporal, ou número de mãos.

Do ponto de vista da interpretabilidade, um dos efeitos mais notados foi a eliminação consistente do fundo verde (chroma key) em todos os 20 exemplos de classe, onde a região de *background* recebeu atenção próxima de zero em todos os mapas, demonstrando que o modelo aprendeu a ignorar o artefato controlado do estúdio e focar no sinalizante. Isso é essencial para qualquer perspectiva de generalização em ambientes não controlados.

As imagens analisadas seguem APÊNDICE D – IMAGENS ANALISADAS PARA MAPAS *CROSS-ATTENTION*.

4.6.5.2 Mapas Cross-attention – Fundo Ruidoso

O conjunto de 20 classes de fundo ruidoso mostrou uma queda da acurácia média de 97,7% para 92,8%, das imagens analisadas, o que correspondeu a uma redução de 4,9% em relação ao *baseline* de fundo verde. As confianças individuais medidas, estavam em um intervalo entre 49,3%, registrado na classe voltar e 99,9%, registrado nas classes neutro, ver e casa, resultando em uma amplitude interna de 50,6% frente aos 10,3% no *baseline* de fundo verde. A disparidade demonstrou que o fundo ruidoso não degradou de maneira uniforme entre todas as imagens analisadas, dessa forma ficou exposto quais classes são frágeis e quais se mantiveram estáveis. Um segundo fenômeno identificado foi que dez classes apresentaram aumento de confiança no fundo ruidoso em comparação ao verde, dentre elas a classe nove com um aumento de +1,9 pp, a classe avancar com um aumento de +4,6 pp e a classe quatro com um aumento de +2,7 pp, o que sugere que, para algumas classes, o fundo verde homogêneo funcionava como âncora, fenômeno que foi desfeito pela utilização do fundo ruidoso.

A análise dos padrões de atenção revelou que a supressão do fundo, que atingia 100% dos casos com a utilização da chroma key, falhou com a utilização do fundo ruidoso, uma vez que todos os 20 mapas de atenção exibiram algum tipo de ativação em algumas regiões do cenário, o que sugere que a supressão verificada no fundo verde decorreu de uma resposta ao estímulo homogêneo.

O padrão *unimodal* se preservou apenas nos sinais cujas características gestuais foram mais discriminativas como as classes ver, roubar, cinco e neutro, ao passo que se notou um aumento de mapas difuso extenso ou irregular no restante dos *frames*. Um padrão novo surgiu no fundo ruidoso, que antes era inexistente no fundo verde, notou-se que na classe zero o fundo fundos compete de forma ativa com o gesto, onde elementos visuais do cenário receberam intensidade de ativação similar à da região da mão. A classe casa sofreu uma mudança, a *bimodal*idade que foi observada no fundo verde com dois focos representando as duas mãos sofreu degradação e gerando um padrão quase *unimodal* com apenas um foco dominante, o que

indicou que a precisão *bimodal* no fundo verde dependia também do sinalizante específico, e não apenas da estrutura do gesto.

A análise das *queries* demonstra que a estrutura de detecção permaneceu idêntica ao do fundo verde: Query 9 para classe neutro, Query 24 para classe casa e Query 13 para os demais 18 sinais. A Query 9 mostrou-se a mais estável com a classe neutro mantendo 99,9% de confiança independentemente do ambiente de fundo, o que sugere que *queries* especializadas para classes *semantic*amente distintas conferem maior estabilidade ao *detector*. A Query 24 preservou a confiança da classe casa, mas existiu a degradação do mapa. O comportamento antes identificado da Query 13 no fundo verde, piorou no fundo ruidoso, pois a amplitude de confiança entre os 18 sinais que compartilham essa query que era de 10,5% passou para 50,6%, isso sugere que a Query 13 generalista não apenas concentrou semânticas distintas em um único *detector*, como passou a exibir comportamento instável diante da combinação de variações de fundo e de sinalizante.

Do ponto de vista da interpretabilidade, notou-se a identificação de sinais robustos como a classe ver, roubar, cinco, neutro, casa e três, pois todos apresentaram confiança acima de 95% em ambos os ambientes fundo verde e fundo ruidoso, o que indica que as características gestuais foram discriminativas o suficiente para sobrepor-se ao ruído visual de qualquer fundo sintético analisado, dessa forma esses gestos representaram o conjunto mais confiável para uma tentativa de teste do modelo em ambiente não controlado.

As imagens analisadas seguem APÊNDICE D – IMAGENS ANALISADAS PARA MAPAS *CROSS-ATTENTION*.

4.6.5.3 Mapas Cross-attention – Fundo Ruidoso juntamente a um novo sintético

A análise do terceiro teste revelou morfologias distintas das observadas nos dois experimentos anteriores, o que é um reflexo da combinação do novo sinalizador sintético que não foi utilizado no treino aliado com fundos ruidosos variados.

O padrão *unimodal* foi preservado nas classes ver, com um *blob* oval sobre a mão em V próxima ao rosto, na classe roubar com um *blob* retangular bem delimitado sobre as duas mãos, a classe homem apresentou um *blob* triangular cobrindo mão e queixo e a classe sim apresentou

um *blob* circular sobre o punho cerrado levantado. O padrão irregular foi observado na classe eu, onde o mapa de atenção cobriu uma coluna contínua que foi do torso à mão apontando para si, a classe avançar mostrou o *blob* se estendendo verticalmente pelo peito sem foco preciso na mão, na classe seis o *blob* estava deslocado para baixo sem correspondência clara com o gesto, a classe sete o *blob* estava com extensão lateral irregular e com uma parte da ativação no fundo noturno com luzes, já a classe oito, o *blob* em forma de gota invertida estava com o centro deslocado do punho, perdendo a *bimodal*idade horizontal observada nos experimento anteriores, nos conjuntos anteriores para este mesmo sinal, a classe nove apresentou um *blob* grande cobrindo mão e parte do peito e a classe cinco o *blob* cobriu a parte principal do gesto, cobrindo também parte do ombro do sintético.

Em nenhum dos 17 casos o fundo foi efetivamente suprimido, com ativações visíveis em elementos como pás eólicas, redes de pesca, luzes noturnas e estruturas industriais.

A análise das *queries* revelou que a estrutura se manteve idêntica à dos dois conjuntos anteriores, a Query 24 exclusivamente utilizada na identificação da classe casa e Query 13 para todos os demais 16 sinais. O comportamento da Query 13 com um terceiro sinalizante indica que existe um gargalo arquitetural, não relacionado a sinalizante específico, mas a uma característica estrutural do modelo. A amplitude de confiança entre os sinais que compartilharam a Query 13 neste terceiro experimento é de 39,6%, com menor valor encontrado na classe não com de 60,1% a 99,7% na classe roubar, valor que ocupou uma posição intermediária entre o experimento de fundo verde com 10,3% e o fundo ruidoso com 50,6%. A Query 24 detectou a classe casa com 98,4% de confiança, queda de 1,4% em relação ao fundo verde, sugerindo que o modelo conseguiu desatrelar o gesto do fundo. A classe não obteve 60,1% de confiança sendo que a mesma foi detectada pela Query 13, o *blob unimodal* centrado é morfologicamente coerente com o gesto, mas a confiança da detecção foi extremamente baixa indicando que o modelo encontrou dificuldade para a discriminação desta classe com o novo sinalizante sintético, possivelmente porque a identidade corporal dos sinalizantes podem ter sido aprendidas como parte da característica para este sinal em específico.

Do ponto de vista da interpretabilidade, os resultados do terceiro experimento forneceram evidência da existência de possível viés, pois com a substituição de todos os sinalizantes anteriores por um único sintético nunca visto durante o treino, o modelo registrou sua menor acurácia média e seus casos mais extremos de queda de confiança. Sinais que apresentaram desempenho ótimo nos experimentos um e experimentos dois como a classe seis,

que operava entre 96,6% e 97,1% diminuiu a confiança para 73,3% com o novo sinalizante, sugerindo que a representação do sinal aprendido para essa classe pode estar atrelado às características físicas dos sinalizantes de treino. Em contraste as classes roubar, ver e três mantiveram a confiança acima dos 97% nos três experimentos, sugerindo que o *transfer learning* dessas classes foram suficientes para superar variações de sinalizante, fundo e iluminação simultaneamente. Dessa forma notou-se que análise comparativa dos três experimentos apresentou dois comportamentos distintos, sendo o primeiro relacionado a um agrupamento de quatro classes com alta invariância a sinalizante e fundo, classe roubar, ver, três e casa, o segundo apresentou um agrupamento de classe suscetíveis a uma brusca variação de confiança relacionada ao sinalizante, fundo controlado ou ruidoso, dessa forma confirmando a necessidade de diversificação do *dataset* de treino.

As imagens analisadas seguem APÊNDICE D – IMAGENS ANALISADAS PARA MAPAS *CROSS-ATTENTION*.

4.7 Síntese dos Três Experimentos

A Tabela 26 apresenta a progressão dos resultados ao longo dos três experimentos, permitindo avaliar o comportamento do *DETR-Custom* à medida que a complexidade do problema aumentou.

