CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

CURSO DE ENGENHARIA DE COMPUTAÇÃO

HELTON POJO CONCEIÇÃO

REIDENTIFICAÇÃO DE PESSOAS EM REDES DE MÚLTIPLAS CÂMERAS:

Comparação entre arquiteturas híbrida e edge-first

Leopoldina

2026

HELTON POJO CONCEIÇÃO

REIDENTIFICAÇÃO DE PESSOAS EM REDES DE MÚLTIPLAS CÂMERAS:

Comparação entre arquiteturas híbrida e edge-first

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, do Campus Leopoldina, como parte do requisito para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Fabiano Pereira Bhering

Leopoldina

2026

Conceição, Helton Pojo.
C744r Reidentificação de pessoas em redes de múltiplas câmeras: comparação entre arquiteturas híbrida e edge-first. / Helton Pojo Conceição. – 2026.
71 f. : il., gráfs.

Orientador; Fabiano Pereira Bhering.
Bibliografia: f. 70-71.

Trabalho de Conclusão de Curso (Graduação) - Centro Federal de Educação Tecnológica de Minas Gerais, *Campus* Leopoldina, apresentada ao curso de Engenharia de Computação. Departamento de Computação, 2026.

1. Inteligência artificial. 2. Identificação humana. 3. Técnicas de vídeo. 4. Visão computacional. 5. Aprendizado de máquina. I. Bhering, Fabiano Pereira. II. Título.

CDU: 004.8

Elaboração da ficha catalográfica pela bibliotecária Luzia Adriana Damasceno, CRB 6º- 2305 / CEFET-MG *campus* Leopoldina.

HELTON POJO CONCEIÇÃO

REIDENTIFICAÇÃO DE PESSOAS EM REDES DE MÚLTIPLAS CÂMERAS :
Comparação entre arquiteturas híbrida e edge-first

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, do Campus Leopoldina, como parte do requisito para obtenção do título de Bacharel em Engenharia de Computação.

Aprovado em: 19 de junho de 2026

Banca Examinadora:

Prof. Dsc. Fabiano Pereira Bhering- Orientador
CEFET-MG

Prof. Dsc. Lindolpho de Oliveira Araújo Jr
CEFET-MG

Prof. Dsc. Fernando Garcia Diniz Campos Ferreira
CEFET-MG

(Assinaturas Digitais)

Leopoldina
2026

CÓPIA DE FOLHA DE ASSINATURAS Nº 1/2026 - DECOMLP (11.61.11)

(Nº do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 10/07/2026 13:34)
FABIANO PEREIRA BHERING
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOMLP (11.61.11)
Matrícula: ###564#0

(Assinado digitalmente em 10/07/2026 17:40)
FERNANDO GARCIA DINIZ CAMPOS FERREIRA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOMLP (11.61.11)
Matrícula: ###845#9

(Assinado digitalmente em 10/07/2026 00:28)
LINDOLPHO OLIVEIRA DE ARAUJO JUNIOR
PROFESSOR ENS BASICO TECN TECNOLOGICO
DEELP (11.61.04)
Matrícula: ###903#1

Visualize o documento original em https://sig.cefetmg.br/documentos/ informando seu número: ***1****, ano:* ***2026****, tipo:* ***CÓPIA DE FOLHA DE ASSINATURAS****, data de emissão:* ***09/07/2026*** *e o código de verificação:* ***56bd5d2b83***

Dedico este trabalho à minha mãe Edilene Leal Pojo, à minha tia Edileuza Leal Pojo e à minha avó Maria Canuta Leal Pojo, que sempre foram exemplos de resiliência e dedicação, e que se tornaram os pilares da minha formação como indivíduo.

AGRADECIMENTOS

Agradeço primeiramente à minha família, que me deu todo o suporte e a instrução para seguir resiliente e concluir a minha graduação. Em especial à minha mãe, Edilene Leal Pojo, à minha tia, Edileuza Leal Pojo, e à minha avó, Maria Canuta Leal Pojo, que sempre serão meus maiores exemplos.

Ao Centro Federal de Educação Tecnológica de Minas Gerais, Campus Leopoldina, e a todos os professores desta instituição, que colaboraram para a minha formação e foram exemplos de profissionalismo e competência. Em especial, ao meu orientador, Prof. Dr. Fabiano Pereira Bhering, pelo suporte e pela mentoria não apenas durante o desenvolvimento deste trabalho, mas ao longo de boa parte da minha formação. Àqueles de quem tive a oportunidade de me aproximar fora da sala de aula, agradeço pela amizade, pelos conselhos e pelas oportunidades que me proporcionaram nesta trajetória.

À minha companheira, Thailany, que esteve ao meu lado nos momentos mais difíceis e que, mesmo quando a caminhada parecia pesada demais, me lembrou do que eu era capaz. Obrigado por acreditar em mim antes mesmo de eu acreditar, por transformar dias comuns em algo mais leve e por ser, todos os dias, o melhor motivo para seguir em frente. Esta conquista também é sua.

E aos meus amigos, que tornaram esta trajetória mais leve ao me acompanharem em tantos desses desafios. Em especial, aos meus amigos Valéria, Luan, Rian, Rickson, Bruno, José Mário, Rogério, Eduardo e Igor.

RESUMO

Sistemas de videovigilância com múltiplas câmeras são amplamente empregados em segurança pública, controle de acesso e monitoramento de ambientes urbanos. A reidentificação de pessoas (*Person Re-Identification* (ReID)) é a tarefa de reconhecer um mesmo indivíduo em imagens capturadas por câmeras distintas, sendo essencial para o rastreamento contínuo em grandes áreas. Os principais desafios incluem variações de iluminação, pose e oclusão entre câmeras, além das restrições de largura de banda e latência impostas por redes sem fio. Nesse contexto, este trabalho apresenta o desenvolvimento e a avaliação comparativa de duas arquiteturas de processamento distribuído para reidentificação de pessoas (*person re-identification*) em redes de múltiplas câmeras. A arquitetura Híbrida executa a detecção de pessoas (YOLOv12n) nos nós de borda e delega a extração de *embeddings* (OSNet) e o *matching* ao servidor central. A arquitetura *Edge-First* executa a detecção e a extração de *embeddings* nos nós, transmitindo ao servidor apenas vetores compactos para o *matching* com a galeria global de identidades. Os experimentos foram conduzidos em diferentes *datasets* de múltiplas cameras, variando os limiares de similaridade. Os resultados demonstram que a arquitetura *Edge-First* foi a alternativa mais adequada para sistemas de ReID em redes sem fio, ao oferecer simultaneamente menor consumo de banda, menor latência e maior acurácia, sendo especialmente indicada para cenários com restrições de largura de banda nos quais dispositivos de borda dispõem de capacidade computacional suficiente para a extração local de características.

Palavras-chave: Reidentificação de Pessoas; Redes de Múltiplas Câmeras; *Edge Computing*; Visão Computacional; Aprendizado Profundo.

ABSTRACT

Multi-camera video surveillance systems are widely used in public safety, access control, and urban environment monitoring. Person re-identification (ReID) is the task of recognizing the same individual across images captured by different cameras, being essential for continuous tracking over large areas. Key challenges include variations in lighting, pose, and occlusion across cameras, as well as bandwidth and latency constraints imposed by wireless networks. In this context, this work presents the development and comparative evaluation of two distributed processing architectures for person re-identification in multi-camera networks. The Hybrid architecture performs person detection (YOLOv12n) on edge nodes and delegates embedding extraction (OSNet) and matching to the central server. The Edge-First architecture performs both detection and embedding extraction on the nodes, transmitting only compact vectors to the server for matching against the global identity gallery. Experiments were conducted on different multi-camera datasets, varying the similarity thresholds. Results show that the Edge-First architecture was the most suitable alternative for ReID systems over wireless networks, simultaneously offering lower bandwidth consumption, lower latency, and higher accuracy, being especially suited for bandwidth-constrained scenarios where edge devices have sufficient computational capacity for local feature extraction.

Keywords: Person Re-Identification. Multi-Camera Networks. Edge Computing. Computer Vision. Deep Learning.

TERMO DE RESPONSABILIDADE POR AUTORIA DO TRABALHO DE CONCLUSÃO DE CURSO

Eu, **HELTON POJO CONCEIÇÃO**, matrícula **20183019986**, discente do Curso de Engenharia de Computação, declaro para os devidos fins, que o presente Trabalho de Conclusão de Curso, de título **REIDENTIFICAÇÃO DE PESSOAS EM REDES DE MÚLTIPLAS CÂMERAS**, é de minha autoria e que estou ciente dos:

- Artigos 297 a 299 do Código Penal;
- Decreto-Lei n 2.848 de 7 de dezembro de 1940;
- Lei no 9.610, de 19 de fevereiro de 1998, sobre os Direitos Autorais;
- Regulamento Disciplinar Discente do CEFET-MG;
- E que plágio consiste na reprodução de obra alheia e submissão dela como trabalho próprio ou na inclusão, em trabalho próprio, de ideias, textos, tabelas ou ilustrações (quadros, figuras, gráficos, fotografias, retratos, lâminas, desenhos, organogramas, fluxogramas, plantas, mapas e outros) transcritos de obras de terceiros sem a devida e correta citação da referência.

Por ser verdade, e por ter ciência do referido artigo, firmo a presente declaração, isentando o(a) professor(a) orientador(a) **FABIANO PEREIRA BHERING** , os membros da banca examinadora e o Centro Federal de Educação Tecnológica de Minas Gerais, Campus Leopoldina, de qualquer responsabilidade.

Leopoldina 19 de junho de 2026

gov.br
Documento assinado digitalmente
HELTON POJO CONCEICAO
Data: 13/07/2026 09:42:28-0300
Verifique em https://validar.iti.gov.br

Assinatura do Discente

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AIN *Adaptive Instance Normalization*

AP *Average Precision*

CMC *Cumulative Matching Characteristic*

CNN *Convolutional Neural Network*

COCO *Common Objects in Context*

FIFO *First-In, First-Out*

FPS *Frames per Second*

HOG *Histograms of Oriented Gradients*

IoU *Intersection over Union*

JPEG *Joint Photographic Experts Group*

LGPD Lei Geral de Proteção de Dados

mAP *mean Average Precision*

MQTT *Message Queuing Telemetry Transport*

NPU *Neural Processing Unit*

ONNX *Open Neural Network Exchange*

OSNet *Omni-Scale Network*

ReID *Person Re-Identification*

RPN *Region Proposal Network*

SVM *Support Vector Machine*

TCP *Transmission Control Protocol*

UDP *User Datagram Protocol*

YOLO *You Only Look Once*

LISTA DE SÍMBOLOS

$\mathbf{a}, \mathbf{b}$	Vetores de *embedding* no espaço $\mathbb{R}^d$
d	Dimensionalidade do vetor de *embedding* ($d = 512$)
$\bar{\mathbf{g}}_j$	Centroide (vetor médio) da identidade j na galeria
G	Matriz de centroides da galeria ($G \in \mathbb{R}^{n \times 512}$)
n	Número de identidades na galeria
$\text{sim}(\mathbf{a}, \mathbf{b})$	Similaridade cosseno entre os vetores $\mathbf{a}$ e $\mathbf{b}$
τ	Limiar de similaridade para decisão de *matching*
Q	Número de consultas (*queries*) na avaliação
AP_i	*Average Precision* da i-ésima consulta
$P(k)$	Precisão na posição k do *ranking*
$\Delta r(k)$	Variação de *recall* na posição k
$\mathbb{1}[\cdot]$	Função indicadora (1 se a condição é verdadeira, 0 caso contrário)

SUMÁRIO

1 INTRODUÇÃO

O avanço tecnológico observado nas últimas décadas tem promovido transformações profundas na sociedade contemporânea, impactando diretamente os modos de vida, a organização social e os sistemas de segurança. Entre os fatores que se destacam nesse processo, está a crescente presença de câmeras e softwares inteligentes em ambientes urbanos e privados, fenômeno que reflete tanto a popularização de dispositivos de baixo custo quanto a evolução das áreas de visão computacional e inteligência artificial (SZEWCZYK et al., 2020).

Esses dispositivos, que anteriormente se limitavam a funções de registro de imagens para consulta posterior, passaram a desempenhar papéis mais sofisticados ao serem integrados a sistemas de análise em tempo real. Essa evolução viabiliza a identificação e a distinção de indivíduos em diferentes contextos, possibilitando aplicações voltadas ao monitoramento automatizado, ao controle de acesso e ao reforço da segurança pública e privada (FEI; HAN, 2023).

Nesse cenário, técnicas de aprendizado de máquina e redes neurais convolucionais têm se mostrado especialmente relevantes, uma vez que permitem a extração e interpretação de características visuais com alto grau de precisão. A literatura destaca a eficiência desses métodos em tarefas como reconhecimento facial, reidentificação de pessoas (ReID) e detecção de anomalias em ambientes monitorados (LIU et al., 2020).

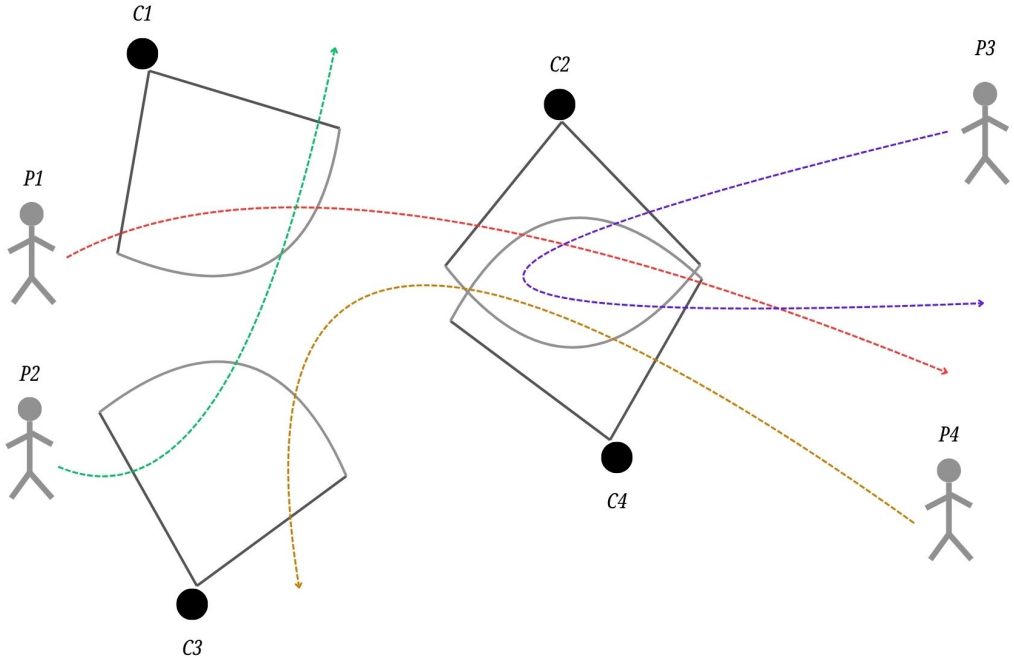
Diante dessas perspectivas, observa-se que o estudo e o desenvolvimento de sistemas inteligentes baseados em visão computacional constituem um campo de investigação promissor, que alia inovação tecnológica às demandas sociais contemporâneas por soluções de segurança e monitoramento mais eficazes. Pesquisas voltadas à implementação de tais sistemas têm potencial não apenas de ampliar a confiabilidade dos processos de vigilância, mas também de contribuir para a formulação de políticas públicas e estratégias empresariais orientadas à proteção e ao bem-estar coletivo.