Tabela 26: Progressão dos resultados do *DETR-Custom.*

Parâmetro	Exp. 1 (3 classes)	Exp. 2 (20 classes)	Exp. 3 (diversificado)
Classes	3	20	20
Sinalizadores	1 (autor)	1 (autor)	1 (autor) + 2 sintéticos
mAP@0.5	0,333	0,9836	0,9114
F1-Score	—	0,9915	0,4702
Precision	—	0,9924	0,3221
Recall	—	—	0,9993
Mean IoU	—	0,8868	0,8056
Train loss final	1,4249	~0,30	0,2947
Val *loss* final	1,5319	~0,40	0,2535
Gap	0,1070	-0,0273	0,0412
FPS	125,0	156,70	104,87
Inf. time (ms)	8,00	6,38	9,54
Épocas	100	1.495	1.000

Fonte: Resultados da pesquisa (2025).

A análise dos três experimentos revelou uma progressão durante os experimentos.

O Experimento 1 apresentou a viabilidade técnica da arquitetura *DETR* para reconhecimento de sinais em Libras, embora com acurácia limitada pelo *dataset* reduzido de 360 imagens e apenas 3 classes.

O Experimento 2 apresentou que, com volume de dados adequado e 20 classes, o *DETR-Custom* foi capaz de atingir *mAP@0.5* de 0,9836, superando os critérios da hipótese, porém considerando os testes feitos em ambiente controlado.

O Experimento 3 confirmou que a introdução de variabilidade visual por meio de perfis sintéticos manteve o desempenho do modelo dentro dos critérios estabelecidos pela hipótese de pesquisa, com *mAP@0.5* de 0,9114, valor inferior ao dos demais modelos avaliados em ambiente controlado, *YOLOv8* 0,9205, *Faster R-CNN* 0,9800 e *DETR Original* 0,9980. Apesar dessa diferença, todos os modelos atenderam ao critério mínimo definido na hipótese (*mAP@0.5* $\geq$ 0,80), confirmando a viabilidade do *DETR-Custom* para o reconhecimento dos 20 sinais de Libras propostos. Sendo dessa forma a hipótese foi confirmada parcialmente.

O desenvolvimento alcançado partindo do experimento 2 para o experimento 3 indicou um *trade-off* no comportamento do modelo, o *Recall* manteve-se em patamar elevado de 0,9993 e o *mAP@0.5* atendeu ao critério da hipótese, porém houve degradação significativa em *Precision* para 0,3221 e *F1-Score* para 0,4702, reflexo do aumento de FP com a maior variabilidade visual introduzida pelos perfis sintéticos, sendo que a adição de perfis sintéticos é uma estratégia viável para ampliar a diversidade do *dataset* sem necessidade de coleta de dados com novos participantes reais, embora o impacto no comportamento do modelo, especialmente no aumento de FP, deva ser considerado e mitigado em trabalhos futuros, possivelmente por meio de ajuste no número de *object queries* ou de estratégias de filtragem pós-detecção.

A consistência da vantagem computacional ao longo dos três experimentos indicou que a eficiência do *DETR-Custom* se mostrou uma propriedade estável da arquitetura, atendendo aos requisitos mínimos de velocidade definidos na hipótese de pesquisa em todos os cenários avaliados, ainda que sem superar todos os *baselines* em termos absolutos quando trabalhado em ambiente controlado.

4.8 Limitações e Discussão

4.8.1 Limitações do dataset

O *dataset* desenvolvido utilizado neste trabalho apresentou algumas limitações que devem ser consideradas para a interpretação dos resultados, pois uma das limitações é, que somente existe o vídeo de um único participante, o próprio autor, que foi transformado em *frames* para desenvolver inicialmente o *dataset*, depois foram gerados mais alguns modelos sintéticos para diminuir o viés de representação que pode restringir a capacidade de generalização do modelo para outros sinalizadores.

Os vídeos que originaram os *frames* foram gravados em ambiente controlado com *chroma-key* e iluminação controlada, o que pode não refletir as condições reais de atendimento, onde variações de iluminação, oclusões parciais e fundos complexos podem ocorrer.

A inserção de fundos ruidosos gerados artificialmente nas imagens sintéticas e na imagem do pesquisador mitiga parcialmente essa limitação, porém não substitui a variabilidade natural de cenários reais.

Essas limitações foram inerentes ao processo de prova de conceito do presente estudo e apontam para a necessidade de, em trabalhos futuros, coletar dados com múltiplos sinalizadores de diferentes biotipos e em ambientes variados.

A construção do *dataset* utilizando imagens sintéticas em sua composição geradas por intermédio de modelos de IA pôde inserir um certo grau de viés, pois o processo que utiliza *Stable Diffusion* com *ControlNet* e *FaceSwap*, estão sujeitos aos vieses nos dados de treinamento desses modelos generativos, que podem favorecer determinados fenótipos, tons de pele ou estilos de representação corporal em detrimento de outros.

Embora os *prompt*s utilizados tenham sido pensados para gerar diversidade étnica e etária, a limitação de somente três perfis sintéticos restringe o nível de variabilidade do *dataset*, podendo gerar um modelo que apresenta desempenho inferior para indivíduos cujas características físicas diferem consideravelmente dos perfis representados.

Esse viés potencial reforçou a necessidade em trabalhos implementações futuras, realizar a coleta de dados com sinalizadores variados, garantindo uma maior representatividade demográfica e diversidade nos estilos de sinalização.

A aplicação da análise de *cross-attention* do *decoder*, descrita na sub-subseção 3.5.5, constitui uma estratégia para investigar se o modelo está, de fato, utilizando as regiões corretas da imagem para suas predições, ou se está se apoiando em artefatos visuais como o *background* da imagem, roupa ou outros elementos visuais de fundo.

4.8.2 Teste de robustez com fundos variados

Com o objetivo de verificar a robustez do modelo em situações de cenários ruidoso, um teste complementar foi conduzido com imagens cujos fundos ruidosos foram gerados segundo o procedimento descrito na sub-subseção 3.3.1 e no Anexo C. As imagens não integraram os conjuntos de treinamento nem de validação, sendo que o modelo foi treinado e validado unicamente com imagens de fundo verde (*chroma-key*).

O teste de robustez consistiu na avaliação dos mesmos *checkpoint*s registrados ao longo do treinamento, por meio do *script* de avaliação independente descrito na subseção 4.4, aplicado ao subconjunto de imagens com fundos ruidosos.

A Tabela 27 apresenta a comparação entre os dois cenários na época 1.000.

Tabela 27: Análise de robustez do modelo *DETR-Custom* sob diferentes condições de fundo.

Métrica	Fundo verde	Fundo ruidoso	Variação	Interpretação
mAP@0.5	0,9114	0,4107	-50,1 p.p.	Queda crítica na precisão geral
mAP@0.5:0.95	0,6008	0,0912	-50,9 p.p.	Queda severa em *IoU* rigoroso
F1-Score	0,4702	0,5081	+3,8 p.p.	Equilíbrio precisão/*recall* afetado
Precision	0,3221	0,5050	+18,3 p.p.	Quase metade dos positivos são falsos
Recall	0,9993	0,5334	-46,6 p.p.	Metade dos objetos não detectados
Mean IoU	0,8056	0,6267	-17,9 p.p.	Menor impacto relativo (sobreposição)
FPS	104,87	626,2	+497,1%	Inferência mais rápida (menos processamento)

Fonte: Resultados da pesquisa (2025).

A queda acentuada nas métricas de detecção evidenciou que o modelo apresentou elevada sensibilidade à variação de fundo, esse resultado era esperado uma vez que o treinamento se restringiu a imagens de fundo verde, e o modelo não dispôs de exemplos que lhe permitissem dissociar o sinal do sinalizador das características visuais do cenário.

A evolução das métricas ao longo das épocas no cenário com fundo ruidoso revelou convergência mais lenta e com patamar inferior ao do cenário controlado. O *mAP@0.5* alcançou seu valor máximo de 0,4353 na época 590, menos da metade do registrado com fundo verde 0,9114. O *mAP@0.5:0.95*, que mensura a precisão das *bounding boxes* sob *threshold*s mais rigorosos, limitou-se a 0,0912, sinalizando que a localização espacial dos sinais sofre degradação expressiva na presença de fundos complexos.

O *recall* de 0,5334 mostrou que o modelo detectou pouco mais da metade dos sinais quando exposto a fundos ruidosos, a *precision* de 0,5050 revela que cerca de metade das detecções correspondem a FP, mantendo o padrão de excesso de detecções já observado no cenário controlado (*Precision* 0,3221), agora combinado com a redução do *Recall*. Esses valores traduzem a dificuldade do modelo em discriminar as regiões de interesse como as mãos e tronco dos elementos visuais concorrentes presentes no fundo.

O *Mean IoU* de 0,6267, embora inferior ao do cenário controlado 0,8056, permanece acima do *threshold* 0,50, o que sugere que, nos casos em que o modelo de fato detecta um sinal em fundo ruidoso, a delimitação espacial da bounding box preserva sobreposição razoável com a anotação *ground truth*.

O aumento do *FPS* de 104,87 no cenário controlado para 626,2 no fundo ruidoso refletiu a redução do volume de predições geradas pelo modelo, uma vez que a etapa de pós-processamento de detecções depende do número de candidatos produzidos. Esse ganho de velocidade, contudo, não representa melhoria de desempenho, sendo consequência da incapacidade do modelo de identificar sinais com confiança suficiente em fundos não controlados.

Essa limitação apontou diretamente para a necessidade de incorporar imagens com fundos variados ao conjunto de treinamento em trabalhos futuros. O *dataset* já inclui imagens com fundos ruidosos que não foram empregadas nesta etapa experimental, e sua integração ao treinamento tende a elevar a robustez do modelo frente a variações de cenário.

Em um contexto real de uso do sistema para teleatendimento de emergência, a padronização do fundo de captura, por exemplo, com superfície lisa ou *chroma-key* no posto de

atendimento, pode configurar uma medida operacional factível, capaz de atenuar essa limitação enquanto o modelo não dispuser de treinamento com maior diversidade de fundos.

4.8.3 Limitações da comparação com baselines

A comparação experimental descrita na subseção 4.5 apresentou uma limitação quanto às condições de *transfer learning*, pois o *YOLOv8* e o *Faster R-CNN* foram inicializados com pesos pré-treinados no *COCO*, um *dataset* voltado à detecção de objetos, ao passo que o *DETR-Custom* e o *DETR* Original utilizaram apenas pesos provenientes do ImageNet, restrito à classificação de imagens, com o módulo *Transformer* treinado integralmente a partir do zero.

Essa diferença implicou que os modelos inicializados com pesos do COCO já possuíam representações internas ajustadas à tarefa de detecção, o que lhes conferiu vantagem na velocidade de convergência.

4.8.4 Recomendações para trabalhos futuros

A partir dos resultados obtidos e das limitações observadas neste estudo, sugeriram-se as seguintes investigações futuras:

Incorporação de mais sintéticos, com a geração de *frames* correspondentes aos 20 sinais executados por *avatar*es de características distintas, tons de pele, idade, ângulos de execução dos gestos, roupas, grau de iluminação. Tal estratégia elimina a exigência de submissão a comitê de ética e favorece a diversificação do *dataset*.

Gravações com múltiplos sinalizadores reais: sugere-se a captação de vídeos com, no mínimo, quinze sinalizadores de biotipos, faixas etárias e estilos de sinalização diversos, em ambientes heterogêneos, a fim de mensurar a capacidade de generalização do modelo sob condições ao uso cotidiano.

Crescimento do vocabulário reconhecido: recomenda-se a inclusão progressiva de classes para além das 20 vinculadas ao contexto de emergência, com monitoramento do comportamento das métricas de desempenho à medida que o repertório lexical se expande.

Incorporação de informação temporal por meio de arquiteturas que processem sequências de *frames* em vez de imagens estáticas, como redes recorrentes acopladas ao *DETR*, de modo a capturar a direção do movimento em sinais que dependem dessa informação para serem distinguidos. Os erros estruturais observados nesta pesquisa, especialmente as confusões persistentes entre os pares voltar e avançar, evidenciam que um único *frame* não é suficiente para desambiguar sinais que compartilham a mesma CM e diferem apenas pela direção do gesto.

Investigação de configurações intermediárias do *DETR-Custom*: indica-se a avaliação de arquiteturas com 2 a 4 camadas de *encoder*/*decoder* e dimensões ocultas intermediárias, de modo a traçar uma curva entre capacidade representacional e custo computacional, assim identificando a configuração mais adequada a cada cenário de aplicação.

Fazer teste de inferência em *hardware*s diferentes ao de referência, sendo necessária validação em dispositivos com recursos mais limitados.

Avaliação com múltiplos sinais simultâneos por *frame*, cenário que pode ocorrer em comunicações mais complexas.

Avaliação com usuários surdos, por meio da condução de uso da solução com membros da comunidade surda em cenário simulado de emergência, condicionados à aprovação prévia de comitê de ética, com o objetivo de verificar tanto a precisão do reconhecimento quanto a usabilidade da interface pelo público-alvo.

4.9 Validação da Hipótese

A hipótese de pesquisa que o *DETR-Custom* alcançaria o reconhecimento de 20 classes de sinais de Libras com *mAP@0.5* $\geq$ 0,80 e *F1-Score* médio $\geq$ 0,80, ao mesmo tempo em que manteria *FPS* $\geq$ 15 e latência $\leq$ 50 ms.

A Tabela 28 consolidou os resultados da validação com base no Experimento 3, que constitui a avaliação mais rigorosa do estudo por incorporar variabilidade entre sinalizadores.

Tabela 28: Validação dos critérios da hipótese — Experimento 3.

Critério	Valor Obtido	Margem	Status
mAP@0.5 ≥ 0,80	0,9114	+13,9%	ATENDE
F1-Score ≥ 0,80	0,4702	-41,2%	NÃO ATENDE
FPS ≥ 15	104,87	7,0×	ATENDE
Latência ≤ 50 ms	9,54 ms	5,2×	ATENDE

Fonte: Resultados da pesquisa (2025).

Dos quatro critérios da hipótese, três foram atendidos no Experimento 3, e a hipótese de pesquisa foi parcialmente confirmada. A análise individual de cada critério revelou os pontos em que a arquitetura customizada se mostrou suficiente e aqueles em que demandou ajustes.

O *mAP@0.5* confirmou-se com margem de 13,9% acima do *threshold* 0,80, atingindo 0,9114, indicando que o modelo identificou corretamente os sinais de Libras na maioria dos casos sob o critério de *IoU* ≥ 0,5 evidenciando que o *DETR-Custom*, mesmo com *encoder* e *decoder* reduzidos a uma única camada e dimensão oculta de 256, manteve capacidade representacional suficiente para discriminar corretamente as 20 classes de sinais.

O *FPS* de 104,87 ultrapassou em 7,0 vezes o requisito mínimo de 15 *frames* por segundo, enquanto a latência de 9,54 ms se situou 5,2 vezes abaixo do limite estipulado de 50 ms, atendendo aos critérios de tempo real necessários para uma aplicação assistiva, sustentando a viabilidade do modelo em aplicações como o atendimento e*merge*ncial visado pelo trabalho, e validaram o objetivo central da customização proposta.

O *F1-Score*, ao contrário, foi refutado com valor de 0,4702 contra o *threshold* 0,80, déficit de 41,2%, resultado proveniente do desequilíbrio entre *Recall* 0,9993 e *Precision* 0,3221, causado pelo elevado número de FP gerados pelo modelo no experimento 3 (2.962 FP contra 1.198 VP). Esse desequilíbrio decorreu da combinação entre o número reduzido de *object queries* da arquitetura customizada, calibrado para favorecer eficiência computacional, e a maior variabilidade visual introduzida pelos perfis sintéticos no Experimento 3. O modelo passou a propor mais detecções do que o necessário diante de cenários visuais menos previsíveis.

A comparação com as arquiteturas de referência (subseção 4.5) revelou que o *DETR-Custom* apresentou desempenho competitivo em velocidade frente às arquiteturas *Transformer* e *CNN* tradicionais, com o fator de 2,0x em relação ao *DETR* Original e 6,9x em relação ao *Faster R-CNN*, mas ficou abaixo do *YOLOv8* 153,74 *FPS* tanto em velocidade quanto em *mAP@0.5* 0,9205. Em precisão de localização sob *IoUs* mais exigentes

(*mAP@0.5:0.95*), o *DETR-Custom* apresentou o menor valor entre os quatro modelos avaliados.

Os resultados apoiaram parcialmente a ideia central do trabalho. Uma versão customizada do *DETR*, com redução de seis para uma camada de *encoder* e *decoder* e dimensão oculta de 256, atingiu desempenho aceitável em *mAP@0.5* e velocidade competitiva, atendendo aos requisitos mínimos de uma aplicação assistiva em tempo real voltada ao atendimento de emergência em ambiente controlado, ainda que com limitações em *Precision* e em precisão de localização sob *IoUs* estritos.

É relevante ressaltar que a validação dos critérios da hipótese foi conduzida em cenário controlado com fundo verde padronizado. Um teste adicional de robustez (sub-subseção 4.8.2) demonstrou que o desempenho do modelo é sensível a variações de fundo, com *mAP@0.5* reduzido para 0,4107 quando avaliado sobre imagens com fundos ruidosos não presentes no treinamento. Essa constatação delimitou o escopo de aplicação dos resultados validados: os critérios atendidos da hipótese aplicam-se a cenários com fundo controlado, como postos de atendimento com fundo padronizado, e a generalização para cenários com fundos variados requer a inclusão dessas condições no treinamento, conforme discutido na sub-subseção 4.8.2.

4.10 Considerações finais do capítulo

Este capítulo apresentou três experimentos progressivos que avaliaram o *DETR-Custom* desde a prova de conceito até a comparação experimental com *baselines* consolidadas, utilizando o mesmo *dataset* com imagens geradas em condições controladas.

O modelo atingiu *mAP@0.5* de 0,9114 no experimento 3. Valor que supera o necessário estipulado na hipótese, ao mesmo tempo, fica abaixo de todos os modelos concorrentes (*DETR* Original 0,9980; *Faster R-CNN* 0,9800; *YOLOv8* 0,9205). Porém quando avaliado a dimensão computacional o *DETR-Custom* entregou 104,87 *FPS* com latência de 9,54 ms, atendendo aos valores estipulados na hipótese ($FPS \geq 15$, latência ≤ 50 ms). Apresentou desempenho 2,0 vezes mais rápido que o *DETR* Original e 6,9 vezes mais veloz que o *Faster R-CNN*, porém nesse quesito não performou melhor que o modelo do *YOLOv8*.

A utilização de perfis sintéticos no experimento 3 demonstrou um *trade-off* nítido. O *recall* se manteve elevadíssimo 0,9993 e o *mAP@0.5* cumpriu o critério da hipótese, mas a *precision* caiu para 0,3221 e o *F1-Score* recuou para 0,4702. Indicando que o aumento de variabilidade visual inflou os *FP*. A estratégia de ampliar o *dataset* com sintéticos é, sim, viável, mas o custo em precisão precisará ser mitigado em trabalhos futuros, sob pena de inviabilizar a aplicação em cenários reais com exigência de baixa taxa de alarme falso.