Nesse contexto, o uso de múltiplas câmeras mostra-se especialmente relevante, uma vez que possibilita a ampliação da área monitorada, a coleta de dados mais precisos e a projeção da posição do indivíduo no espaço, contribuindo para análises mais completas e confiáveis.

As aplicações práticas da reidentificação de pessoas em ambientes multi-câmera abrangem diversos setores. No varejo inteligente, sistemas de ReID possibilitam o rastreamento de clientes entre diferentes seções de uma loja, viabilizando análises de fluxo, tempo de

permanência e a construção de mapas de calor comportamentais que orientam decisões estratégicas de *layout* e *marketing*. Em cidades inteligentes, a tecnologia contribui para o monitoramento urbano e a segurança pública, permitindo o rastreamento de indivíduos suspeitos entre câmeras de rua e a gestão de multidões em eventos de grande porte. No contexto de hospitais inteligentes, a reidentificação pode ser empregada no controle de acesso, no rastreamento de pacientes e profissionais de saúde, na detecção de permanência indevida em áreas restritas e na segurança de grupos vulneráveis, como idosos e crianças (YE et al., 2022). A Figura 1.1 ilustra esses cenários de aplicação.

Figura 1.1 – Exemplos de aplicações de reidentificação de pessoas em diferentes cenários.



Fonte: Elaborado pelo autor.

1.1 Justificativa

O crescimento do número de dispositivos de captura de vídeo em ambientes urbanos e corporativos, associado à necessidade de soluções inteligentes para segurança e monitoramento, tem impulsionado pesquisas voltadas à identificação e ao acompanhamento automático de indivíduos em múltiplas câmeras. Sob essa perspectiva, técnicas de ReID têm se destacado por permitir a atribuição de um identificador único a cada indivíduo, mesmo quando capturado em

diferentes ângulos, câmeras ou condições de iluminação (ZHENG et al., 2016).

O uso de arquiteturas baseadas em aprendizado profundo, como as redes convolucionais e modelos otimizados para ReID, possibilitou avanços significativos na acurácia da tarefa, permitindo superar desafios relacionados à variação de pose, oclusões parciais e similaridade de vestimentas (SUN et al., 2018; ZHOU et al., 2019). Além disso, a integração com técnicas eficientes de detecção de pessoas, como a família de modelos YOLO, tem permitido a implementação de sistemas em tempo real, requisito fundamental em cenários de segurança e análise comportamental (REDMON; FARHADI, 2018).

Entretanto, a aplicação prática desses sistemas em cenários compostos por múltiplas câmeras em redes introduz desafios adicionais relacionados à largura de banda disponível, à latência de comunicação, à privacidade dos dados e às restrições inerentes a sistemas embarcados de baixo custo. A execução de algoritmos de aprendizado de máquina nesses dispositivos é limitada por recursos computacionais reduzidos, como capacidade de processamento, memória e orçamento energético. Nesse contexto, arquiteturas baseadas em *edge computing* que realizam o processamento local de dados visuais apresentam-se como uma abordagem eficiente, pois minimizam o volume de informações transmitidas pela rede ao restringir o envio a representações vetoriais compactas, como *embeddings* extraídos por redes neurais profundas, em detrimento do streaming contínuo de vídeo. Além de otimizar o uso da infraestrutura de comunicação, essa estratégia contribui para a mitigação de riscos à privacidade, ao evitar a transmissão de dados brutos de imagem, preservando informações sensíveis dos indivíduos monitorados (WU et al., 2021).

Dessa forma, a presente pesquisa justifica-se pela relevância prática e científica de se desenvolver e avaliar diferentes arquitetura distribuída para ReID em múltiplas câmeras, considerando aspectos de eficiência, escalabilidade e segurança da informação. O estudo contribui para a área de visão computacional aplicada, fornecendo subsídios tanto para o aprimoramento de sistemas de monitoramento inteligentes quanto para futuras investigações sobre computação em borda e redes neurais otimizadas para dispositivos com recursos limitados.

1.2 Objetivo Geral

Desenvolver e avaliar soluções de *person re-identification* (ReID) em ambientes com múltiplas câmeras conectadas em rede, comparando uma arquitetura Híbrida, na qual o servidor centraliza a extração de características, e uma arquitetura *Edge-First*, na qual a detecção e a

extração de características ocorrem nos dispositivos de borda, reduzindo os dados transmitidos na rede, de forma a atribuir identificadores únicos a indivíduos, considerando eficiência, escalabilidade, privacidade e viabilidade prática.

1.3 Objetivo Específico

Para a execução do objetivo geral, foi necessária a divisão nos seguintes objetivos:

a) Implementar modelos de detecção e ReID adequados ao cenário de múltiplas câmeras, utilizando YOLO para detecção de pessoas e OSNet para extração de *embeddings*;

b) Projetar e implementar duas arquiteturas de sistema: a Híbrida e *Edge-First*;

c) Definir protocolos de comunicação e formatos de transmissão para envio de detecções e *embeddings*, garantindo eficiência e baixa latência;

d) Avaliar experimentalmente ambas as arquiteturas, mensurando métricas de acurácia (Rank-1, Rank-5, *mean Average Precision* (mAP)), consumo de recursos computacionais (CPU e memória) e latência e consumo de banda da rede;

e) Comparar os resultados obtidos entre as duas arquiteturas, analisando vantagens e limitações de cada abordagem em termos de desempenho e aplicabilidade em cenários reais de monitoramento.

1.4 Delimitação do Escopo

Este trabalho delimita-se ao desenvolvimento e à avaliação de um protótipo de software para reidentificação de pessoas em múltiplas câmeras, com as seguintes restrições:

a) **Protótipo em software**: o sistema foi implementado integralmente em Python, sem utilização de hardware embarcado dedicado (como NVIDIA Jetson, Raspberry Pi ou módulos com *Neural Processing Unit* (NPU)). Os nós de câmera e o servidor executam como processos independentes em uma única máquina.

b) **Vídeos pré-gravados**: os fluxos de vídeo processados são provenientes de *datasets* públicos (EPFL Terrace e PRID2011), não de câmeras físicas capturando cenas em tempo real.

c) **Comunicação via *localhost***: a transmissão de dados entre nós e servidor ocorre pela interface de *loopback* local, sem tráfego em redes reais. Essa configuração elimina variáveis como interferência de canal, perda de pacotes por congestionamento e variação de latência de rede, permitindo isolar o desempenho intrínseco das arquiteturas de processamento.

d) **Foco na comparação arquitetural**: o objetivo central é a comparação entre as arquiteturas Híbrida e *Edge-First* em termos de acurácia, latência, consumo de banda e uso de recursos computacionais, não a implantação de um sistema operacional em ambiente de produção.

Essas restrições são consistentes com o objetivo do trabalho: avaliar o impacto da distribuição de carga entre borda e servidor sobre o desempenho de ReID, em condições controladas e reprodutíveis. A extensão para hardware embarcado real e redes sem fio é indicada como direção para trabalhos futuros no Capítulo 5.

1.5 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos. O primeiro capítulo corresponde à introdução, apresentando a contextualização do tema, a justificativa, os objetivos e a delimitação do estudo. O segundo capítulo é dedicado à fundamentação teórica, na qual são abordados os conceitos fundamentais relacionados à visão computacional, reindentificação de pessoas, arquiteturas distribuídas e protocolos de comunicação. O terceiro capítulo trata do desenvolvimento do sistema, detalhando a implementação das arquiteturas Híbrida e *Edge-First*, os componentes de software e a metodologia experimental. O quarto capítulo apresenta os resultados obtidos e a análise comparativa entre as abordagens propostas. Por fim, o quinto capítulo expõe as conclusões, destacando as contribuições, limitações e perspectivas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo são apresentados os fundamentos teóricos que embasam o desenvolvimento deste trabalho, abrangendo os principais conceitos, técnicas e abordagens utilizadas em sistemas de ReID.

2.1 Fundamentos de Visão Computacional e Aprendizado Profundo

Os processos automáticos de identificação de pessoas são realizados por meio da extração e análise de informações visuais capturadas por câmeras. Esses processos envolvem o reconhecimento de padrões em imagens e vídeos, tarefa viabilizada pelo uso de técnicas avançadas de visão computacional com Aprendizado Profundo.

Nos últimos anos, o avanço das técnicas de Aprendizado Profundo (*Deep Learning)* tem provocado transformações significativas no campo da visão computacional. A disponibilidade de grandes volumes de dados, combinada com o aumento da capacidade computacional e o desenvolvimento de novas arquiteturas de redes neurais, permitiu melhorias substanciais no desempenho de tarefas como classificação de imagens, detecção de objetos, segmentação semântica e rastreamento de pessoas em múltiplas câmeras (GOODFELLOW; BENGIO; COURVILLE, 2016). Neste contexto as Redes Neurais Convolucionais (*Convolutional Neural Network* (CNN)s, do inglês *Convolutional Neural Networks*) apresentam uma arquitetura especializada para processar imagens e vídeos, fundamentais em tarefas de detecção, classificação e ReID, devido à sua capacidade de extrair descritores robustos e discriminativos de imagens, mesmo sob variações de iluminação, ângulo e aparência.

2.1.1 Redes Neurais Convolucionais

As CNNs constituem uma das principais bases para aplicações modernas de visão computacional. Elas são projetadas para processar dados com estrutura de grade, como imagens, explorando a correlação espacial entre os pixels por meio de operações de convolução (LECUN et al., 1998).

A arquitetura de uma CNN típica é composta por três tipos fundamentais de camadas, que se combinam para formar representações hierárquicas dos dados visuais:

a) **Camadas convolucionais**: constituem o núcleo da arquitetura. Cada camada aplica um conjunto de filtros (*kernels*) que percorrem a imagem de entrada realizando

operações de convolução, produzindo mapas de características (*feature maps*). Nas primeiras camadas, os filtros aprendem a detectar padrões simples, como bordas e texturas; nas camadas mais profundas, combinam essas características para representar estruturas complexas, como partes do corpo humano ou padrões de vestimenta. Essa extração automática de características hierárquicas é o que diferencia as CNNs dos métodos clássicos baseados em descritores projetados manualmente.

b) **Camadas de *pooling* (subamostragem)**: reduzem a dimensionalidade espacial dos mapas de características, mantendo as informações mais relevantes. A operação mais comum é o *max pooling*, que seleciona o valor máximo em cada região do mapa. Essa redução aumenta a invariância do modelo a pequenas translações e variações de escala, além de diminuir o custo computacional das camadas subsequentes.

c) **Camadas totalmente conectadas (*fully connected*)**: situam-se ao final da rede e combinam todas as características extraídas pelas camadas anteriores para produzir a saída desejada, seja uma classificação, seja um vetor de representação (*embedding*). Em redes de ReID como o OSNet, a última camada totalmente conectada produz o vetor de 512 dimensões que codifica a identidade visual da pessoa.

A popularização das CNNs ocorreu com o trabalho de Krizhevsky, Sutskever e Hinton (2012), que apresentou a rede AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), vencedora do ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Desde então, a evolução das arquiteturas seguiu direções complementares:

a) **VGGNet** (SIMONYAN; ZISSERMAN, 2014): demonstrou que o aumento da profundidade da rede, utilizando filtros convolucionais pequenos (3×3) empilhados em até 19 camadas, melhora significativamente a capacidade de representação. A VGG-16 tornou-se referência como *backbone* para extração de características em diversas tarefas de visão computacional.

b) **ResNet** (HE et al., 2016): introduziu as conexões residuais (*skip connections*), que permitem que o gradiente flua diretamente entre camadas distantes durante o treinamento. Essa inovação viabilizou o treinamento de redes com centenas de camadas (ResNet-152), superando o problema de degradação do gradiente em redes muito profundas. O princípio de conexões residuais é utilizado no OSNet, cujos

blocos *Omni-Scale Residual* combinam múltiplos caminhos com campos receptivos distintos.

c) **MobileNet** (HOWARD et al., 2017): propôs convoluções separáveis em profundidade (*depthwise separable convolutions*), que fatoram a convolução padrão em duas operações sequenciais, reduzindo drasticamente o número de parâmetros e operações. Essa abordagem orientou o desenvolvimento de modelos eficientes para dispositivos de borda, filosofia que também guia o YOLOv12n utilizado neste trabalho, que opera com apenas 6,1 milhões de parâmetros.

Inicialmente, as Redes Neurais Convolucionais (CNNs) dominaram a área de visão computacional devido à sua capacidade de explorar relações espaciais locais presentes nas imagens por meio de operações de convolução. Essas arquiteturas foram responsáveis por avanços expressivos em diversos problemas clássicos da área, como reconhecimento de objetos e detecção em tempo real. A partir de modelos como AlexNet, VGG e ResNet, tornou-se possível construir representações hierárquicas de imagens capazes de capturar desde características simples até padrões visuais complexos.

Com o amadurecimento das CNNs, surgiram arquiteturas mais especializadas para tarefas específicas de visão computacional. Entre elas destacam-se modelos de detecção de objetos em tempo real, como a família YOLO, que introduziu abordagens capazes de realizar detecção em uma única etapa, reduzindo significativamente o tempo de inferência e permitindo aplicações em cenários que exigem processamento em tempo real, como sistemas de vigilância e monitoramento inteligente (REDMON et al., 2016). Esses avanços foram fundamentais para viabilizar aplicações práticas em ambientes com múltiplas câmeras e processamento contínuo de vídeo.