Dos quatro critérios estabelecidos na hipótese, três foram satisfeitos no experimento 3, *mAP@0.5*, *FPS* e latência. O *F1-Score* ficou abaixo do *threshold*. A análise de *cross-attention* do *decoder*, que detalhada na subseção 4.6 e subseção 4.8.2, confirma que o modelo concentra a atenção nas regiões anatômicas pertinentes ao sinal em ambiente controlado. Ao mesmo tempo, expõe sensibilidade a mudanças de fundo e ao tipo de sinalizador, fragilidade do modelo treinado que os números de precisão já haviam denunciado.

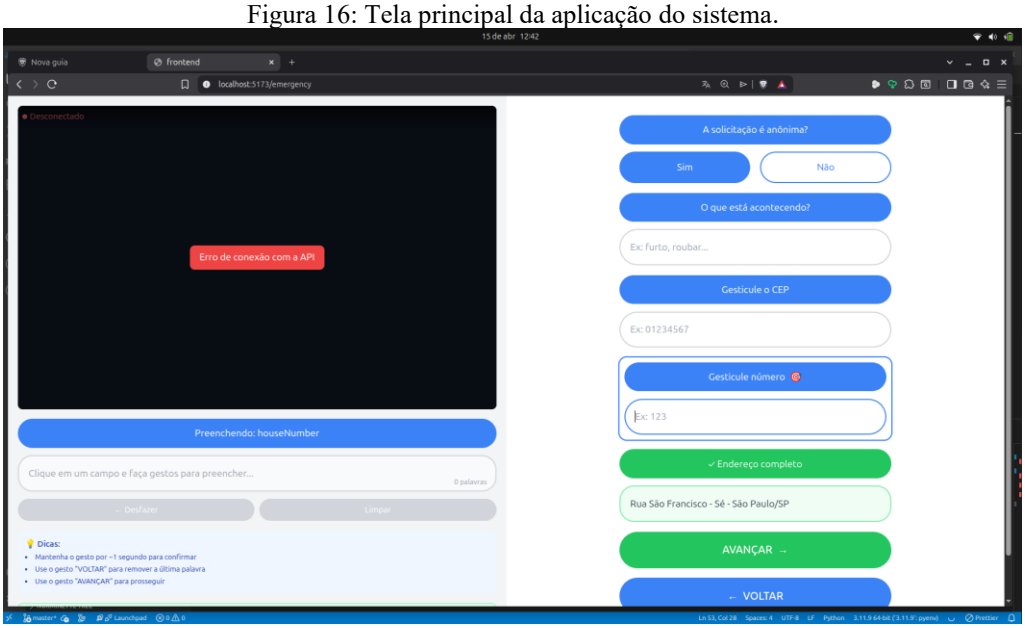
Três limitações merecem ser comunicadas com clareza, o *dataset*, mesmo expandido com sintéticos, ainda varia pouco. Validar o modelo com sinalizadores reais e diversificados é um passo necessário; os *baselines* baseados em *CNN* (*YOLOv8* e *Faster R-CNN*) se ampararam em pesos pré-treinados no *COCO*, ao passo que o *DETR-Custom* partiu de pesos do *ImageNet* com o módulo *Transformer* treinado do zero, dessa forma a comparação direta, possui essa diferença de ponto de partida, e deve ser analisada com essa ressalva metodológica; a sensibilidade do modelo a fundos ruidoso não presentes no treinamento derrubou o *mAP@0.5* para 0,4107, demarcando com precisão o escopo atual do sistema, ambientes de fundo controlado. Por fim, o vocabulário de 20 sinais, pensado para emergências, é restrito pela proposta do trabalho, investigar como a arquitetura se comporta com conjuntos maiores ou com outros domínios de sinais demandará nova rodada de experimentos.

5. FRONTEND DA APLICAÇÃO

O *Frontend* do sistema foi desenvolvido utilizando-se a biblioteca *React*, na versão 19, associada ao *bundler Vite* e ao *framework* de estilização *Tailwind CSS*, o que resultou em uma arquitetura de página única (*Single Page Application*), alinhada a padrões modernos do desenvolvimento web. A navegação entre as telas é gerenciada pela biblioteca *React Router DOM*, com a configuração centralizada no componente *App.jsx*. A troca de dados com o *backend* emprega a biblioteca *Axios*, a captura de vídeo do dispositivo do usuário é gerida pelo pacote *React-webcam*, permitindo a conexão entre a interface e o modelo de reconhecimento de gestos mantido no *backend*.

A estrutura interna do diretório *src/* segue o princípio de separação de responsabilidades, distribuído em três camadas, a primeira camada de páginas (*pages/*), composta por oito telas que correspondem aos estados navegacionais da aplicação, a segunda camada de componentes (*components/*), que reúne elementos reutilizáveis, entre os quais se destaca o *GestureTranscription.jsx*, responsável pela captura contínua de *frames* via webcam e pelo envio dessas imagens à *API* de inferência para reconhecimento de sinais em Libras e a terceira camada de recursos estáticos (*assets/*), reservada a mídias e arquivos de suporte visual. Essa organização em camadas contribui para a manutenibilidade do código e para a delimitação clara entre lógica de apresentação e lógica de negócio, dois princípios centrais na construção de interfaces acessíveis e escaláveis.

Figura 16: Tela principal da aplicação do sistema.



Fonte: Resultados da pesquisa (2025).

6. CONCLUSÃO

A pesquisa desenvolveu e avaliou uma prova de conceito para reconhecimento automático de sinais da Libras voltada ao atendimento e*merge*ncial. A fundamentação do projeto foi adaptar a arquitetura *DETR*, nomeada de *DETR-Custom* para um modelo computacionalmente leve, capaz de operar próximo do tempo real mesmo em *hardware* com recursos limitados.

O trajeto metodológico incluiu a montagem de um *dataset* próprio com 20 sinais, sinalizados pelo autor e anotados manualmente, depois diversificados utilizando perfis sintéticos gerados por modelos generativos. Para preservar a integridade dos subconjuntos de treino, validação e teste e conter o risco de *data leakage*, definiu-se um protocolo específico de divisão. A arquitetura *DETR-Custom* partiu de uma simplificação estrutural do *DETR* original, com menos camadas no *encoder* e no *decoder*, redução da dimensão oculta, de modo a equilibrar eficiência computacional e acurácia dentro do domínio restrito da aplicação de teste.

A avaliação comparativa entre o desempenho determinado na hipótese e o efetivamente obtido pelas customizações propostas revelou uma confirmação parcial. Treinado com o *dataset* diversificado (autor + perfis sintéticos), o *mAP@0.5*, com *threshold* planejado de 0,80 obteve 0,9114, superando o planejado em 13,9%. O *FPS* mínimo delimitado é de 15 *frames* por segundo e obteve-se 104,87, sete vezes acima do planejado. O teto de latência máxima estipulado é de 50 ms por imagem e obteve-se 9,54 ms, 5,2 vezes abaixo do limite. O *F1-Score* possui um *threshold* planejado de 0,80 e obteve 0,4702, déficit de 41,2% em relação ao planejado. Já em comparação aos *baselines*, o *DETR-Custom* apresentou o menor *mAP@0.5* entre os quatro modelos (0,9114 vs. 0,9980 do *DETR* Original, 0,9800 do *Faster R-CNN* e 0,9205 do *YOLOv8*). Em velocidade, superou o *DETR* Original (104,87 vs. 52,39 *FPS*) e o *Faster R-CNN* (104,87 vs. 15,30 *FPS*), mas ficou abaixo do *YOLOv8* (104,87 vs. 153,74 *FPS*). Em relação ao tempo de inferência o *DETR-Custom* apresentou um desempenho superior em relação aos modelos *DETR* Original (9,54 ms vs. 19,09 ms) e o *Faster R-CNN* (9,54 ms vs. 65,38 ms), mas não superou o *YOLOv8* (9,54 ms vs. 6,50 ms).

A análise de interpretabilidade, ancorada nos mapas de *cross-attention* do *decoder*, forneceu indícios qualitativos de que, com fundo controlado, o modelo direciona a atenção para as regiões anatomicamente pertinentes à execução dos sinais. Quando se introduziram fundos ruidosos, a sensibilidade a essa mudança ficou evidente. A constatação reforça a necessidade

de incorporar diversidade de cenários ao conjunto de treinamento para aplicações fora de ambientes controlados.

A hipótese de pesquisa confirmou-se parcialmente, pois o *DETR-Custom* atendeu a três dos quatro critérios quantitativos (*mAP@0.5*, *FPS* e latência), mas o *F1-Score* permaneceu abaixo do patamar definido. Ainda assim, os resultados demonstram que customizar arquiteturas *Transformer* para domínios específicos e vocabulários controlados pode gerar soluções adequadas a dispositivos com poucos recursos computacionais, mesmo que ajustes adicionais sejam necessários para equilibrar *Precision* e *Recall* em cenários de maior variabilidade visual.

O trabalho contribuiu na área tecnológica ao suprir uma lacuna da literatura, levando o *DETR* ao reconhecimento de Libras em contexto prático e propondo uma versão customizada validada experimentalmente. A metodologia de construção de *dataset* com dados sintéticos também representa um caminho replicável para cenários com escassez de amostras. Na área social, a solução configura-se como tecnologia assistiva com potencial para reduzir barreiras comunicacionais em situações de emergência, alinhando-se as ODS relativos à inclusão e ao acesso à justiça.