Mais recentemente, uma nova geração de arquiteturas baseada em *Transformers* tem ganhado destaque na visão computacional. Originalmente desenvolvidos para tarefas de processamento de linguagem natural, os *Transformers* foram adaptados para imagens por meio de modelos como o *Vision Transformer* (ViT) (DOSOVITSKIY et al., 2021), que utiliza mecanismos de *self-attention* para modelar relações globais entre diferentes regiões da imagem. Essa abordagem permite capturar dependências de longo alcance que nem sempre são facilmente aprendidas por redes convolucionais tradicionais.

Além dos modelos puramente baseados em *Transformers*, pesquisas recentes também exploram arquiteturas híbridas, que combinam CNNs e mecanismos de atenção. Essas

abordagens buscam aproveitar simultaneamente a eficiência das convoluções na captura de padrões locais e a capacidade dos *Transformers* de modelar dependências globais entre regiões da imagem. Essa combinação tem demonstrado resultados promissores em diversas tarefas, como detecção de objetos, segmentação e análise de vídeo. Um exemplo recente é o YOLOv12, que integra mecanismos de atenção de área a uma arquitetura predominantemente convolucional (TIAN; YE; DOERMANN, 2025).

Em conjunto, esses avanços têm ampliado significativamente as capacidades dos sistemas de visão computacional modernos. O desenvolvimento de arquiteturas mais eficientes e robustas, aliado a novas estratégias de treinamento e disponibilidade de grandes conjuntos de dados, tem permitido a aplicação dessas técnicas em cenários cada vez mais complexos, incluindo sistemas distribuídos de monitoramento por múltiplas câmeras, veículos autônomos, análise de comportamento humano e identificação de indivíduos em ambientes de vigilância inteligente (SZEWCZYK et al., 2020).

2.2 Detecção de Objetos e a Família YOLO

A detecção de objetos em imagens e vídeos é uma etapa fundamental em sistemas de vigilância multi-câmera, pois precede qualquer processo de reidentificação de pessoas. A qualidade e a velocidade da detecção influenciam diretamente o desempenho do sistema como um todo. Nesta seção são revisados os principais marcos na evolução dos detectores de objetos, desde abordagens clássicas até os detectores de estágio único da família YOLO, utilizados neste trabalho.

2.2.1 Evolução dos detectores de objetos

Os primeiros detectores automáticos de objetos em imagens baseavam-se em características visuais projetadas manualmente (*hand-crafted features*). O trabalho seminal de Viola e Jones (2001) propôs um detector de faces em cascata, utilizando características tipo Haar e um classificador AdaBoost, capaz de operar em tempo real para a época (VIOLA; JONES, 2001). Posteriormente, Dalal e Triggs (2005) introduziram o descritor *Histograms of Oriented Gradients* (HOG), combinado com um classificador *Support Vector Machine* (SVM), que se tornou referência para detecção de pedestres (DALAL; TRIGGS, 2005).

Embora eficazes em cenários controlados, esses métodos apresentavam limitações significativas em condições variáveis de iluminação, pose e escala, além de exigirem um processo

manual de engenharia de características para cada domínio de aplicação. A ascensão das redes neurais convolucionais, a partir de 2012, trouxe uma mudança de paradigma: em vez de projetar descritores manualmente, os modelos passaram a aprender automaticamente representações hierárquicas dos dados (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

No contexto de detecção de objetos, surgiram inicialmente abordagens de dois estágios, como a família R-CNN (*Region-based Convolutional Neural Networks*). O modelo Faster R-CNN (REN et al., 2015) introduziu a *Region Proposal Network* (RPN), que gera propostas de regiões de interesse diretamente a partir da rede neural, eliminando a necessidade de algoritmos externos de segmentação. Apesar da elevada acurácia, os detectores de dois estágios apresentavam tempo de inferência incompatível com aplicações em tempo real, especialmente em dispositivos com recursos computacionais limitados.

Essas limitações motivaram o desenvolvimento dos detectores de estágio único, que unificam a localização e a classificação de objetos em uma única passagem pela rede, resultando em ganhos substanciais de velocidade, conforme detalhado na Subseção 2.2.2.

2.2.2 Família YOLO

A família YOLO foi proposta por Redmon et al. (2016) como uma abordagem radicalmente diferente para detecção de objetos (REDMON et al., 2016). Em vez de dividir o processo em propostas de região seguidas de classificação, o YOLO trata a detecção como um problema de regressão: a imagem de entrada é dividida em uma grade, e cada célula prediz simultaneamente as coordenadas das *bounding boxes*, as probabilidades de classe e os escores de confiança, em uma única passagem pela rede.

Essa formulação confere ao YOLO vantagens significativas em termos de velocidade. Enquanto detectores de dois estágios como o Faster R-CNN processavam cerca de 7 *frames* por segundo (*Frames per Second* (FPS)), o YOLOv1 original alcançava 45 FPS, tornando-o adequado para aplicações em tempo real (REDMON et al., 2016). Versões subsequentes aprimoraram tanto a acurácia quanto a eficiência do modelo:

a) **YOLOv3** (REDMON; FARHADI, 2018) introduziu predições em múltiplas escalas e o *backbone* Darknet-53, melhorando a detecção de objetos pequenos;

b) As versões **YOLOv5** a **YOLOv8**, desenvolvidas pela comunidade e pela Ultralytics, incorporaram técnicas modernas como *anchor-free detection*, aumento de dados

avançado (*mosaic augmentation*), e otimizações de treinamento como *cosine learning rate scheduling*;

c) **YOLOv12** (TIAN; YE; DOERMANN, 2025), a versão mais recente utilizada neste trabalho, introduziu uma arquitetura centrada em mecanismos de atenção (*attention-centric*), substituindo parcialmente as convoluções tradicionais por blocos de *area attention* que capturam dependências de longo alcance mantendo a eficiência computacional.

A arquitetura interna do YOLOv12 segue a estrutura de três estágios comum aos detectores modernos de estágio único:

1. ***Backbone* (extrator de características)**: rede convolucional profunda que processa a imagem de entrada e gera mapas de características em múltiplas escalas. No YOLOv12, o *backbone* incorpora blocos de *area attention* que capturam dependências de longo alcance, substituindo parcialmente as convoluções tradicionais por mecanismos de atenção.

2. ***Neck* (fusão de escalas)**: módulo que combina os mapas de características de diferentes escalas produzidos pelo *backbone*, permitindo que o detector localize tanto objetos grandes quanto pequenos. Utiliza conexões *top-down* e *bottom-up* para propagar informações entre os diferentes níveis de resolução.

3. ***Head* (predição)**: módulo final que, para cada posição da grade de saída, prediz simultaneamente as coordenadas da *bounding box*, o escore de confiança e a probabilidade de classe. No YOLOv12, a *head* utiliza uma formulação *anchor-free*, eliminando a necessidade de âncoras pré-definidas.

No sistema implementado neste trabalho, foi utilizado o modelo YOLOv12n, onde o sufixo "n" indica a variante *nano*, a mais compacta da família. Essa variante é projetada especificamente para execução em dispositivos com recursos computacionais limitados, operando com aproximadamente 6,1 milhões de parâmetros e 9,3 GFLOPs (*giga floating-point operations per second*). Para comparação, a variante YOLOv12x (*extra-large*) possui 59,1 milhões de parâmetros e 199,0 GFLOPs, evidenciando a redução de complexidade da versão *nano*. O modelo é pré-treinado no *dataset Common Objects in Context* (COCO) (LIN et al., 2014), que contém mais de 330 mil imagens anotadas em 80 classes de objetos, incluindo a classe "pessoa" utilizada

neste trabalho. O pré-treinamento permite que o modelo já possua capacidade de detecção adequada sem necessidade de treinamento adicional no domínio específico de vigilância.

Nos experimentos, o modelo opera com um limiar de confiança de 0,7, filtrando apenas detecções da classe "pessoa". Essa configuração equilibra a acurácia da detecção com a necessidade de processamento em tempo real nos nós de borda, onde o modelo YOLO é executado em ambas as arquiteturas avaliadas.

A escolha do YOLO como detector justifica-se pela sua velocidade de inferência, pela disponibilidade de modelos pré-treinados no COCO e pela integração nativa com a biblioteca Ultralytics, que oferece uma interface unificada em Python para treinamento, inferência e exportação de modelos.

2.3 Reidentificação de Pessoas

A reidentificação de pessoas (ReID) é a tarefa de reconhecer um mesmo indivíduo em imagens capturadas por câmeras diferentes, em momentos distintos. Esta seção apresenta a definição do problema, os principais desafios, as técnicas de extração de *embeddings*, as métricas de avaliação e o mecanismo de galeria e *matching* utilizado neste trabalho.

2.3.1 Conceitos e importância no contexto de vigilância multi-câmera

Formalmente, dado um conjunto de câmeras $C = \{c_1, c_2, \ldots, c_N\}$ e uma imagem de consulta (*probe*) de uma pessoa capturada pela câmera c_i, o problema de ReID consiste em encontrar imagens do mesmo indivíduo em um banco de referência (*gallery*) obtido pelas demais câmeras c_j, com $j \neq i$ (ZHENG et al., 2016).

No contexto de vigilância multi-câmera, a reidentificação é essencial para manter a continuidade do rastreamento de indivíduos quando estes transitam entre os campos de visão de diferentes câmeras. Diferentemente do rastreamento intra-câmera, que pode explorar a continuidade temporal do vídeo, a ReID *cross-câmera* enfrenta descontinuidades espaciais e temporais, exigindo representações visuais robustas que capturem a identidade do indivíduo de forma invariante às condições de captura (ZHENG et al., 2016).

A importância prática da ReID estende-se a diversas aplicações, desde a segurança pública, como monitoramento de grandes eventos e investigação forense, até a análise de fluxo de pessoas em ambientes comerciais e a gestão de tráfego em centros urbanos (SZEWCZYK et al.,

2020). Com a proliferação de câmeras de vigilância, a demanda por sistemas automatizados de ReID tem crescido significativamente, impulsionando o desenvolvimento de métodos baseados em aprendizado profundo.

2.3.2 Principais desafios

A reidentificação de pessoas é considerada um dos problemas mais desafiadores em visão computacional, devido à combinação de fatores que dificultam o reconhecimento visual de indivíduos entre câmeras distintas (YE et al., 2022). Os principais desafios incluem:

a) **Variação de pose:** uma mesma pessoa pode ser observada de frente, de costas ou de perfil em diferentes câmeras, alterando significativamente a aparência visual percebida;

b) **Variação de iluminação:** diferenças nas condições de iluminação entre câmeras, como ambientes internos versus externos, ou variações ao longo do dia, modificam as cores e o contraste das imagens;

c) **Oclusão parcial:** objetos, outras pessoas ou elementos do ambiente podem obstruir partes do corpo do indivíduo, dificultando a extração de características completas;

d) **Baixa resolução:** câmeras de vigilância frequentemente capturam imagens em baixa resolução, especialmente quando o indivíduo está distante, resultando em perda de detalhes discriminativos;

e) **Vestimentas similares:** indivíduos diferentes usando roupas de cores e padrões semelhantes podem ser confundidos pelo sistema, elevando a taxa de falsos positivos;

f) **Mudança de aparência:** alterações como remoção de casaco, uso de acessórios ou mudança de postura entre câmeras adicionam variabilidade intra-classe.

Esses desafios tornam insuficientes abordagens baseadas em correspondência direta de pixels ou em características visuais simples, motivando o uso de representações profundas (*deep embeddings*) que codificam a identidade visual de forma compacta e robusta às variações mencionadas.

2.3.3 Extração de embeddings com OSNet

A abordagem predominante para ReID baseada em *deep learning* consiste em treinar uma rede neural para mapear imagens de pessoas em vetores de representação (*embeddings*) em

um espaço de alta dimensão, de modo que *embeddings* de imagens do mesmo indivíduo sejam próximos entre si e distantes dos de outros indivíduos (YE et al., 2022). Diversas arquiteturas têm sido propostas para essa tarefa, como a PCB (*Part-based Convolutional Baseline*) (SUN et al., 2018), que divide o corpo em partes horizontais para extração de características locais, e o *framework* FastReID (HE et al., 2023), que oferece uma plataforma modular para experimentação com diferentes *backbones* e estratégias de treinamento.

Neste trabalho, foi adotado o modelo OSNet, proposto por Zhou et al. (2019) (ZHOU et al., 2019). A principal inovação do OSNet é a capacidade de aprender características em múltiplas escalas (*omni-scale features*) de forma unificada, por meio de blocos denominados *Omni-Scale Residual Blocks*. Cada bloco contém múltiplos fluxos (*streams*) com diferentes tamanhos de campo receptivo, cujas saídas são dinamicamente combinadas por um mecanismo de *channel-wise attention*. Essa arquitetura permite capturar simultaneamente detalhes finos, como texturas e padrões de roupa, e características globais, como a silhueta e a proporção corporal.

A variante utilizada, OSNet x1_0, produz um vetor de *embedding* de 512 dimensões para cada imagem de entrada. Apesar de sua compacidade, com aproximadamente 2,2 milhões de parâmetros, o modelo atinge resultados competitivos com arquiteturas significativamente maiores nos principais *benchmarks* de ReID (ZHOU et al., 2019). Essa combinação de acurácia e eficiência torna o OSNet particularmente adequado para implantação em dispositivos de borda.

Para garantir portabilidade entre diferentes plataformas, tanto nos nós de borda quanto no servidor, o modelo OSNet foi exportado para o formato *Open Neural Network Exchange* (ONNX) e executado via ONNX Runtime. Essa estratégia elimina a dependência de *frameworks* de treinamento (como PyTorch) no ambiente de produção e permite otimizações específicas de hardware, incluindo aceleração por GPU ou NPU quando disponível.

2.3.4 Métricas de Avaliação e Matching por Similaridade Cosseno

A avaliação de sistemas de ReID segue protocolos bem estabelecidos na literatura, baseados no paradigma *probe* × *gallery*: dada uma imagem de consulta (*probe*), o sistema ordena todas as imagens de referência (*gallery*) por similaridade e verifica se as correspondências corretas aparecem nas primeiras posições do *ranking* (ZHENG et al., 2016).