Algumas limitações precisam ser registradas.

O *dataset* foi construído a partir de um único sinalizador juntamente com mais três perfis sintéticos gerados por modelos generativos, embora a diversificação tenha ampliado a variabilidade, o *dataset* permaneceu restrito a um número reduzido de sinalizadores, biotipos e estilos de sinalização. Em teste com um perfil sintético diferente dos utilizados no treino do modelo, com características físicas, roupas diferentes apresentou desempenho inferior no experimento.

O treinamento e a validação foram conduzidos em fundo verde controlado (*chroma-key*). O teste de robustez com fundos ruidosos (sub-subseção 4.8.2) demonstrou uma queda de desempenho, com *mAP@0.5* caindo de 0,9114 para 0,4107. Os resultados desta pesquisa aplicam-se a cenários com fundo controlado, e a generalização para ambientes mais complexos requer a inclusão dessas condições no treinamento do modelo.

A redução do número de *object queries*, adotada para beneficiar a eficiência computacional, contribuiu para o excesso de FP observado no Experimento 3. O desequilíbrio entre *Recall* (0,9993) e *Precision* (0,3221) evitou que o *F1-Score* estabelecido na hipótese fosse atingido. Ajustes na configuração de *object queries* e estratégias de filtragem pós-detecção são

necessários para que o modelo opere de forma equilibrada em cenários de maior variabilidade visual.

O vocabulário ficou restrito a 20 sinais associados ao contexto de atendimento e*merge*ncial. Embora adequado para uma prova de conceito, esse escopo cobre uma fração pequena do léxico da Libras necessário para comunicação em cenários reais. Adicionalmente, o processamento independente de *frames* dificultou o reconhecimento de sinais que se distinguem apenas pela direção do movimento, como os pares voltar e avançar, conforme observado nos resultados.

A comparação com *YOLOv8* e *Faster R-CNN* foi conduzida sob condição assimétrica de *transfer learning*, pois esses modelos foram inicializados com pesos pré-treinados em COCO, enquanto o *DETR-Custom* e o *DETR* Original utilizaram apenas pesos do ImageNet.

Finalmente, toda a avaliação ocorreu em ambiente simulado, sem validação com usuários surdos em cenários reais de atendimento e*merge*ncial.

Os resultados obtidos e as limitações identificadas indicam frentes específicas de continuidade para esta pesquisa:

a) Expandir e diversificar o *dataset*, incorporando vídeos de múltiplos sinalizadores reais e *avatar*es sintéticos adicionais, com maior variedade de biotipos, ângulos de execução e, sobretudo, fundos complexos e não controlados, para elevar a robustez e a capacidade de generalização do modelo.

b) Ajustar o número de *object queries* e investigar estratégias de filtragem pós-detecção com o objetivo de reduzir FP e recuperar o equilíbrio entre *Precision* e *Recall*, viabilizando o critério de *F1-Score* em cenários de maior variabilidade.

c) Ampliar progressivamente o vocabulário para além do contexto e*merge*ncial básico e examinar como o *DETR-Custom* se comporta diante de um espaço de classes expandido.

d) Investigar arquiteturas intermediárias, variando o número de camadas e as dimensões ocultas, a fim de mapear o *trade-off* entre capacidade representacional e custo computacional.

e) Realizar testes de usabilidade e de precisão com membros da comunidade surda em ambiente simulado de atendimento e*merge*ncial, mediante aprovação de comitê de ética.

f) Explorar técnicas de compressão de modelos e plataformas de *hardware* alternativas para avaliar a viabilidade de execução do sistema em dispositivos móveis de menor capacidade computacional.

Este trabalho demonstrou que um sistema de reconhecimento de sinais de Libras baseado em uma arquitetura *DETR* customizada é tecnicamente viável, o artefato entregue, software, modelo treinado, *API* de comunicação e interface de simulação constitui uma prova de conceito promissora. Os resultados obtidos estabelecem uma base para o desenvolvimento de ferramentas assistivas mais robustas e de maior cobertura lexical, capazes de contribuir para a construção de uma sociedade mais inclusiva.

REFERÊNCIAS

ABDUL AMEER, R. S. *et al.* Empowering Communication: A Deep Learning Framework for Arabic Sign Language Recognition with an Attention Mechanism. **Computers**, v. 13, n. 6, p. 153, 19 jun. 2024.

ALY, Mohammed; FATHI, Islam S. Recognizing American Sign Language gestures efficiently and accurately using a hybrid transformer model. **Scientific Reports**, v. 15, n. 1, p. 20253, 23 jun. 2025.

BARRETO, Ricardo Souza. A Comunicação e suas novas narrativas: desafios e oportunidades no acesso aos serviços emergenciais 190 pelas pessoas com deficiência auditiva ou surdez. **Interfaces da Comunicação**, v. 1, n. 1, p. 1–21, 3 maio 2023.

BENEVIDES, Marcello Pereira *et al.* Sign Talk Assistive Technology: Real-Time Recognition of the Libras Typical Alphabet Using Artificial Intelligence. **Revista de Gestão Social e Ambiental**, v. 18, n. 12, p. e010610, 31 dez. 2024.

BHADOURIA, Aman *et al.* LSTM-Based Recognition of Sign Language. *In*: IC3 2024: 2024 SIXTEENTH INTERNATIONAL CONFERENCE ON CONTEMPORARY COMPUTING. **Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing**. Noida India: ACM, 8 ago. 2024. Disponível em: <https://dl.acm.org/doi/10.1145/3675888.3676105>. Acesso em: 26 nov. 2025.

BHAT, Sheethal *et al.* **Exemplar Med-DETR: Toward Generalized and Robust Lesion Detection in Mammogram Images and beyond**. arXiv, , 30 jul. 2025. Disponível em: <http://arxiv.org/abs/2507.19621>. Acesso em: 27 nov. 2025.

BRASIL. Constituição da República Federativa do Brasil de 1988. 1988.

BRASIL. Lei nº 10.436, de 24 de abril de 2002. . 24 abr. 2002, Sec. 1, p. 23, col. 3.

BRASIL. Decreto nº 6.949, de 25 de agosto de 2009. . 25 ago. 2009, Sec. 1, p. 3, col. 1.

CARION, Nicolas *et al.* End-to-End Object Detection with Transformers. *In*: VEDALDI, Andrea *et al.* (Orgs.). **Computer Vision – ECCV 2020**. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020. v. 12346 p. 213–229.

CARREIRA, Joao; ZISSERMAN, Andrew. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *In*: 2017 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, HI: IEEE, jul. 2017. Disponível em: <http://ieeexplore.ieee.org/document/8099985/>. Acesso em: 26 nov. 2025.

CARVALHO, Dariel De; MANZINI, Eduardo José. Aplicação de um Programa de Ensino de Palavras em Libras Utilizando Tecnologia de Realidade Aumentada. **Revista Brasileira de Educação Especial**, v. 23, n. 2, p. 215–232, jun. 2017.

CHAKRABORTY, Biplab Ketan *et al.* Review of constraints on vision-based gesture recognition for human–computer interaction. **IET Computer Vision**, v. 12, n. 1, p. 3–15, fev. 2018.

CUNHA, Mariane Maria De Carvalho; FRANCO, Raquel Aparecida Soares Reis. Glossário em Libras como instrumento para auxiliar na inclusão de alunos surdos. **Revista de Estudos e Pesquisas sobre Ensino Tecnológico (EDUCITEC)**, v. 7, p. e132521, 26 jan. 2021.

DE AVELLAR SARMENTO, Amanda Hellen; ANTONELLI PONTI, Moacir. A Cross-Dataset Study on the Brazilian Sign Language Translation. *In*: 2023 IEEE/CVF INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (ICCVW). **2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)**. Paris, France: IEEE, 2 out. 2023. Disponível em: <https://ieeexplore.ieee.org/document/10350372/>. Acesso em: 27 nov. 2025.

DE COSTER, Mathieu; HERREWEGHE, Mieke Van; DAMBRE, Joni. Sign Language Recognition with Transformer Networks. Proceedings of the Twelfth Language Resources and Evaluation Conference, p. 6018–6024, maio 2020.

DENG, Jia *et al.* ImageNet: A large-scale hierarchical image database. *In*: 2009 IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS (CVPR WORKSHOPS). **2009 IEEE Conference on Computer Vision and Pattern Recognition**. Miami, FL: IEEE, jun. 2009. Disponível em: <https://ieeexplore.ieee.org/document/5206848/>. Acesso em: 26 nov. 2025.

DEY, Subhadeep *et al.* **Clean Text and Full-Body Transformer: Microsoft's Submission to the WMT22 Shared Task on Sign Language Translation**. arXiv, , 2022. Disponível em: <https://arxiv.org/abs/2210.13326>. Acesso em: 4 dez. 2025.

DIPIETRO, L.; SABATINI, A. M.; DARIO, P. A Survey of Glove-Based Systems and Their Applications. **IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)**, v. 38, n. 4, p. 461–482, jul. 2008.

DOSOVITSKIY, Alexey *et al.* **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. arXiv, , 3 jun. 2021a. Disponível em: <http://arxiv.org/abs/2010.11929>. Acesso em: 26 nov. 2025.

DUBOVA, Marina *et al.* Is Ockham's razor losing its edge? New perspectives on the principle of model parsimony. **Proceedings of the National Academy of Sciences**, v. 122, n. 5, p. e2401230121, 4 fev. 2025.