As principais métricas utilizadas são:

a) **Rank-k (*Cumulative Matching Characteristic* (CMC)):** indica a probabilidade de a identidade correta aparecer entre os k primeiros resultados do *ranking*. O Rank-1 é a métrica mais reportada e corresponde à acurácia de identificação na primeira posição. O Rank-5 captura a probabilidade de acerto nas cinco primeiras posições, sendo útil quando o sistema apresenta as melhores correspondências para validação humana.

Formalmente, para um conjunto de Q consultas:

$$\text{Rank-}k = \frac{1}{Q} \sum_{i=1}^{Q} 1 \left[\text{rank}_i \in \text{top-}k\right] \tag{2.1}$$

onde $\mathbb{1}[\cdot]$ é a função indicadora.

b) **mAP:** avalia a qualidade do *ranking* como um todo, não apenas as primeiras posições. Para cada consulta, calcula-se a *Average Precision* (AP), que corresponde à área sob a curva precisão-*recall*. A AP de uma consulta individual é definida como:

$$\text{AP} = \sum_{k=1}^{N} P(k) \cdot \Delta r(k) \tag{2.2}$$

onde N é o número total de itens na galeria, $P(k)$ é a precisão na posição k do *ranking* (proporção de correspondências corretas entre as k primeiras posições) e $\Delta r(k)$ é a variação de *recall* na posição k (igual a $1/R$ se o item na posição k é uma correspondência correta, onde R é o número total de correspondências corretas, e zero caso contrário). Intuitivamente, a AP penaliza *rankings* que posicionam identidades corretas em posições mais baixas: se a identidade correta aparece na primeira posição, a contribuição para a AP é máxima; se aparece na centésima posição, a contribuição é significativamente menor.

O mAP é então obtido pela média aritmética das APs sobre todas as Q consultas:

$$\text{mAP} = \frac{1}{Q} \sum_{i=1}^{Q} \text{AP}_i \tag{2.3}$$

A distinção entre AP e mAP é relevante: a AP avalia o desempenho do sistema para uma consulta específica, enquanto o mAP agrega os resultados sobre todas as consultas, fornecendo uma medida global da qualidade de recuperação do sistema. Valores de mAP próximos de 1,0 indicam que o sistema posiciona consistentemente as identidades corretas nas primeiras posições do *ranking* para todas as consultas.

Enquanto o Rank-1 reflete a precisão do sistema em cenários de identificação direta, o mAP fornece uma visão mais completa da capacidade de recuperação, penalizando *rankings* que posicionam identidades corretas em posições intermediárias (YE et al., 2022). Neste trabalho, ambas as métricas são reportadas para os dois *datasets* avaliados, nos quatro limiares de similaridade testados.

O componente central do sistema de ReID é a galeria de identidades, uma estrutura de dados mantida no servidor que armazena os *embeddings* acumulados ao longo do tempo para cada pessoa identificada. A cada nova detecção, o sistema compara o *embedding* recebido com as representações existentes na galeria para decidir se o indivíduo corresponde a alguém já conhecido ou se trata de uma nova identidade.

2.3.5 Representação no espaço de embeddings

Cada *embedding* gerado pelo OSNet pode ser interpretado como um ponto em um espaço vetorial de 512 dimensões ($\mathbb{R}^{512}$). Nesse espaço, imagens de um mesmo indivíduo tendem a formar agrupamentos (*clusters*) próximos entre si, enquanto imagens de indivíduos distintos ficam afastadas. A distância entre dois *embeddings* serve, portanto, como medida de similaridade visual entre as pessoas representadas.

2.3.6 Similaridade cosseno

A métrica adotada neste trabalho para comparação de *embeddings* é a similaridade cosseno, definida como:

$$\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|} = \frac{\sum_{i=1}^{d} a_i \, b_i}{\sqrt{\sum_{i=1}^{d} a_i^2} \, \sqrt{\sum_{i=1}^{d} b_i^2}} \tag{2.4}$$

onde $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ são dois vetores de *embedding* e $d = 512$. O resultado varia no intervalo $[-1, 1]$, onde valores próximos de 1 indicam alta similaridade e valores próximos de 0 ou negativos indicam dissimilaridade.

Na prática, como os *embeddings* do OSNet são normalizados (norma unitária), o cálculo da similaridade cosseno reduz-se ao produto interno:

$$\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a} \cdot \mathbf{b} \quad \text{quando} \quad \|\mathbf{a}\| = \|\mathbf{b}\| = 1 \tag{2.5}$$

Essa simplificação permite que o *matching* seja realizado de forma vetorizada utilizando operações de álgebra linear do NumPy, comparando um *embedding* de consulta contra toda a galeria em uma única operação matricial, aspecto detalhado no Capítulo 3.

2.3.7 Limiar de similaridade

A decisão de associar uma detecção a uma identidade existente ou criar uma nova identidade depende de um limiar de similaridade (τ). Se a maior similaridade entre o *embedding* de consulta e as identidades na galeria excede τ, a detecção é associada à identidade correspondente; caso contrário, uma nova identidade é criada:

$$\text{decisão} = \begin{cases} \text{associar a } \arg\max_j \text{sim}(\mathbf{q}, \bar{\mathbf{g}}_j), & \text{se } \max_j \text{sim}(\mathbf{q}, \bar{\mathbf{g}}_j) \geq \tau \\ \text{criar nova identidade}, & \text{caso contrário} \end{cases} \tag{2.6}$$

onde $\mathbf{q}$ é o *embedding* de consulta e $\bar{\mathbf{g}}_j$ é o *embedding* médio da identidade j na galeria. A escolha do valor de τ tem impacto direto na acurácia do sistema: valores altos de τ tornam o *matching* mais exigente: o sistema requer alta similaridade para associar uma detecção a uma identidade existente, reduzindo fusões incorretas mas aumentando a fragmentação de uma mesma pessoa em múltiplas entradas na galeria. Valores baixos de τ, por outro lado, favorecem a associação, reduzindo a fragmentação, mas elevando o risco de fundir indivíduos distintos sob uma mesma identidade. A análise de sensibilidade ao limiar é apresentada no Capítulo 4.

2.3.8 Políticas de descarte da galeria

Para evitar crescimento ilimitado da galeria, o que comprometeria tanto a memória quanto o tempo de *matching*, o sistema implementa duas políticas de descarte baseadas no princípio *First-In, First-Out* (FIFO):

a) **Limite por identidade:** cada identidade armazena no máximo 64 *embeddings*. Quando esse limite é excedido, o *embedding* mais antigo é descartado. Essa política mantém a representação atualizada para pessoas observadas por longos períodos.

b) **Limite global:** a galeria suporta no máximo 500 identidades. Ao atingir esse teto, a identidade com observação mais antiga é removida integralmente. Essa restrição garante que erros acumulados de *matching*, que poderiam criar centenas de identidades espúrias, não degradem o desempenho do sistema ao longo do tempo.

Essas políticas são configuráveis por meio dos parâmetros `max_gallery_per_person` e `max_gallery_size`, permitindo ajuste conforme as características do cenário de implantação.

2.4 Paradigmas de Processamento Distribuído

A definição de onde executar cada etapa do *pipeline* de ReID, detecção, extração de *embeddings* e *matching*, constitui uma decisão arquitetural central em sistemas multi-câmera. Nesta seção são revisados os conceitos de computação centralizada e *edge computing*, e apresentadas as duas arquiteturas avaliadas neste trabalho.

2.4.1 Computação Centralizada, Híbrida e Edge Computing (Edge-First)

Na abordagem tradicional de processamento centralizado, os dados brutos (vídeo ou imagens) são transmitidos dos dispositivos de captura para um servidor central, que executa todas as etapas de análise. Embora essa estratégia concentre o poder computacional e simplifique a manutenção, ela apresenta limitações em cenários de vigilância distribuída: o volume de dados transmitidos cresce linearmente com o número de câmeras, gerando demanda significativa de largura de banda; a latência é limitada pelo tempo de transmissão somado ao tempo de processamento no servidor; e o servidor torna-se um ponto único de falha (SHI et al., 2016).

O paradigma de *edge computing* (computação de borda) propõe uma alternativa: parte do processamento é migrada para dispositivos próximos à fonte dos dados, os chamados *nós de borda*, reduzindo a quantidade de dados que precisa ser transmitida e a latência percebida (SATYANARAYANAN, 2017). No contexto de vigilância por vídeo, essa abordagem traz benefícios adicionais de privacidade, uma vez que os *frames* brutos podem ser processados localmente e apenas informações derivadas (como *embeddings* ou eventos detectados) são transmitidas ao servidor (LIU et al., 2020).

A viabilidade da computação de borda para tarefas de visão computacional tem crescido com a disponibilização de hardware embarcado com aceleradores de inferência, como o NVIDIA Jetson e aceleradores NPU em dispositivos como o Raspberry Pi 5. Esses avanços permitem a execução de modelos de *deep learning* diretamente nos nós, com latência e consumo de energia aceitáveis para aplicações em tempo real.

2.4.1.1 Arquitetura Híbrida

Uma das arquiteturas adotadas neste trabalho segue um modelo Híbrido, no qual os nós de borda executam apenas a detecção de pessoas (YOLO), enquanto o servidor central é responsável pela extração de *embeddings* (OSNet) e pelo *matching* com a galeria de identidades.

Nessa configuração, cada nó processa os *frames* de vídeo localmente, detecta as pessoas presentes e transmite ao servidor os recortes (*crops*) das detecções no formato *Joint Photographic Experts Group* (JPEG) comprimido, via datagrama UDP. O servidor recebe os *crops*, extrai o vetor de *embedding* de cada um utilizando o modelo OSNet, e realiza o *matching* vetorizado com a galeria global.

A vantagem dessa abordagem é a menor carga computacional nos nós, que executam apenas o modelo YOLO. Entretanto, ela apresenta desvantagens: (i) a transmissão de imagens JPEG consome mais banda do que vetores numéricos compactos; (ii) a compressão JPEG pode degradar a qualidade visual do *crop*, afetando a extração de *embeddings* no servidor; e (iii) o servidor acumula a carga de inferência OSNet de todos os nós simultaneamente, podendo gerar gargalos de processamento quando o número de câmeras cresce.

2.4.1.2 Arquitetura Edge-First

Uma outra de arquitetura adotada neste trabalho, denominada *Edge-First*, distribui tanto a detecção quanto a extração de *embeddings* para os nós de borda. Cada nó executa o YOLO para detecção e o OSNet para extração do vetor de representação, transmitindo ao servidor apenas o *embedding* de 512 dimensões (2.048 bytes fixos, em formato `float32`) via UDP.

O servidor, nessa configuração, atua exclusivamente como agregador de identidades: recebe os *embeddings*, realiza o *matching* vetorizado com a galeria global e atualiza as identidades. Essa divisão de responsabilidades resulta em:

a) **Menor consumo de banda:** cada detecção transmite exatamente 2.048 bytes, independentemente do tamanho ou da resolução do *crop*, em contraste com os 2–9 KB variáveis do JPEG na arquitetura Híbrida;

b) **Menor latência no servidor:** ao eliminar a etapa de inferência OSNet no lado do servidor, o tempo de processamento por detecção reduz-se ao *matching* vetorial, que é computacionalmente trivial;

c) **Preservação da qualidade visual:** o *embedding* é extraído diretamente do *crop*

original no nó, sem passar por compressão JPEG, potencialmente preservando detalhes discriminativos perdidos na transmissão de imagens comprimidas.

O custo dessa abordagem é a maior demanda computacional nos nós, que passam a executar dois modelos de inferência (YOLO + OSNet) simultaneamente, aumentando o consumo de CPU e memória. A análise quantitativa desse *trade-off* é apresentada no Capítulo 4.

2.4.2 Trabalhos relacionados

A literatura relacionada ao presente trabalho pode ser organizada em três eixos temáticos: (i) sistemas multi-câmera para vigilância e rastreamento, (ii) aplicações de *edge computing* em visão computacional e (iii) modelos de extração de características para ReID. A seguir, os principais trabalhos são contextualizados em relação a cada eixo e ao sistema proposto.

2.4.2.1 Sistemas multi-câmera para vigilância e rastreamento

Khan e Shah (2006) propuseram uma abordagem multi-câmera para rastreamento de pessoas em multidões utilizando homografia planar, permitindo a correspondência de posições entre câmeras por meio de transformações geométricas (KHAN; SHAH, 2006). Embora pioneiro na fusão de informações espaciais entre câmeras, o método requer calibração geométrica precisa e não utiliza representações baseadas em aprendizado profundo.

Fei e Han (2023) apresentaram uma revisão abrangente de técnicas de rastreamento multi-objeto e multi-câmera (MTMC) baseadas em *deep learning* para transporte inteligente (FEI; HAN, 2023). Os autores identificam a reidentificação de pessoas como componente central dos sistemas MTMC e destacam a tendência de integração entre detecção, rastreamento e ReID em *pipelines* unificados.

Zheng et al. (2016) estabeleceram o protocolo padrão de avaliação para ReID, introduzindo o *dataset* Market-1501 e formalizando as métricas CMC e mAP como referências da área (ZHENG et al., 2016). O protocolo *probe* × *gallery* proposto por esses autores é o mesmo adotado neste trabalho.

2.4.2.2 Edge computing em visão computacional

Liu et al. (2020) investigaram a preservação de privacidade em sistemas de vigilância por vídeo executados em dispositivos de borda, demonstrando que o processamento local reduz a exposição de dados visuais sensíveis ao evitar a transmissão de *frames* brutos pela rede (LIU

et al., 2020). Esse princípio é diretamente aplicado na arquitetura *Edge-First* proposta neste trabalho.

Shi et al. (2016) formalizaram o paradigma de *edge computing*, propondo a migração de tarefas computacionais para dispositivos próximos à fonte de dados como estratégia para reduzir latência e consumo de banda (SHI et al., 2016). Satyanarayanan (2017) estendeu essa discussão ao conceito de *cloudlet*, um servidor intermediário entre a borda e a nuvem, relevante para cenários onde os dispositivos de borda possuem recursos insuficientes para processar modelos de inferência completos (SATYANARAYANAN, 2017).