DYZEL, Vernandi *et al.* Assistive Technology to Promote Communication and Social Interaction for People With Deafblindness: A Systematic Review. **Frontiers in Education**, v. 5, p. 578389, 17 set. 2020.

EL-QORAYCHY, Fatima-Zahrae et al. Explainable AI for sign language recognition models: Integrating Grad-Cam LIME and Integrated Gradients. PLOS One, [s. l.], v. 20, n. 12, e0336481, 10 dez. 2025. DOI: https://doi.org/10.1371/journal.pone.0336481. Disponível em: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0336481.

EROL, Ali *et al.* Vision-based hand pose estimation: A review. **Computer Vision and Image Understanding**, v. 108, n. 1–2, p. 52–73, out. 2007.

EVERINGHAM, Mark *et al.* The Pascal Visual Object Classes (VOC) Challenge. **International Journal of Computer Vision**, v. 88, n. 2, p. 303–338, jun. 2010.

FELIPE, Tanya Amara. **Libras Em Contextos Cursos Basicos - Livro Do Estudante**. *[S.l.]*: Libregraf, 2005.

FERNANDES, Cristiane Lima Terra. LIBRAS IN LAW AND SCHOOL PRACTICE: what we have and what we need. **Momento - Diálogos em Educação**, v. 31, n. 02, p. 528–553, 28 jul. 2022.

FOTEINOS, Konstantinos *et al.* **Visual Hand Gesture Recognition with Deep Learning: A Comprehensive Review of Methods, Datasets, Challenges and Future Research Directions**. arXiv, , 9 nov. 2025. Disponível em: <http://arxiv.org/abs/2507.04465>. Acesso em: 24 nov. 2025.

GAO, Feng *et al.* RC-DETR: Improving DETRs in crowded pedestrian detection via rank-based contrastive learning. **Neural Networks**, v. 182, p. 106911, fev. 2025.

GIRSHICK, Ross *et al.* Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *In*: 2014 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **2014 IEEE Conference on Computer Vision and Pattern Recognition**. Columbus, OH, USA: IEEE, jun. 2014. Disponível em: <http://ieeexplore.ieee.org/document/6909475/>. Acesso em: 25 nov. 2025.

GIRSHICK, Ross. Fast R-CNN. *In*: 2015 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV). **2015 IEEE International Conference on Computer Vision (ICCV)**. Santiago, Chile: IEEE, dez. 2015. Disponível em: <http://ieeexplore.ieee.org/document/7410526/>. Acesso em: 25 nov. 2025.

GONZÁLEZ-RODRÍGUEZ, Jaime-Rodrigo *et al.* Towards a Bidirectional Mexican Sign Language–Spanish Translation System: A Deep Learning Approach. **Technologies**, v. 12, n. 1, p. 7, 5 jan. 2024.

GOODFELLOW, Ian et al. Deep Learning: adaptive computation and machine learning series. [S.I]: The Mit Press, 2016. 800 p. Disponível em: https://www.deeplearningbook.org/. Acesso em: 02 fev. 2026.

GUPTA, Rinki. Expanding Indian Sign Language Recognition System using Class Incremental Learning. *In*: 2022 INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATION AND MATERIALS (ICACCM). **2022 International Conference on Advances in Computing, Communication and Materials (ICACCM)**. Dehradun, India: IEEE, 10 nov. 2022. Disponível em: <https://ieeexplore.ieee.org/document/10009218/>. Acesso em: 4 dez. 2025.

HE, Kaiming *et al.* Deep Residual Learning for Image Recognition. *In*: 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, NV, USA: IEEE, jun. 2016. Disponível em: <http://ieeexplore.ieee.org/document/7780459/>. Acesso em: 25 nov. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Cidades e Estados**. , 2022. Disponível em: <https://www.ibge.gov.br/cidades-e-estados>. Acesso em: 8 ago. 2025
INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Pesquisa Nacional de Saúde 2019: Tabela 8217 - Pessoas de 18 anos ou mais de idade, por avaliação do próprio estado de saúde, segundo o sexo e a cor ou raça**. Rio de Janeiro: Instituto Brasileiro de Geografia e Estatística, 2019. Disponível em: <https://sidra.ibge.gov.br/tabela/8217#resultado>.

ISMAIL, Mohammad H.; DAWWD, Shefa A.; ALI, Fakhradeen H. Dynamic hand gesture recognition of Arabic sign language by using deep convolutional neural networks. **Indonesian Journal of Electrical Engineering and Computer Science**, v. 25, n. 2, p. 952, 1 fev. 2022.

JIANG, Yujian *et al.* Multi-Category Gesture Recognition Modeling Based on sEMG and IMU Signals. **Sensors**, v. 22, n. 15, p. 5855, 5 ago. 2022.

KAREEM MURAD, Bothina; H. HASSIN ALASADI, Abbas. Advancements and Challenges in Hand Gesture Recognition: A Comprehensive Review. **Iraqi Journal for Electrical and Electronic Engineering**, v. 20, n. 2, p. 154–164, 15 dez. 2024.

KOTHADIYA, Deep *et al.* Deepsign: Sign Language Detection and Recognition Using Deep Learning. **Electronics**, v. 11, n. 11, p. 1780, 3 jun. 2022.

KUHN, H. W. The Hungarian method for the assignment problem. **Naval Research Logistics Quarterly**, v. 2, n. 1–2, p. 83–97, mar. 1955.

KUMARI, Diksha; ANAND, Radhey Shyam. Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. **Electronics**, v. 13, n. 7, p. 1229, 26 mar. 2024.

Libras - Aprender. [S.I.], 23 ago. 2012. Disponível em: <https://www.youtube.com/watch?v=AGEJRkxHG_s&t=1s>. Acesso em: 8 ago. 2025.

Libras - Sábado. [S.I.], 23 ago. 2012. Disponível em: <https://www.youtube.com/watch?v=JoDtyWV69oY&t=2s>. Acesso em: 8 ago. 2025.

Libras - Televisão. [S.I.], 18 out. 2012. Disponível em: . Acesso em: 8 ago. 2025 Libras - Trabalhar. [S.I.], 19 out. 2012. Disponível em: . Acesso em: 8 ago. 2025.

Libras - Trabalhar. [S.I.], 19 out. 2012. Disponível em:<https://www.youtube.com/watch?v=kMnAK7Zf1lE>. Acesso em: 8 ago. 2025.

LIN, Tsung-Yi *et al.* **Microsoft COCO: Common Objects in Context**. arXiv, , 21 fev. 2015. Disponível em: <http://arxiv.org/abs/1405.0312>. Acesso em: 26 nov. 2025.
LOSHCHILOV, Ilya; HUTTER, Frank. **Decoupled Weight Decay Regularization**. arXiv, , 4 jan. 2019. Disponível em: <http://arxiv.org/abs/1711.05101>. Acesso em: 26 nov. 2025.

MAHMUD, Hasan. **BdSL47: A Complete Depth-based Bangla Sign Alphabet and Digit Dataset**. Mendeley Data, , 5 nov. 2023. Disponível em: <https://data.mendeley.com/datasets/pbb3w3f92y/3>. Acesso em: 10 nov. 2025.

MAIOLINI, Sérgio Pereira; MAIOLINI, Iara Lopes. PERCURSO DA LÍNGUA BRASILEIRA DE SINAIS (LIBRAS) COMO DISCIPLINA CURRICULAR OBRIGATÓRIA NA UNIVERSIDADE FEDERAL DE MATO GROSSO. **Uniletras**, v. 45, p. 1–26, 2023.

MENEZES, Júlia Salles; FAIAD, Cristiane; SILVA, Lara Sthefane Pericoli. Adaptação Transcultural da Bateria Fatorial de Personalidade para Língua Brasileira de Sinais. **Psico-USF**, v. 29, p. e272225, 2024.

MENG, Depu *et al.* **Conditional DETR for Fast Training Convergence**. arXiv, , 29 set. 2023. Disponível em: <http://arxiv.org/abs/2108.06152>. Acesso em: 27 nov. 2025

MINISTÉRIO DO TRABALHO E EMPREGO (MTE). **Painel de informações da RAIS**. *[S.l.: S.n.]*. Disponível em: <https://app.powerbi.com/view?r=eyJrIjoiNjk3M2IwZDYtOGQzMS00YmE1LWE3M2MtZWRjODA4NTk3YTQ2IiwidCI6IjNlYzkyOTY5LTVhNTEtNGYxOC04YWM5LWVmOThmYmFmYTk3OCJ9>. Acesso em: 8 ago. 2025.

MINISTÉRIO DO TRABALHO E EMPREGO (MTE). **Painel de informações da RAIS**. [S.l.: S.n.]. Disponível em: <https://app.powerbi.com/view?r=eyJrIjoiNjk3M2IwZDYtOGQzMS00YmE1LWE3M2MtZWRjODA4NTk3YTQ2IiwidCI6IjNlYzkyOTY5LTVhNTEtNGYxOC04YWM5LWVmOThmYmFmYTk3OCJ9>. Acesso em: 8 ago. 2025.

MOHAMED, Aws Saood; HASSAN, Nidaa Flaih; JAMIL, Abeer Salim. Real-Time Hand Gesture Recognition: A Comprehensive Review of Techniques, Applications, and Challenges. **Cybernetics and Information Technologies**, v. 24, n. 3, p. 163–181, 1 set. 2024.

MUNKRES, James. Algorithms for the Assignment and Transportation Problems. **Journal of the Society for Industrial and Applied Mathematics**, v. 5, n. 1, p. 32–38, mar. 1957.