2.4.2.3 Modelos de extração de características para ReID

Zhou et al. (2019) propuseram o OSNet, utilizado neste trabalho, demonstrando que a combinação de fluxos com campos receptivos distintos em blocos *Omni-Scale Residual* permite capturar características discriminativas em múltiplas escalas, com apenas 2,2 milhões de parâmetros (ZHOU et al., 2019). Em trabalho subsequente, Zhou et al. (2021) introduziram a técnica *Adaptive Instance Normalization* (AIN), que melhora a robustez a variações de domínio entre câmeras (ZHOU et al., 2021), conforme demonstrado na análise de separabilidade apresentada no Capítulo 4.

He et al. (2023) desenvolveram o *framework* FastReID, uma plataforma modular para experimentação com diferentes *backbones* e estratégias de treinamento para ReID (HE et al., 2023). Embora ofereça flexibilidade, a dependência de PyTorch para inferência limita sua portabilidade em dispositivos de borda, motivando a escolha do OSNet com ONNX Runtime neste trabalho.

2.4.2.4 Síntese comparativa

A Tabela 2.1 compara os trabalhos citados com o sistema proposto neste trabalho, considerando cinco dimensões: tipo de processamento (centralizado, borda ou híbrido), modelo de detecção, modelo de ReID, número de câmeras e avaliação de *trade-off* arquitetural.

Tabela 2.1 – Comparação entre trabalhos relacionados e o sistema proposto.

Trabalho	Processamento	Detecção	ReID	Câmeras	Comp. de arquiteturas
Khan e Shah (2006)	Centralizado	HOG + SVM	Homografia	4	Não
Zheng et al. (2016)	Centralizado	DPM	CNN genérica	6	Não
Zhou et al. (2019)	Centralizado	–	OSNet	–	Não
Liu et al. (2020)	Borda	CNN	–	–	Não
Fei e Han (2023)	Variável	YOLO/variantes	Variável	Múltiplas	Não
He et al. (2023)	Centralizado	–	FastReID	–	Não
Este trabalho	**Híbrido / Borda**	**YOLOv12n**	**OSNet**	**4**	**Sim**

Nota: "–" indica que o aspecto não é abordado ou não se aplica ao escopo do trabalho citado.

A principal contribuição diferenciadora deste trabalho em relação à literatura é a comparação direta e quantitativa entre duas arquiteturas de processamento (Híbrida e *Edge-First*) aplicadas à mesma tarefa de ReID, com avaliação simultânea de acurácia, latência, banda e uso de recursos computacionais. Os trabalhos existentes concentram-se em um único paradigma de processamento ou em um único aspecto do sistema (modelo de ReID, privacidade, ou rastreamento), sem avaliar o *trade-off* completo entre centralização e distribuição de carga.

Em ambientes industriais, plataformas como o NVIDIA DeepStream oferecem soluções para análise de vídeo em tempo real com fusão de dados multi-câmera, incluindo funcionalidades de triangulação de posição 3D e rastreamento global. Embora essas capacidades representem uma extensão natural do sistema proposto neste trabalho, sua implementação requer infraestrutura de GPU dedicada e calibração geométrica das câmeras, extrapolando o escopo da presente avaliação. A incorporação de triangulação multi-câmera é indicada como uma das direções para trabalhos futuros, conforme discutido no Capítulo 5.

2.5 Infraestrutura de Comunicação em Redes de Câmeras

A comunicação entre os nós de borda e o servidor central é um aspecto crítico do sistema, influenciando diretamente a latência, a largura de banda consumida e a confiabilidade da operação. Nesta seção são revisados os protocolos disponíveis e justificada a escolha de sockets UDP com formato de pacote próprio para o sistema proposto.

2.5.1 Protocolos de Transmissão de Vídeo

Em sistemas de vigilância tradicionais, a transmissão de vídeo entre câmeras e servidor central é comumente realizada por protocolos dedicados de *streaming*:

a) **RTSP** (*Real Time Streaming Protocol*): protocolo amplamente adotado em câmeras IP, que opera sobre RTP (*Real-time Transport Protocol*) para a entrega dos dados de mídia. Oferece controle de sessão (play, pause, teardown) e suporte a múltiplos codecs de vídeo. Contudo, a pilha RTSP/RTP introduz *overhead* de sinalização e requer manutenção de estado de sessão no servidor.

b) **WebRTC** (*Web Real-Time Communication*): projetado para comunicação ponto a ponto em navegadores, com suporte nativo a codificação adaptativa e travessia de NAT (*Network Address Translation*). A complexidade de configuração (ICE, STUN, TURN) e o foco em interatividade bidirecional o tornam excessivo para o cenário unidirecional de vigilância.

c) **SRT** (*Secure Reliable Transport*): protocolo de código aberto focado em transmissão de vídeo com baixa latência sobre redes instáveis, com correção de erros (ARQ). Embora adequado para contribuição de vídeo profissional, introduz complexidade de implementação nos nós de borda.

Neste trabalho, nenhum desses protocolos foi adotado porque o sistema não transmite fluxo de vídeo contínuo. Em ambas as arquiteturas, os *frames* são processados localmente nos nós e apenas os produtos da detecção, recortes JPEG ou vetores de *embedding*, são transmitidos como mensagens individuais. Essa característica torna desnecessária a infraestrutura de *streaming*, favorecendo protocolos de mensagens mais leves.

2.5.2 Protocolos de Mensageria IoT

Para comunicação entre dispositivos de borda e servidores em cenários de Internet das Coisas (IoT) e microserviços, diversas alternativas estão disponíveis:

a) ***Message Queuing Telemetry Transport*** **(MQTT)**: protocolo *publish-subscribe* projetado para redes com largura de banda limitada e dispositivos com poucos recursos. Oferece qualidade de serviço configurável (QoS 0, 1 ou 2) e desacoplamento entre produtores e consumidores. Contudo, depende de um *broker* intermediário e opera sobre TCP, adicionando latência de conexão.

b) **gRPC** (*Google Remote Procedure Call*): *framework* de RPC (*Remote Procedure Call*) baseado em HTTP/2 e *Protocol Buffers* (Protobuf) para serialização eficiente. Oferece tipagem forte, geração automática de código e suporte a *streaming* bidirecional.

A dependência de HTTP/2 e a complexidade de configuração tornam-no menos adequado para dispositivos de borda com restrições de recursos.

Embora ambos ofereçam vantagens em termos de estruturação e confiabilidade de mensagens, a introdução de camadas intermediárias (*broker* MQTT ou servidor HTTP/2) adicionaria latência e complexidade de implantação incompatíveis com os requisitos de simplicidade e portabilidade do sistema proposto. A escolha recaiu, portanto, sobre sockets nativos, conforme descrito na subseção seguinte.

2.5.3 Sockets UDP e TCP

O sistema implementado utiliza comunicação direta via *sockets*, a interface de mais baixo nível disponível para comunicação em rede, operando com dois protocolos da camada de transporte:

a) **UDP**: protocolo sem conexão e sem garantia de entrega, ordenação ou controle de fluxo. Cada mensagem é um datagrama independente, transmitido com *overhead* mínimo (cabeçalho de 8 bytes). A ausência de controle de congestionamento e de *handshake* resulta em latência mínima e implementação simplificada.

b) **TCP**: protocolo orientado a conexão, com garantia de entrega, ordenação e controle de fluxo. Adequado para mensagens cuja perda comprometeria a integridade da operação.

No sistema proposto, o UDP é utilizado para a transmissão de detecções (recortes JPEG ou vetores de *embedding*), onde a perda eventual de um datagrama equivale à perda de uma detecção individual, evento tolerável, dado que o mesmo indivíduo será detectado novamente em *frames* subsequentes. O sinal de encerramento (`done`), por sua vez, é enviado via UDP com confirmação implícita pelo protocolo da aplicação, garantindo que o servidor identifique corretamente o término do processamento de cada nó.

A escolha do UDP justifica-se pelo *trade-off* favorável entre latência e confiabilidade no cenário de vigilância: a prioridade é minimizar o atraso entre a detecção e o *matching*, mesmo que detecções esporádicas sejam perdidas. Em todos os experimentos realizados, a taxa de perda de pacotes na rede local foi de 0,0%, validando a viabilidade dessa abordagem no ambiente de avaliação.

2.5.4 Privacidade e transmissão de dados visuais

Um aspecto relevante na escolha da arquitetura de comunicação é o impacto sobre a privacidade dos indivíduos monitorados. Em sistemas que transmitem imagens, sejam *frames* completos ou recortes de detecções, a interceptação do tráfego de rede permite a reconstrução visual dos indivíduos, configurando uma exposição significativa de dados pessoais (LIU et al., 2020).

A arquitetura Híbrida, ao transmitir recortes JPEG das detecções, está sujeita a esse risco: cada datagrama contém uma imagem reconhecível de uma pessoa. A arquitetura *Edge-First*, por outro lado, transmite apenas o vetor de *embedding*, uma representação numérica de 512 dimensões que não permite a reconstrução visual do indivíduo. Essa diferença confere à abordagem *Edge-First* uma vantagem intrínseca em termos de privacidade, alinhando-se com princípios de minimização de dados presentes em legislações como a Lei Geral de Proteção de Dados (LGPD) (Lei Geral de Proteção de Dados, Lei nº 13.709/2018).

É importante ressaltar que, embora os *embeddings* não permitam a reconstrução direta da imagem, eles ainda constituem dados biométricos passíveis de identificar indivíduos por comparação. A transmissão segura desses dados, por meio de criptografia de transporte (como DTLS sobre UDP), é uma extensão recomendada para implantações em ambientes de produção, embora não implementada no escopo deste trabalho.

3 DESENVOLVIMENTO E METODOLOGIA EXPERIMENTAL

Este capítulo descreve o projeto, a implementação e a metodologia experimental do sistema de ReID em múltiplas câmeras. A Seção 3.1 apresenta a visão geral das duas arquiteturas comparadas. A Seção 3.2 detalha os componentes de software. A Seção 3.3 especifica os *datasets*, parâmetros e protocolos de avaliação utilizados nos experimentos.

3.1 Arquitetura e Visão Geral do Sistema

O sistema proposto é composto por nós de câmera e um servidor central, comunicando-se por meio de datagramas UDP em rede local. Cada nó processa um fluxo de vídeo independente e transmite dados ao servidor, que mantém uma galeria global de identidades e atribui um identificador único a cada pessoa detectada.

É importante destacar que o sistema constitui um protótipo funcional de software, executado integralmente em um único computador. Os nós de câmera e o servidor operam como processos Python independentes, comunicando-se via *sockets* UDP sobre a interface de *loopback* (`localhost`). Os fluxos de vídeo são provenientes de gravações pré-existentes dos *datasets* utilizados, não de câmeras físicas em tempo real. Essa configuração permite avaliar o comportamento do sistema, incluindo a comunicação via rede e o processamento distribuído, em condições controladas e reprodutíveis, sem a necessidade de infraestrutura de hardware embarcado ou redes sem fio reais. A delimitação completa do escopo experimental é apresentada na Seção 1.4.

Ambas as arquiteturas descritas a seguir foram integralmente implementadas como sistemas funcionais completos: cada uma possui seu próprio módulo de nó (`node.py`) e servidor (`server.py`), executados como processos independentes com comunicação real via *sockets* UDP. Não se trata de simulações ou modelos teóricos, mas de implementações operacionais que processam vídeo, transmitem dados pela rede e produzem resultados de reidentificação de ponta a ponta.

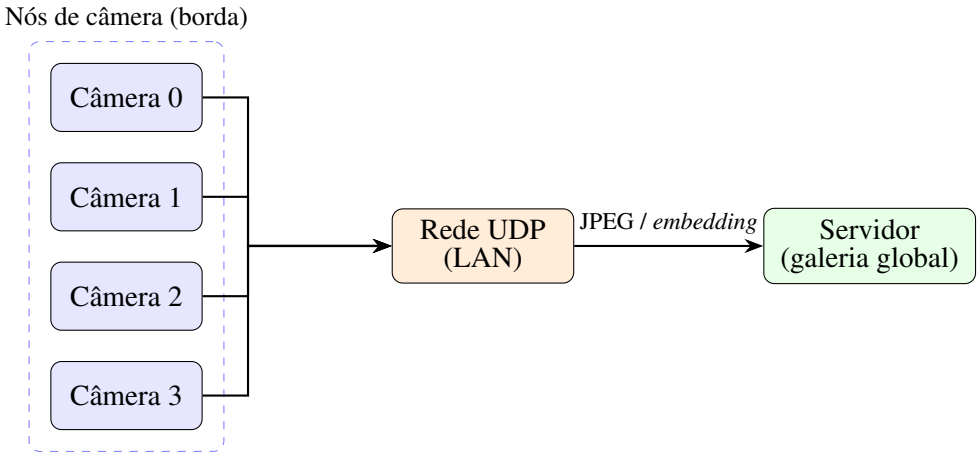
Duas arquiteturas foram implementadas e comparadas, diferindo na distribuição de carga entre nó e servidor:

1. **Arquitetura Híbrida**: o nó executa apenas a detecção de pessoas (YOLO) e transmite o recorte da imagem (*crop* JPEG) ao servidor, que realiza a extração de *embeddings* (OSNet)

e o *matching* contra a galeria.

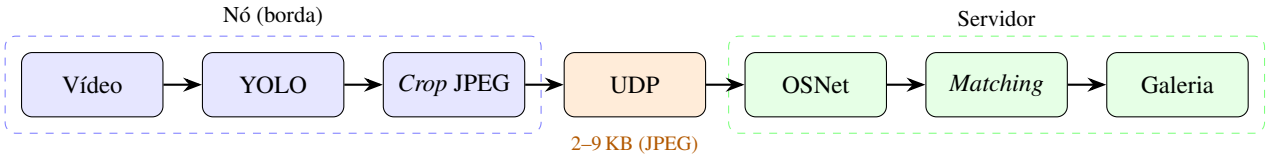
2. **Arquitetura *Edge-First***: o nó executa tanto a detecção quanto a extração de *embeddings* localmente, transmitindo apenas o vetor de características (2.048 bytes fixos) ao servidor, que realiza somente o *matching*.

Figura 3.1 – Visão geral do sistema multi-câmera: quatro nós de câmera transmitem dados via UDP a um servidor central, que mantém a galeria global de identidades.



Fonte: Elaborado pelo autor.