NASCIMENTO, Maria Isabel Do; TORRES, Rhian Costa; RIBEIRO, Klynsman Grisotto Faria. Tecnologias assistivas para deficiência visual e auditiva ofertadas aos estudantes de medicina no Brasil. **Revista Brasileira de Educação Médica**, v. 46, n. 1, p. e037, 2022.

OLIVEIRA MARTINS, Vanessa Regina De; LOPES, Maura Corcini. DIREITO LINGUÍSTICO-EDUCACIONAL PARA ALUNOS SURDOS E O "ALÉM-ACESSIBILIDADE". **Cadernos de Pesquisa**, v. 54, p. e10710, 2024.

OUDAH, Munir; AL-NAJI, Ali; CHAHL, Javaan. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. **Journal of Imaging**, v. 6, n. 8, p. 73, 23 jul. 2020.

PASZKE, Adam *et al.* **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. arXiv, 3 dez. 2019. Disponível em: <http://arxiv.org/abs/1912.01703>. Acesso em: 26 nov. 2025.

PISHARADY, Pramod Kumar; SAERBECK, Martin. Recent methods and databases in vision-based hand gesture recognition: A review. **Computer Vision and Image Understanding**, v. 141, p. 152–165, dez. 2015. .

PREATO, Dânlei De Oliveira *et al.* INCLUSÃO DO ALUNO SURDO NA REDE REGULAR DE ENSINO / INCLUSION OF THE DEAF STUDENT IN THE REGULAR EDUCATION NETWORK. **Brazilian Journal of Development**, v. 6, n. 9, p. 73692–73705, 2020.

PRECHELT, Lutz. Automatic early stopping using cross validation: quantifying the criteria. **Neural Networks**, [S.L.], v. 11, n. 4, p. 761-767, jun. 1998. Elsevier BV. http://dx.doi.org/10.1016/s0893-6080(98)00010-0.

QI, Delong *et al.* A Dataset and System for Real-Time Gun Detection in Surveillance Video Using Deep Learning. *In*: 2021 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS (SMC). **2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. Melbourne, Australia: IEEE, 17 out. 2021. Disponível em: <https://ieeexplore.ieee.org/document/9659207/>. Acesso em: 27 nov. 2025.

QI, Jing *et al.* Computer vision-based hand gesture recognition for human-robot interaction: a review. **Complex & Intelligent Systems**, v. 10, n. 1, p. 1581–1606, fev. 2024.

QI, Jing; XU, Kun; DING, Xilun. Approach to hand posture recognition based on hand shape features for human–robot interaction. **Complex & Intelligent Systems**, v. 8, n. 4, p. 2825–2842, ago. 2022.

QUADROS, Ronice Müller de; KARNOPP, Lodenir Becker. **Línguas de sinais brasileira: estudos língüísticos**. Porto Alegre: Artmed, 2004.

RASTGOO, Razieh; KIANI, Kourosh; ESCALERA, Sergio. Sign Language Recognition: A Deep Survey. **Expert Systems with Applications**, v. 164, p. 113794, fev. 2021.
REDMON, Joseph *et al.* You Only Look Once: Unified, Real-Time Object Detection. *In*: 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, NV, USA: IEEE, jun. 2016. Disponível em: <http://ieeexplore.ieee.org/document/7780460/>. Acesso em: 25 nov. 2025.

REDMON, Joseph et al. You Only Look Once: Unified, Real-Time Object Detection. In: 2016 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, jun. 2016. Disponível em: . Acesso em: 25 nov. 2025.

REDMON, Joseph; FARHADI, Ali. YOLO9000: Better, Faster, Stronger. *In*: 2017 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Honolulu, HI: IEEE, jul. 2017. Disponível em: <http://ieeexplore.ieee.org/document/8100173/>. Acesso em: 25 nov. 2025.

REDMON, Joseph; FARHADI, Ali. **YOLOv3: An Incremental Improvement**. arXiv, , 8 abr. 2018. Disponível em: <http://arxiv.org/abs/1804.02767>. Acesso em: 25 nov. 2025.

REN, Shaoqing *et al.* Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 6, p. 1137–1149, 1 jun. 2017.

RENOTTE, Nicholas. SignDETR: **a walkthrough on building a custom sign language detection model by training a DETR model from scratch**. [S. l.], 2024. Repositório GitHub. Disponível em: <https://github.com/nicknochnack/SignDETR>. Acesso em: 08 set. 2025.

REZATOFIGHI, Hamid *et al.* Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. *In*: 2019 IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. Long Beach, CA, USA: IEEE, jun. 2019 Disponível em: <https://ieeexplore.ieee.org/document/8953982/>. Acesso em: 26 nov. 2025.

RODRIGUES, Ailton Rodrigues; ZANCHETTIN, Cleber Zanchettin. **V-Librasil - A New Dataset with Signs in Brazilian Sign Language (Libras)**. IEEE DataPort, , *[S.d.]*. Disponível em: <https://ieee-dataport.org/documents/v-librasil-new-dataset-signs-brazilian-sign-language-libras>. Acesso em: 26 nov. 2025.

RODRIGUES, Tuane Telles. A relação semiótica entre a linguagem cartográfica e a língua brasileira de sinais. **GEOGRAFIA (Londrina)**, v. 29, n. 1, p. 231, 4 jan. 2020.

RHEINER, Jonas; KERGER, Daniel; DRÜPPEL, Matthias. From pixels to letters: A high-accuracy CPU-real-time American Sign Language detection pipeline. **Machine Learning with Applications**, [s. l.], v. 20, 100650, 2025. DOI: https://doi.org/10.1016/j.mlwa.2024.100650. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827025000337>.

SARMA, Debajit; BHUYAN, M. K. Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review. **SN Computer Science**, v. 2, n. 6, p. 436, nov. 2021.

SAROWAR, Selim *et al.* Hand Gesture Recognition Systems: A Review of Methods, Datasets, and Emerging Trends. **International Journal of Computer Applications**, v. 187, n. 2, p. 33, maio 2025.

SECRETARIA DA SEGURANÇA PÚBLICA. **1ª Del. de Polícia da Pessoa com Deficiência - 2019 A 2022 Histórico do Registro dos B.Os**. *[S.l.: S.n.]*. Disponível em: <https://basededadosdeficiencia.sp.gov.br/1-delegacia-da-pessoa-com-deficiencia/>. Acesso em: 8 set. 2025.

SELVARAJU, R. R. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. **International Journal of Computer Vision**, v. 128, p. 336–359, 2020. DOI: 10.1007/s11263-019-01228-7

SHI, Yuanyuan *et al.* Review of dynamic gesture recognition. **Virtual Reality & Intelligent Hardware**, v. 3, n. 3, p. 183–206, jun. 2021.

SHIN, Jungpil *et al.* Korean Sign Language Recognition Using Transformer-Based Deep Neural Network. **Applied Sciences**, v. 13, n. 5, p. 3029, 27 fev. 2023.

SHIN, Jungpil *et al.* A Methodological and Structural Review of Hand Gesture Recognition Across Diverse Data Modalities. **IEEE Access**, v. 12, p. 142606–142639, 2024.

SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. A survey on Image Data Augmentation for Deep Learning. **Journal of Big Data**, v. 6, n. 1, p. 60, dez. 2019.

SILVA, Rubia Carla Donda Da; GAVALDÃO, Natália; MARTINS, Sandra Eli Sartoreto De Oliveira. AS IMPLICAÇÕES DAS POLÍTICAS EDUCACIONAIS INCLUSIVAS PARA A FORMAÇÃO DE UNIVERSITÁRIOS SURDOS. **Muiraquitã - Revista de Letras e Humanidades**, v. 8, n. 1, p. 174–188, 2020.

SILVEIRA, Wellington *et al.* SynLibras: A Disentangled Deep Generative Model for Brazilian Sign Language Synthesis. *In*: 2022 35TH SIBGRAPI CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES (SIBGRAPI). **2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. Natal, Brazil: IEEE, 24 out. 2022. Disponível em: <https://ieeexplore.ieee.org/document/9991748/>. Acesso em: 24 nov. 2025.

SIMONYAN, Karen; ZISSERMAN, Andrew. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. arXiv, , 10 abr. 2015. Disponível em: <http://arxiv.org/abs/1409.1556>. Acesso em: 25 nov. 2025.

SOARES, Lidiane Sacramento. LIBRAS IN HEALTH TRAINING: IMPACTS ON CARE FOR DEAF PATIENTS. *In*: I CONGRESSO INTERNACIONAL MULTIDISCIPLINAR DE CIÊNCIAS DA SAÚDE - I CIMS. **I Congresso Internacional Multidisciplinar de Ciências da Saúde - I CIMS**. New Science Publishers, 22 nov. 2024. Disponível em: <https://periodicos.newsciencepubl.com/ans/article/view/1494>. Acesso em: 24 nov. 2025.

SOARES, Romulo Augusto Aires. **Detecção de Armas de Fogo usando DETR com Múltiplas Redes Neurais Autocoordenadas**. *[S.l.]*: Universidade Federal do Maranhão, 2 set. 2024.

ŠUMAK, Boštjan; BRDNIK, Saša; PUŠNIK, Maja. Sensors and Artificial Intelligence Methods and Algorithms for Human–Computer Intelligent Interaction: A Systematic Mapping Study. **Sensors**, v. 22, n. 1, p. 20, 21 dez. 2021.

VASWANI, Ashish *et al.* **Attention Is All You Need**. arXiv, , 2 ago. 2017. Disponível em: <http://arxiv.org/abs/1706.03762>. Acesso em: 9 nov. 2025.