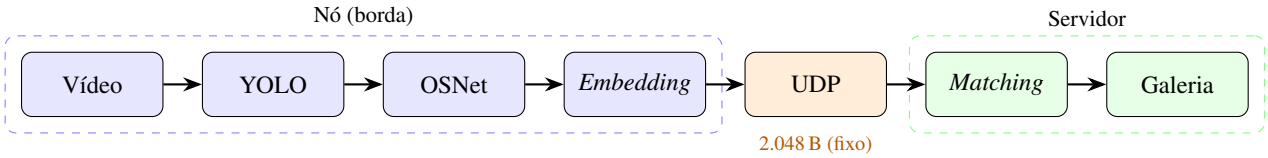
Figura 3.2 – Pipeline de processamento da arquitetura Híbrida.



Fonte: Elaborado pelo autor.

Na arquitetura Híbrida (Figura 3.2), o nó captura quadros do vídeo, executa o detector YOLO para localizar pessoas e codifica cada recorte como imagem JPEG. Os *crops* são enviados via UDP ao servidor, que decodifica a imagem, extrai um vetor de *embedding* de 512 dimensões por meio da rede OSNet e compara esse vetor com a galeria de identidades já registradas. O tamanho de cada pacote transmitido é variável, tipicamente entre 2 e 9 kilobytes, dependendo das dimensões da pessoa detectada e da qualidade de compressão JPEG.

Figura 3.3 – Pipeline de processamento da arquitetura *Edge-First*.



Fonte: Elaborado pelo autor.

Na arquitetura *Edge-First* (Figura 3.3), o nó executa tanto a detecção YOLO quanto a extração de *embeddings* OSNet localmente. O dado transmitido ao servidor é exclusivamente o vetor de características: 512 valores em ponto flutuante de 32 bits, totalizando exatamente 2.048 bytes por detecção. O servidor recebe o vetor e realiza apenas a comparação vetorial contra a galeria, sem necessidade de decodificar imagens ou executar redes neurais.

A Tabela 3.1 resume as diferenças-chave entre as duas arquiteturas.

Tabela 3.1 – Comparação estrutural entre as arquiteturas Híbrida e *Edge-First*.

Aspecto	**Híbrida**	***Edge-First***
Processamento na câmera	YOLO	YOLO + OSNet
Serviços no servidor	OSNet + *matching*	Somente *matching*
Dado transmitido	*Crop* JPEG (variável)	*Embedding* float32 (fixo)
Tamanho por detecção	2–9 KB	2.048 bytes
Privacidade do dado transmitido	Imagem da pessoa	Vetor numérico

3.2 Implementação dos Componentes

O sistema foi implementado em Python, utilizando bibliotecas consolidadas para inferência de redes neurais e processamento de imagens. Todos os componentes são compartilhados entre as duas arquiteturas, diferindo apenas na distribuição das etapas entre nó e servidor. O código-fonte está organizado nos diretórios `projeto/hybrid/` e `projeto/edge_first/`, cada um contendo os módulos `node.py` (nó de câmera) e `server.py` (servidor central), além de arquivos de configuração `config.yaml`, e está disponível publicamente no repositório[1].

3.2.1 Detecção de Pessoas com YOLOv12n

A detecção de pessoas é realizada pelo modelo YOLOv12n (*nano*), carregado pela biblioteca *Ultralytics*. O modelo opera em estágio único: recebe um quadro completo e produz,

[1] <https://github.com/HeltonPojo/multi-camera-person-reid-over-wireless-network>

em uma única passagem, as coordenadas de *bounding boxes* e os respectivos escores de confiança para cada pessoa detectada.

Os seguintes parâmetros controlam o comportamento da detecção:

a) **Limiar de confiança** (`model.conf`): fixado em 0,7. Detecções com escore inferior são descartadas.

b) **Frequência de quadros** (`model.frame_freq`): parâmetro disponível na configuração, porém mantido desativado nos experimentos (todos os quadros foram submetidos ao detector).

c) **Classe de interesse**: apenas a classe "pessoa" (classe 0 do COCO) é considerada.

Para cada pessoa detectada, o nó extrai o recorte (*crop*) correspondente da imagem original. Na arquitetura Híbrida, esse recorte é codificado como JPEG e enfileirado para transmissão. Na arquitetura *Edge-First*, o recorte é passado diretamente ao extrator de *embeddings* no próprio nó.

3.2.2 Extração de *Embeddings* com OSNet

A extração de vetores de características utiliza a rede OSNet, proposta por (ZHOU et al., 2019) para a tarefa de ReID. A arquitetura OSNet foi projetada para capturar simultaneamente padrões em múltiplas escalas espaciais, permitindo a geração de representações discriminativas mesmo em cenários com variação significativa de pose, escala e oclusão parcial.

O modelo utilizado é a variante `osnet_x1_0`, convertido para o formato ONNX. A inferência é executada por meio do *ONNX Runtime*, que permite a utilização do mesmo arquivo de modelo tanto nos nós de borda quanto no servidor, independentemente da plataforma ou acelerador disponível. Essa portabilidade foi um dos critérios determinantes na escolha do OSNet em detrimento de alternativas como FastReID(HE et al., 2023) ou PCB(SUN et al., 2018), que dependem de *frameworks* mais pesados para inferência.

A saída do modelo é um vetor de 512 dimensões em ponto flutuante de 32 bits, denominado *embedding*. Esse vetor constitui uma representação compacta da aparência visual da pessoa detectada em um hiperespaço métrico, no qual indivíduos visualmente similares são mapeados para regiões próximas.

Na arquitetura Híbrida, a extração ocorre no servidor, após a decodificação do *crop* JPEG recebido do nó. Na arquitetura *Edge-First*, a extração ocorre no próprio nó, imediatamente após

a detecção. Essa diferença implica que, na *Edge-First*, o *embedding* é computado a partir do recorte original da imagem, sem a degradação introduzida pela compressão e descompressão JPEG.

3.2.3 Galeria de Identidades e *Matching* Vetorizado

O servidor mantém uma galeria em memória contendo os *embeddings* acumulados para cada pessoa identificada. A estrutura de dados associa um identificador numérico (`id_0`, `id_1`, ...) a um conjunto de até 64 vetores de 512 dimensões, representando observações distintas da mesma pessoa ao longo do tempo.

3.2.3.1 Algoritmo de *Matching*

Para cada novo *embedding* recebido, o servidor executa o seguinte procedimento:

1. **Cálculo do centroide**: para cada identidade na galeria, computa-se o vetor médio (*mean embedding*) de todos os vetores armazenados para aquela pessoa.

2. **Distância cosseno vetorizada**: os centroides de todas as identidades são organizados em uma matriz $G \in \mathbb{R}^{n \times 512}$, onde n é o número de pessoas na galeria. A distância cosseno entre o *embedding* de entrada $\mathbf{e}$ e cada centroide é calculada de forma vetorizada:

$$d_i = 1 - \frac{G_i \cdot \mathbf{e}}{\|G_i\| \cdot \|\mathbf{e}\|} \tag{3.1}$$

   onde G_i é o centroide da i-ésima identidade. O produto $G \cdot \mathbf{e}$ é computado como uma única operação de produto matricial utilizando a biblioteca NumPy, resultando em complexidade $O(n)$ para a comparação com todas as identidades.

3. **Decisão**: se a menor distância $d_{\min} = \min_i d_i$ for inferior ao limiar de similaridade (*similarity threshold*), o *embedding* é atribuído à identidade correspondente. Caso contrário, uma nova identidade é criada.

3.2.3.2 Gerenciamento da Galeria

Duas políticas de evicção FIFO previnem o crescimento ilimitado da galeria:

a) **Por pessoa**: cada identidade armazena no máximo `max_gallery_per_person` = 64 vetores. Ao atingir esse limite, o vetor mais antigo é substituído pelo novo,

permitindo que a representação da pessoa se atualize conforme novas observações são recebidas.

b) **Global**: o número total de identidades na galeria é limitado a `max_gallery_size` = 500. Quando esse limite é atingido, a identidade mais antiga (primeira registrada) é removida integralmente. Essa política evita a saturação de memória em cenários de longa duração, nos quais erros de *matching* podem gerar identidades espúrias.

O processamento de *matching* é paralelizado por meio de duas *threads* que consomem pacotes de uma fila compartilhada, aumentando a vazão do servidor.

3.2.4 Protocolo de Comunicação Nó–Servidor

A comunicação entre nós e servidor utiliza datagramas UDP, priorizando baixa latência em detrimento de entrega garantida. Cada detecção é transmitida como um único datagrama, cujo formato é descrito na Figura 3.4.

Figura 3.4 – Formato do datagrama UDP transmitido pelo nó ao servidor.

Identificador: `"{nome}:{quadro}"` (preenchido com nulos)	0–31 32 B
Tamanho do *payload* (dígitos ASCII, zero à esquerda)	32–35 4 B
x_1 (int32 com sinal)	36–51 16 B
y_1 (int32 com sinal)	
x_2 (int32 com sinal)	
y_2 (int32 com sinal)	
Payload: JPEG (Híbrida) ou *embedding* float32 (*Edge-First*) ...	52+ variável

Nota: Na arquitetura Híbrida, o *payload* é o recorte JPEG da pessoa detectada (2–9 KB). Na *Edge-First*, é o vetor de *embedding* de 512 dimensões (2.048 bytes fixos). O cabeçalho de 52 bytes é idêntico em ambas as arquiteturas.

O sinal de encerramento é transmitido como um datagrama especial com tamanho de *payload* igual a zero (`"0000"`) e identificador no formato `"nome:done"`. Ao receber sinais de encerramento de todos os nós esperados, o servidor drena a fila de processamento, grava os resultados em disco e encerra automaticamente.

A escolha por UDP em detrimento de TCP justifica-se pelo cenário de aplicação: em um sistema de vigilância com múltiplas câmeras transmitindo detecções simultaneamente, a retransmissão automática do TCP introduziria latência adicional e potencialmente agravaria o congestionamento. A perda eventual de um datagrama representa a não-identificação de uma única detecção em um único quadro, impacto aceitável dado que a mesma pessoa será detectada novamente em quadros subsequentes. Nos experimentos realizados, a taxa de perda observada foi de 0,0% em todos os cenários, confirmando a viabilidade do UDP em rede local.

3.3 Metodologia Experimental

Esta seção descreve os elementos que compõem o protocolo experimental adotado para comparar as arquiteturas Híbrida e *Edge-First*. A Subseção 3.3.1 apresenta os *datasets* públicos utilizados e suas características distintivas. A Subseção 3.3.2 especifica os parâmetros fixos e variáveis dos experimentos. A Subseção 3.3.4 detalha as métricas e o protocolo de avaliação de acurácia adotado para cada *dataset*. Por fim, a Subseção 3.3.3 caracteriza o ambiente computacional no qual todos os experimentos foram conduzidos.

3.3.1 *Datasets* Utilizados

Dois *datasets* públicos foram selecionados para a avaliação do sistema, cada um exercitando aspectos distintos do *pipeline*:

EPFL Terrace (FLEURET et al., 2008): *dataset* principal, composto por vídeos de quatro câmeras sincronizadas que monitoram um terraço com até sete pedestres. As câmeras apresentam sobreposição parcial de campo de visão, permitindo a avaliação *cross-câmera* do sistema completo, desde a detecção YOLO até o *matching* no servidor. O *ground truth* é fornecido como trajetórias anotadas por câmera, incluindo identificador de *track*, coordenadas de *bounding box* e número do quadro.

PRID2011 (HIRZER et al., 2011): *dataset* de validação cruzada, composto por recortes pré-extraídos de 200 pessoas fotografadas por duas câmeras, com 549 identidades adicionais presentes apenas na câmera B (*distractors*). Por utilizar recortes prontos, a avaliação contorna a etapa de detecção YOLO e mede exclusivamente a qualidade da extração de *embeddings* e do *matching* contra uma galeria com *distractors*, protocolo amplamente adotado na literatura de ReID.

Os dois *datasets* diferem substancialmente em suas características visuais, o que impacta diretamente a dificuldade da tarefa de reidentificação:

a) **Ângulo de câmera.** No EPFL Terrace, as quatro câmeras estão posicionadas em um mesmo terraço, com ângulos de visão relativamente próximos e sobreposição parcial de campo. No PRID2011, as duas câmeras capturam cenas em locais distintos de um campus, com ângulos frontais e laterais que produzem variações significativas de pose e silhueta para a mesma pessoa.

b) **Iluminação.** O EPFL Terrace apresenta iluminação natural uniforme em uma cena externa, com condições estáveis ao longo da sequência. O PRID2011 exibe diferenças marcantes de iluminação entre as câmeras, incluindo variações de contraste, sombras e temperatura de cor que alteram a aparência das roupas e da pele.

c) **Resolução e qualidade.** Os recortes do EPFL Terrace, extraídos de vídeo contínuo a 25 fps, apresentam resolução moderada e consistente entre câmeras. No PRID2011, a resolução dos recortes varia entre câmeras e entre pessoas, com imagens frequentemente de baixa resolução e com artefatos de compressão.

Essas diferenças tornam o PRID2011 um cenário significativamente mais desafiador para modelos de extração de *embeddings*, conforme discutido na análise de separabilidade apresentada na Subseção 4.3.6.

Figura 3.5 – Exemplos comparativos dos *datasets* utilizados: EPFL Terrace (esquerda) e PRID2011 (direita), evidenciando as diferenças de ângulo, iluminação e resolução.

(a) EPFL Terrace — cena com múltiplas pessoas, iluminação uniforme.

(b) PRID2011 — mesma pessoa pelas câmeras A e B, com variações de iluminação e pose.

Fonte: Adaptado dos *datasets* originais (FLEURET et al., 2008; HIRZER et al., 2011).

A Tabela 3.2 apresenta a comparação entre os *datasets* considerados.

Tabela 3.2 – Comparação dos *datasets* considerados para avaliação.

Dataset	**Câmeras**	**Pessoas**	**Formato**	**GT ReID**	**Acesso**	**Pipeline**
EPFL Terrace	4	até 7	Vídeo AVI	Sim	Livre	Completo
PRID2011	2	200 + 549	Crops PNG	Sim	Livre	Parcial
iLIDS-VID	2	300	Seq. PNG	Sim	Restrito	Parcial
MARS	6	1.261	Tracklets	Sim	Livre	Completo
Market-1501	6	1.501	Crops PNG	Sim	Livre	Parcial

Nota: “Pipeline completo” indica que o *dataset* permite avaliar detecção + transmissão + ReID. “Parcial” indica que apenas o *matching* isolado é avaliável.