WILCOXON, F. Individual comparisons by ranking methods. **Biometrics Bulletin**, v. 1, n. 6, p. 80–83, 1945.

WU, Meng. Gesture Recognition Based on Deep Learning: A Review. **EAI Endorsed Transactions on e-Learning**, v. 10, 7 mar. 2024.

XIANGZU, Han; FEI, Lu; GUOHUI, Tian (ORGS.). Sign Language Recognition Based on Lightweight 3D MobileNet-v2 and Knowledge Distillation. **International Conference on Electronic Technology and Information Science**, n. 7, p. 6, 2022.

XING, Yinghui *et al.* MS-DETR: Multispectral Pedestrian Detection Transformer With Loosely Coupled Fusion and Modality-Balanced Optimization. **IEEE Transactions on Intelligent Transportation Systems**, v. 25, n. 12, p. 20628–20642, dez. 2024.

YAO, Zhuyu *et al.* **Efficient DETR: Improving End-to-End Object Detector with Dense Prior**. arXiv, , 3 abr. 2021. Disponível em: <http://arxiv.org/abs/2104.01318>. Acesso em: 26 nov. 2025.

ZHANG, Hao *et al.* **DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection**. arXiv, , 11 jul. 2022. Disponível em: <http://arxiv.org/abs/2203.03605>. Acesso em: 26 nov. 2025.

ZHANG, Yanqiong; JIANG, Xianwei. Recent Advances on Deep Learning for Sign Language Recognition. **Computer Modeling in Engineering & Sciences**, v. 139, n. 3, p. 2399–2450, 2024.

ZHAO, Yian *et al.* **DETRs Beat YOLOs on Real-time Object Detection**. arXiv, , 3 abr. 2024. Disponível em: <http://arxiv.org/abs/2304.08069>. Acesso em: 27 nov. 2025.

ZHU, Xizhou *et al.* **Deformable DETR: Deformable Transformers for End-to-End Object Detection**. arXiv, , 18 mar. 2021. Disponível em: <http://arxiv.org/abs/2010.04159>. Acesso em: 26 nov. 2025.

APÊNDICE A – PARÂMETROS FONOLÓGICOS E CONFIGURAÇÕES DE SINAIS EM LIBRAS

Este apêndice reúne a descrição dos cinco parâmetros fonológicos dos 20 sinais de Libras selecionados para o estudo, conforme detalhado nas seções 2.1.2 e 2.1.4. Os parâmetros documentados compreendem a Configuração de Mão (CM), o Ponto de Articulação (PA), a Orientação da Palma (OM), o Movimento (M) e as Expressões Não Manuais (ENM), elementos fundamentais para a caracterização linguística dos sinais analisados.

O descritivo completo encontra-se disponível em repositório aberto: SANTOS, R. W. Descrição fonológica dos sinais de Libras para reconhecimento em atendimento emergencial. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.

As figuras a seguir apresentam a representação visual de cada um dos 20 sinais, totalizando 20 figuras com a descrição sistemática dos cinco parâmetros fonológicos.

APÊNDICE B – *SCRIPT* DE TREINAMENTO

Este apêndice reúne o script de treinamento utilizado para o desenvolvimento da arquitetura DETR-Custom proposta nesta dissertação, conforme detalhado na seção 3.4. O script automatiza o processo de treinamento do modelo, incluindo o carregamento do dataset de sinais em Libras, a configuração dos hiperparâmetros, o ciclo de otimização e o salvamento dos pesos ao final de cada época, garantindo a reprodutibilidade dos experimentos e a padronização das condições de treinamento.

O código-fonte completo e as dependências associadas encontram-se disponíveis em repositório aberto:

SANTOS, R. W. Script de treinamento para detecção de sinais em Libras utilizando arquitetura DETR-Custom. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.

APÊNDICE C – VERIFICAÇÃO AUTOMATIZADA DE *DATA LEAKAGE*

Este apêndice apresenta o script automatizado utilizado para a verificação de ausência de data leakage entre os subconjuntos de treino, validação e teste do dataset, conforme procedimento metodológico descrito na seção 3.3.4. O script compara os nomes de arquivo entre os três pares possíveis de partições (treino × validação, treino × teste e validação × teste), retornando a mensagem "NENHUM DATA LEAKAGE ENCONTRADO" em caso de sucesso ou, na hipótese de detecção, a listagem completa dos arquivos duplicados, discriminados por par de partições.

O código-fonte completo, que assegura a reprodutibilidade do procedimento de validação, encontra-se disponível em repositório aberto:

SANTOS, R. W. Script de verificação automatizada de data leakage em dataset de reconhecimento de sinais em Libras. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.

A execução do script é realizada por meio do comando bash verify_data_leakage.sh, sendo necessário o ajuste prévio da variável DATASET_PATH para o diretório local do dataset. O arquivo completo encontra-se no diretório scripts/verificacao/ do repositório, juntamente com instruções de uso e exemplos de saída esperada.

APÊNDICE D – EXEMPLOS DE IMAGENS ANALISADAS PARA MAPAS *CROSS-ATTENTION*

Este apêndice reúne exemplos das imagens utilizadas na análise de interpretabilidade por cross-attention do decoder, conforme apresentado na seção 4.6 da dissertação. O material

está organizado em três categorias, correspondentes às condições experimentais avaliadas: imagens com fundo chroma-key (imagens/fundo_verde/), imagens com fundos ruidosos gerados via prompts (imagens/fundo_ruidoso/) e imagens com fundo ruidoso acrescidas de terceiro elemento sintético (imagens/fundo_ruidoso_sintetico/).

O material completo encontra-se disponível em repositório aberto:

SANTOS, R. W. Imagens analisadas para mapas de cross-attention em reconhecimento de sinais de Libras. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.

ANEXO A – IMAGENS DO PORTAL RAIS E MÉTODO DE CONSULTA

Este anexo reúne o fluxo de consulta ao Painel de Informações da Relação Anual de Informações Sociais (RAIS), gerido pelo Ministério do Trabalho e Emprego (MTE), acessado em 2025, utilizado para o levantamento quantitativo de intérpretes de Libras formalmente registrados no Brasil. Os registros documentados compreendem as etapas de acesso ao sistema, a seleção dos filtros de busca para a ocupação de intérprete de língua de sinais e a leitura do quantitativo retornado.

O painel encontra-se disponível em:

https://app.powerbi.com/view?r=eyJrIjoiNjk3M2IwZDYtOGQzMS00YmE1LWE3M2MtZWRjODA4NTk3YTQ2IiwidCI6IjNlYzkyOTY5LTVhNTEtNGYxOC04YWM5LWVmOThmYmFmYTk3OCJ9

SANTOS, R. W. Procedimento de consulta ao Portal RAIS para levantamento de intérpretes de Libras. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.

ANEXO B – PESQUISA DAS IMAGENS SINTÉTICAS NO GOOGLE IMAGENS

Este anexo reúne o resultado do procedimento de verificação realizado no buscador Google Imagens para validação das imagens sintéticas geradas, com o objetivo de assegurar que não correspondam a pessoas reais, conforme descrito na seção 3.3.1 da dissertação. Os registros documentados compreendem as etapas de busca reversa por imagem para cada uma das representações sintéticas utilizadas no estudo, confirmando que nenhuma delas corresponde a indivíduos reais, o que assegura a natureza sintética do dataset e a conformidade ética da pesquisa.

Os resultados das buscas reversas encontram-se arquivados em três arquivos PDF no repositório aberto do estudo:

SANTOS, R. W. Verificação de imagens sintéticas via Google Imagens. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.

ANEXO C – *PROMPT*S DE GERAÇÃO DE FUNDOS RUIDOSOS

Este anexo reúne o conjunto de prompts utilizados para a geração sintética dos fundos ruidosos empregados no estudo, elaborados com o auxílio do modelo de inteligência artificial Gemini (Google) e renderizados no Google Labs ImageFX (modelo Imagen), conforme descrito na seção 3.3.1 da dissertação.

Os prompts documentados compreendem descrições textuais detalhadas de cenários visuais, organizados por categorias temáticas — incluindo Ambientes Urbanos, Comercial e Varejo, Condições Climáticas e Atmosféricas, Construção e Obras, Cultural e Histórico, Desastres e Resgate, Energia e Serviços Públicos, Espaços Educacionais, Esportes e Estádios, Feiras e Espetáculos, Forças de Segurança e Militar, Industrial e Manufatura, Médico e Hospitalar, Natureza e Paisagens, Pesquisa Científica e Laboratórios, Reciclagem e Resíduos, Restaurantes e Gastronomia, Tecnologia e Estúdios, e Transporte e Aviação.

Cada prompt foi estruturado com os campos: número de identificação, categoria, título descritivo e especificação visual detalhada, contendo elementos como texturas, condições de iluminação, padrões geométricos, cores predominantes e desafios específicos de segmentação (por exemplo, reflexos especulares, fios finos, fumaça, padrões repetitivos e baixo contraste). O objetivo foi garantir diversidade e controle sistemático nas variáveis de fundo para uma avaliação robusta do modelo de detecção.

O conjunto completo de prompts encontra-se arquivado em formato estruturado (CSV) no repositório aberto do estudo:

SANTOS, R. W. Prompts elaborados com Gemini para geração de fundos ruidosos no Labs ImageFX. Zenodo, 2026. DOI: 10.5281/zenodo.21069763. Disponível em: https://zenodo.org/records/21069763.