3.3.2 Parâmetros dos Experimentos

Para garantir uma comparação justa entre as arquiteturas, todos os parâmetros de modelo e *matching* foram mantidos idênticos nos dois `config.yaml`, variando-se apenas o limiar de similaridade entre os experimentos. A Tabela 3.3 lista os parâmetros fixos.

Tabela 3.3 – Parâmetros fixos utilizados em todos os experimentos.

Parâmetro	Valor	Descrição
`model.conf`	0,7	Limiar de confiança YOLO
`model.frame_freq`	–	Parâmetro disponível, porém desativado nos experimentos (todos os quadros processados)
`max_gallery_per_person`	64	Embeddings por identidade
`max_gallery_size`	500	Identidades máximas na galeria
`socket_timeout`	2 s	Timeout de recepção UDP
Modelo de detecção	YOLOv12n	Arquivo `.pt` compartilhado
Modelo de extração	OSNet x1_0	Arquivo `.onnx` compartilhado

O parâmetro variável entre os experimentos é o limiar de similaridade (`similarity_threshold`), que controla a distância cosseno máxima para que um *embedding* seja associado a uma identidade existente. Valores menores tornam o *matching* mais conservador (exigem maior similaridade), enquanto valores maiores são mais permissivos. A grade de limiares avaliada foi: 0,20 / 0,30 / 0,35 / 0,40, totalizando quatro experimentos por arquitetura e por *dataset*, 16 execuções ao todo.

3.3.3 Ambiente Computacional

Todos os experimentos foram executados em um único computador de uso geral, sem aceleração de hardware dedicada para inferência de redes neurais. As especificações da máquina são apresentadas na Tabela 3.4.

Tabela 3.4 – Especificações do ambiente computacional utilizado nos experimentos.

Componente	Especificação
Modelo	MacBook Pro (Apple Silicon)
Processador	Apple M1 — 8 núcleos (4 de desempenho + 4 de eficiência)
Memória RAM	16 GB (unificada, compartilhada entre CPU e GPU integrada)
Sistema operacional	macOS 26.3 (Sequoia)
Aceleração de inferência	Nenhuma dedicada (ONNX Runtime CPU)

A ausência de unidade de processamento gráfico dedicada (GPU discreta) ou processador neural (NPU) externo é um fator relevante para a interpretação dos resultados de desempenho apresentados no Capítulo 4. Em particular, as métricas de uso de CPU e tempo de processamento por detecção refletem o custo de inferência exclusivamente em software sobre os núcleos ARM do M1. Em dispositivos equipados com acelerador dedicado, como o NVIDIA Jetson Nano ou

módulos com Hailo-8, a carga computacional seria distribuída de forma diferente, potencialmente reduzindo o tempo de extração de *embeddings* na câmera e o consumo dos núcleos de propósito geral.

3.3.4 Protocolo de Avaliação de Acurácia

As métricas de acurácia de reidentificação seguem o padrão da literatura: Rank-1, Rank-5 e mAP.

3.3.4.1 EPFL Terrace

Para o EPFL Terrace, a avaliação segue o protocolo de correspondência *cross-câmera*:

1. Para cada câmera, as detecções do sistema são associadas ao *ground truth* por sobreposição espacial, utilizando *Intersection over Union* (IoU) com limiar de 0,5.
2. Para cada *track* do GT associado com sucesso, identifica-se o identificador do sistema (PID) mais frequentemente atribuído, o PID dominante.
3. Para cada par ordenado de câmeras (A, B) e cada identidade GT presente em ambas, o PID dominante em A é utilizado como *query*. A galeria é formada pelos *tracks* de B, ranqueados pela contagem de ocorrências do PID de *query*.
4. São calculados Rank-1 (identidade correta no topo do ranking), Rank-5 (identidade correta entre os cinco primeiros) e AP (precisão média) para cada par, com média sobre todos os 108 pares *cross-câmera* avaliados.

3.3.4.2 PRID2011

Para o PRID2011, segue-se o protocolo padrão A$\rightarrow$B da literatura:

1. *Probe*: 200 identidades da câmera A (IDs 1 a 200).
2. *Gallery*: 749 identidades da câmera B (200 com correspondência + 549 *distractors*).
3. Para cada *probe*, identifica-se o PID dominante em A e ranqueia-se a *gallery* de B pela contagem de ocorrências desse PID.
4. Rank-1, Rank-5 e mAP são calculados sobre as 200 consultas.

3.3.5 Protocolo de Avaliação de Desempenho

As métricas de desempenho do sistema são coletadas por meio de linhas de *log* instrumentadas (`PERF`) registradas pelo servidor e pelos nós durante a execução:

a) **Latência de ReID**: tempo em milissegundos entre a recepção do datagrama pelo servidor e a conclusão do *matching*. São reportadas a média e o percentil 95 (p95) por câmera.

b) **Consumo de banda**: tamanho em bytes de cada datagrama recebido, acumulado por câmera. Permite calcular a banda média por detecção e o total transmitido.

c) **Perda de frames**: razão entre datagramas recebidos e efetivamente processados pelo servidor.

d) **Utilização de CPU e memória do nó**: amostrados a cada 50 quadros processados. São reportados o uso médio de CPU (% em relação a um núcleo) e o pico de memória residente.

Os dados são extraídos automaticamente pelo *script* `evaluate_performance.py`, que correlaciona os *logs* do servidor e dos nós pelo *timestamp* da sessão e gera arquivos CSV consolidados para cada experimento.

4 RESULTADOS E ANÁLISE

Neste capítulo são apresentados e discutidos os resultados obtidos nos experimentos descritos no Capítulo 3. A análise está organizada em três eixos: acurácia de reidentificação (Seção 4.1), desempenho do sistema (Seção 4.2) e discussão integrada dos resultados (Seção 4.3).

4.1 Acurácia de Reidentificação

A acurácia de reidentificação foi avaliada por meio das métricas Rank-1, Rank-5 e mAP, conforme o protocolo descrito na Subseção 3.3.4. Os resultados são apresentados em duas dimensões: a sensibilidade ao limiar de similaridade em cada *dataset* (Tabelas 4.1 e 4.2) e a comparação direta entre as arquiteturas nos limiares ótimos selecionados (Tabela 4.3).

Tabela 4.1 – Sensibilidade ao limiar de similaridade na acurácia de ReID — EPFL Terrace (108 pares *cross-câmera*)

Limiar	**Híbrida**			**Edge-First**		
	Rank-1	**Rank-5**	**mAP**	**Rank-1**	**Rank-5**	**mAP**
0,20	43,5%	70,4%	57,0%	50,9%	75,9%	62,4%
0,30	56,5%	77,8%	68,5%	63,9%	85,2%	75,1%
0,35	**58,3%**	88,0%	**72,3%**	60,2%	89,8%	73,9%
0,40	56,5%	**92,6%**	71,8%	**67,6%**	**91,7%**	**79,9%**

4.1.1 Análise dos resultados no EPFL Terrace

A Tabela 4.1 apresenta a acurácia de ReID para ambas as arquiteturas no *dataset* EPFL Terrace, avaliada sobre 108 pares *cross-câmera*. Três observações principais emergem dos dados.

Em primeiro lugar, a arquitetura *Edge-First* supera a Híbrida em quase todas as combinações de métrica e limiar. O melhor resultado absoluto é obtido pela *Edge-First* com limiar 0,40: Rank-1 de 67,6%, Rank-5 de 91,7% e mAP de 79,9%. A Híbrida atinge seu pico de Rank-1 em 58,3% (limiar 0,35), ainda 9,3 pontos percentuais abaixo do melhor resultado *Edge-First*, obtido em outro limiar (0,40). Quando comparadas no mesmo limiar (0,40), a diferença é de 11,1 p.p. (67,6% vs. 56,5%).

Em segundo lugar, o Rank-5 apresenta comportamento monotonicamente crescente com o aumento do limiar em ambas as arquiteturas, atingindo 92,6% (Híbrida) e 91,7% (*Edge-First*) no limiar 0,40. Isso indica que, mesmo quando a identidade correta não ocupa a primeira posição

do *ranking*, ela está consistentemente entre as cinco primeiras, um resultado relevante para sistemas que apresentam candidatos para validação humana.

Em terceiro lugar, o comportamento do Rank-1 da *Edge-First* é não-monotônico: sobe de 50,9% (limiar 0,20) para 63,9% (limiar 0,30), recua para 60,2% (limiar 0,35) e atinge o máximo de 67,6% (limiar 0,40). Esse padrão reflete a interação entre o limiar e o comportamento da galeria: limiares intermediários podem criar situações em que duas observações distintas de uma mesma pessoa são registradas como identidades separadas, fragmentando o *cluster* e prejudicando o Rank-1. Limiares mais altos favorecem a fusão dessas observações, resultando em representações mais completas na galeria.

Com base nesses resultados, o limiar 0,40 foi selecionado como configuração ótima para o EPFL Terrace.

Tabela 4.2 – Sensibilidade ao limiar de similaridade na acurácia de ReID — PRID2011 (200 *probe* / 749 *gallery*, protocolo A→B, *tie-breaking* pessimista)

Limiar	**Híbrida**			**Edge-First**		
	Rank-1	**Rank-5**	**mAP**	**Rank-1**	**Rank-5**	**mAP**
0,20	1,0%	1,5%	1,4%	0,5%	1,0%	0,9%
0,30	**2,5%**	**4,0%**	**3,2%**	**2,0%**	**3,5%**	**2,7%**
0,35	0,5%	3,0%	1,4%	1,0%	2,0%	1,8%
0,40	1,5%	2,0%	2,1%	0,5%	0,5%	1,0%

Nota: *Tie-breaking* pessimista: em empates de *score*, a identidade correta é posicionada após as incorretas, seguindo o protocolo conservador da literatura.

4.1.2 Análise dos resultados no PRID2011

A Tabela 4.2 revela que o sistema não consegue realizar reidentificação efetiva no PRID2011 com o protocolo de *tie-breaking* pessimista, padrão na literatura de ReID. O melhor Rank-1 obtido foi de 2,5% (Híbrida, limiar 0,30), valor próximo ao acaso para uma galeria de 749 identidades (acaso $\approx 0{,}13\%$). Embora ligeiramente acima do aleatório, esse desempenho indica que o sistema atribui corretamente a identidade *cross-câmera* em apenas 5 das 200 *probes* avaliadas.

O Rank-5 apresenta valores marginalmente superiores (máximo de 4,0% para a Híbrida e 3,5% para a *Edge-First* no limiar 0,30), mas igualmente insuficientes para uso prático. O mAP segue o mesmo padrão, com máximo de 3,2%.

A causa fundamental desse insucesso reside na baixa capacidade de discriminação do

modelo `osnet_x1_0` no domínio visual do PRID2011, conforme analisado em detalhe na Subseção 4.3.6. O GAP de separabilidade de apenas 0,062 entre similaridades intra e inter-pessoa faz com que a maioria das *probes* não consiga ser vinculada à identidade correta na galeria da outra câmera. Com a maioria das identidades na galeria obtendo *score* zero para o *pid* de consulta, o *ranking* é dominado por empates, e no protocolo pessimista a identidade correta é posicionada após as incorretas empatadas, resultando nos valores próximos de zero observados.

Apesar do desempenho insatisfatório em todos os limiares, o limiar 0,30 foi selecionado para a comparação entre arquiteturas por apresentar os melhores valores absolutos.

Tabela 4.3 – Comparação de acurácia de ReID entre as arquiteturas Híbrida e *Edge-First* nos dois *datasets* avaliados

Dataset	**Métrica**	**Híbrida**	***Edge-First***	Δ
EPFL Terrace (limiar 0,40)	Rank-1	56,5%	**67,6%**	+11,1 p.p.
	Rank-5	**92,6%**	91,7%	−0,9 p.p.
	mAP	71,8%	**79,9%**	+8,1 p.p.
PRID2011 (limiar 0,30)	Rank-1	**2,5%**	2,0%	−0,5 p.p.
	Rank-5	**4,0%**	3,5%	−0,5 p.p.
	mAP	**3,2%**	2,7%	−0,5 p.p.

Nota: p.p. = pontos percentuais. Δ = *Edge-First* − Híbrida.

4.1.3 Comparação entre arquiteturas

A Tabela 4.3 consolida a comparação de acurácia entre as duas arquiteturas, utilizando os limiares ótimos selecionados para cada *dataset*.

No EPFL Terrace (limiar 0,40), a *Edge-First* supera a Híbrida em Rank-1 por 11,1 pontos percentuais (67,6% vs. 56,5%) e em mAP por 8,1 p.p. (79,9% vs. 71,8%). A única métrica em que a Híbrida apresenta vantagem marginal é o Rank-5, com diferença de apenas 0,9 p.p. (92,6% vs. 91,7%), diferença dentro da margem de variabilidade esperada para 108 pares de avaliação.

No PRID2011 (limiar 0,30), ambas as arquiteturas apresentam desempenho próximo ao acaso: Rank-1 de 2,5% (Híbrida) e 2,0% (*Edge-First*), com mAP de 3,2% e 2,7%, respectivamente. A ligeira vantagem da Híbrida (+0,5 p.p.) não é significativa nesse patamar e está dentro da margem de variabilidade. O insucesso em ambas as arquiteturas confirma que a limitação não é arquitetural, mas do modelo de extração de características (`osnet_x1_0`), cuja baixa separabilidade no domínio visual do PRID2011 impede o *matching cross-câmera* efetivo (ver Subseção 4.3.6).

No EPFL Terrace, a *Edge-First* demonstra acurácia superior à Híbrida. A hipótese para

essa vantagem, a preservação da qualidade do *crop* ao extrair o *embedding* localmente sem compressão JPEG intermediária, é discutida na Seção 4.3. No PRID2011, essa diferença é irrelevante, pois ambas as arquiteturas processam as mesmas imagens pré-recortadas (modo *crops*) e o fator limitante é a capacidade discriminativa do modelo, não a arquitetura de processamento.

Para contextualização, o resultado de 67,6% no EPFL Terrace situa-se na faixa reportada pela literatura para métodos baseados em *deep learning*: Zheng et al. (2016) reportam que abordagens clássicas atingem Rank-1 entre 40–60% nos principais *benchmarks*, enquanto métodos baseados em CNNs alcançam 70–90% (ZHENG et al., 2016). O insucesso no PRID2011 evidencia a dependência do modelo em relação ao domínio visual, discutida na Subseção 4.3.7.

4.2 Desempenho do Sistema

A avaliação de desempenho complementa a análise de acurácia ao examinar os custos computacionais e de comunicação de cada arquitetura. São analisadas três dimensões: latência de ReID (Subseção 4.2.1), consumo de banda (Subseção 4.2.2) e uso de recursos computacionais nas câmeras (Subseção 4.2.3).

Tabela 4.4 – Latência de ReID em função do limiar de similaridade — EPFL Terrace (média das quatro câmeras)

Limiar	**Híbrida**		***Edge-First***	
	Lat. média $\pm$ IC 95%	**p95**	**Lat. média**	**p95**
0,20	$126 \pm 1{,}8$ s	$\approx$314 s	**4,6 ms**	**10,3 ms**
0,30	$115 \pm 2{,}1$ s	$\approx$388 s	**2,0 ms**	**3,8 ms**
0,35	$80 \pm 1{,}1$ s	$\approx$197 s	**1,4 ms**	**2,8 ms**
0,40	$110 \pm 1{,}9$ s	$\approx$328 s	**1,0 ms**	**1,5 ms**

Nota: Valores da Híbrida apresentados como média $\pm$ IC 95%. IC 95% da *Edge-First* inferior a 0,1 ms em todos os limiares. A latência na arquitetura Híbrida corresponde ao tempo de espera na fila do servidor (OSNet + *matching*); na *Edge-First*, corresponde apenas ao *matching* vetorial. Perda de frames: 0,0% em todos os experimentos.

Fonte: Elaborado pelo autor.

4.2.1 Latência de ReID

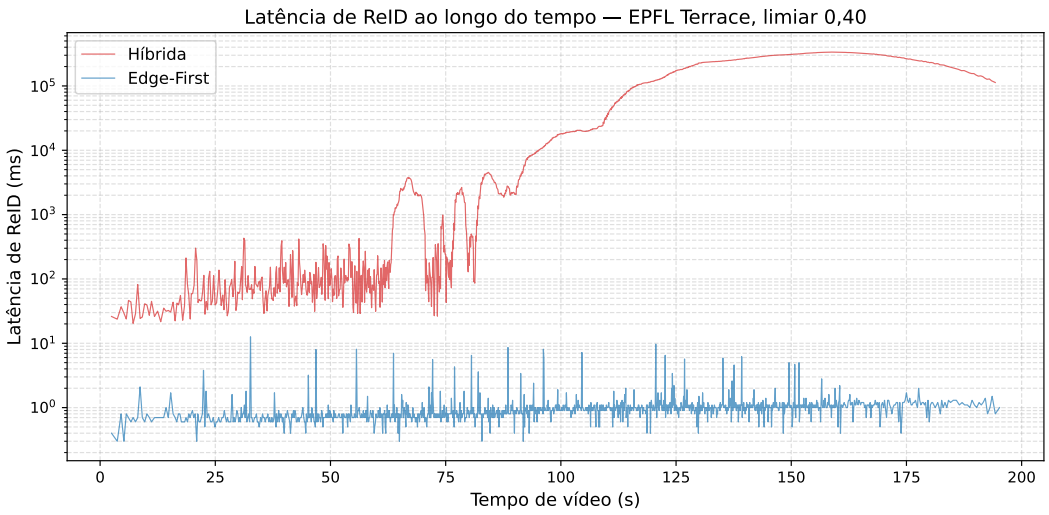
A Tabela 4.4 apresenta a latência média e o percentil 95 (p95) de ReID para cada limiar avaliado no EPFL Terrace. A diferença entre as arquiteturas é de cinco ordens de magnitude: enquanto a Híbrida apresenta latência média de aproximadamente 110 s (limiar 0,40), a *Edge-First* processa cada detecção em cerca de 1,0 ms.

Na arquitetura Híbrida, a latência de ReID corresponde ao tempo total desde o recebimento do *crop* JPEG pelo servidor até a conclusão do *matching*: extração do *embedding* via OSNet (aproximadamente 20 ms por *crop*) e comparação vetorizada com a galeria. Os valores elevados, com p95 de até 388 s no limiar 0,30, não refletem o tempo de inferência individual, mas sim o acúmulo na fila de processamento do servidor: quatro câmeras transmitem detecções simultaneamente a uma taxa superior à capacidade de processamento OSNet do servidor, gerando uma fila crescente. Esse fenômeno é analisado em detalhe na Subseção 4.3.2.

Na *Edge-First*, a latência corresponde exclusivamente ao *matching* vetorial (produto interno NumPy), uma vez que a extração de *embedding* já foi realizada na câmera. Os valores de 1,0 a 4,6 ms são consistentes com a operação de produto matricial entre um vetor de 512 dimensões e a galeria de identidades. Observa-se uma correlação com o limiar: limiares menores produzem galerias maiores (mais identidades distintas), resultando em matrizes maiores para o produto interno e, consequentemente, latência ligeiramente superior: 4,6 ms para limiar 0,20 vs. 1,0 ms para limiar 0,40.

A Figura 4.1 ilustra a evolução da latência ao longo do tempo de execução para ambas as arquiteturas no EPFL Terrace com limiar 0,40. A curva da Híbrida apresenta crescimento monotônico, confirmando o acúmulo progressivo na fila do servidor, enquanto a *Edge-First* mantém latência estável na faixa de milissegundos durante todo o experimento.

Figura 4.1 – Latência de ReID ao longo do tempo — EPFL Terrace, limiar 0,40.



Fonte: Elaborado pelo autor.

Tabela 4.5 – Detecções por câmera e taxa de transmissão — EPFL Terrace, limiar de similaridade 0,40

Câmera	**Híbrida**		***Edge-First***	
	Detecções	**Taxa (Kbps)**	**Detecções**	**Taxa (Kbps)**
Câmera 0	17.086	345	17.528	127
Câmera 1	19.253	326	19.517	138
Câmera 2	18.176	305	18.403	131
Câmera 3	14.920	319	16.260	120
Total	**69.435**	**≈324**	**71.708**	**≈130**

Nota: A diferença de 2.273 detecções entre as arquiteturas decorre de variações de temporização entre execuções independentes. Taxa variável na Híbrida reflete o tamanho do *crop* JPEG; taxa fixa na *Edge-First* reflete o vetor de *embedding* (512×4 bytes = 2.048 bytes por detecção). Perda de frames: 0,0% em todas as câmeras e arquiteturas.

Fonte: Elaborado pelo autor.

4.2.2 Consumo de banda

A Tabela 4.5 detalha o número de detecções e a taxa de transmissão por câmera para ambas as arquiteturas no EPFL Terrace (limiar 0,40). O número total de detecções é similar entre as arquiteturas: 69.435 (Híbrida) vs. 71.708 (*Edge-First*). A diferença de 2.273 detecções decorre de variações de temporização entre execuções independentes, confirmando que a etapa de detecção YOLO opera de forma consistente em ambas as configurações.

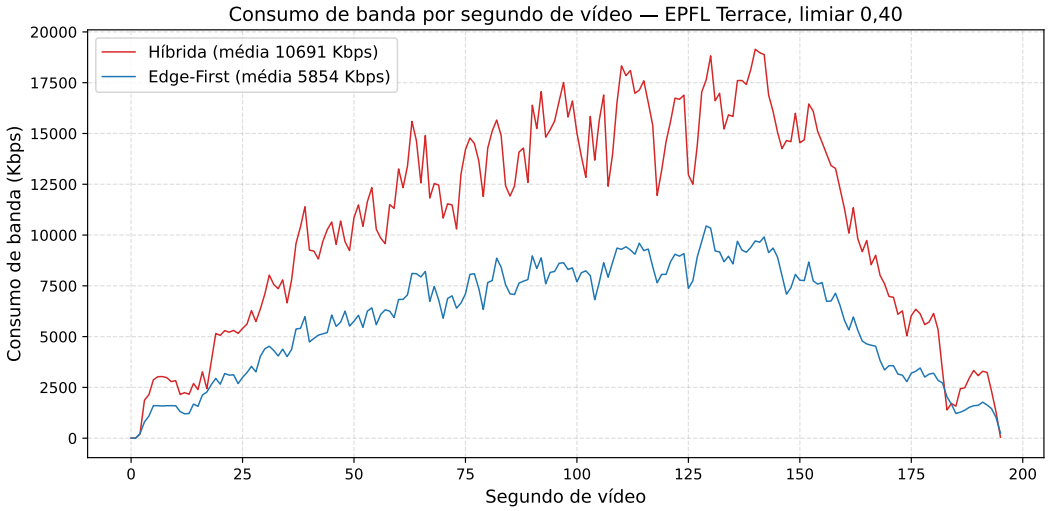
A diferença fundamental reside no tamanho do *payload* transmitido por detecção. Na Híbrida, cada *crop* JPEG tem tamanho variável, dependendo das dimensões da pessoa detectada e da qualidade de compressão. A banda média por detecção foi de aproximadamente 3.863 bytes, com variação de 3.510 B (Câmera 2) a 4.428 B (Câmera 3), pois a Câmera 3 captura pessoas mais próximas, resultando em *crops* maiores. Na *Edge-First*, cada detecção transmite exatamente 2.048 bytes (512 dimensões $\times$ 4 bytes por `float32`), independentemente do tamanho ou da posição da pessoa.

Essa diferença resulta em uma redução de 45,2% na banda total: 255,8 MB (Híbrida) vs. 140,1 MB (*Edge-First*). Em cenários com múltiplas câmeras transmitindo sobre redes sem fio, onde a largura de banda é compartilhada e sujeita a congestionamento, essa redução pode ser determinante para a viabilidade do sistema.

A Figura 4.2 apresenta a taxa de transmissão média (em KB/s) ao longo do tempo para cada arquitetura. A Híbrida exibe picos mais altos e maior variabilidade, reflexo direto da flutuação no tamanho dos *crops* JPEG: quando pessoas aparecem mais próximas da câmera, os

recortes são maiores e a taxa de transmissão aumenta proporcionalmente. A *Edge-First*, por sua vez, mantém uma taxa estável e previsível, limitada apenas pela frequência de detecções, já que cada *embedding* tem tamanho fixo de 2.048 bytes. Essa previsibilidade facilita o dimensionamento da rede e evita rajadas de tráfego que poderiam causar congestionamento em enlaces sem fio compartilhados.

Figura 4.2 – Consumo de banda por segundo de vídeo — EPFL Terrace, limiar 0,40.



Fonte: Elaborado pelo autor.

Tabela 4.6 – Comparação de desempenho do sistema no *dataset* EPFL Terrace (limiar de similaridade 0,40)

Métrica	**Híbrida**	***Edge-First***
Latência média de ReID	≈110 s	**1,0 ms**
Latência p95 de ReID	≈328 s	**1,5 ms**
Taxa de transmissão	≈324 Kbps (variável)	**≈130 Kbps (fixo)**
Banda total, 4 câmeras	255,8 MB	**140,1 MB**
CPU média na câmera	≈3,7 núcleos	≈6,9 núcleos
Memória de pico na câmera	≈616 MB	≈782 MB
Serviços no servidor	OSNet + *matching*	Somente *matching*

Nota: Perda de frames: 0,0% em todos os experimentos. A arquitetura Híbrida executa apenas YOLO nas câmeras e delega a extração de *embeddings* (OSNet) e o *matching* ao servidor. A *Edge-First* executa YOLO e OSNet nas câmeras, transmitindo apenas o vetor de *embedding* (512 dimensões, 2.048 bytes fixos) ao servidor, que realiza somente o *matching* vetorial.

4.2.3 Uso de recursos computacionais

A Tabela 4.6 consolida as métricas de desempenho para ambas as arquiteturas no EPFL Terrace com limiar 0,40. Enquanto a *Edge-First* apresenta vantagens em latência, banda e carga

no servidor, as câmeras arcam com maior demanda computacional.

O consumo de CPU nas câmeras da *Edge-First* é de aproximadamente 6,9 núcleos em média, contra 3,7 na Híbrida. Esse aumento de 85% reflete a execução simultânea de dois modelos de inferência (YOLO + OSNet) na câmera. A memória de pico segue padrão similar: 782 MB na *Edge-First* contra 616 MB na Híbrida, um aumento de 27%.

4.2.3.1 Ambiente de Execução dos Experimentos

O ambiente computacional no qual as medições foram realizadas está detalhado na Subseção 3.3.3 do Capítulo 3. Em síntese, trata-se de um MacBook Pro com processador Apple M1 (8 núcleos), 16 GB de memória RAM unificada e sistema operacional macOS 26.3, sem aceleração de hardware dedicada para inferência.

Esses valores foram obtidos em uma máquina de uso geral, sem aceleração de hardware dedicada para inferência. Em dispositivos de borda com GPU embarcada ou NPU, como o NVIDIA Jetson Nano ou placas com acelerador Hailo, a carga de inferência OSNet seria parcialmente descarregada do processador principal, mitigando o impacto sobre a CPU e potencialmente reduzindo a latência de extração de *embeddings* na câmera.

Do lado do servidor, a diferença é igualmente significativa, porém em sentido oposto. Na Híbrida, o servidor concentra a inferência OSNet de todas as câmeras, tornando-se o gargalo do sistema. Na *Edge-First*, o servidor executa apenas o *matching* vetorial, uma operação trivial de álgebra linear que processa cada detecção em aproximadamente 1 ms, resultando em carga computacional negligível.

4.3 Discussão dos Resultados

Esta seção integra os resultados de acurácia e desempenho, discutindo as causas das diferenças observadas e o *trade-off* entre as arquiteturas.

4.3.1 Vantagem de acurácia da Edge-First: preservação da qualidade do crop

A superioridade da *Edge-First* em acurácia no EPFL Terrace (+11,1 p.p. em Rank-1) sugere uma vantagem intrínseca na qualidade dos *embeddings* extraídos. A hipótese mais provável é a preservação da fidelidade visual do *crop*: na *Edge-First*, o OSNet processa diretamente o recorte da imagem original em memória, sem qualquer transformação *lossy*.

Na Híbrida, o *crop* passa por codificação JPEG antes da transmissão, introduzindo artefatos de compressão (blocos, perda de detalhes finos, alteração sutil de cores) que podem degradar a representação de características discriminativas como texturas de tecido, padrões e bordas.

Embora a compressão JPEG com qualidade padrão preserve a aparência visual para o olho humano, redes neurais treinadas em dados de alta qualidade podem ser sensíveis a essas perturbações, especialmente em regiões de alta frequência espacial que carregam informação discriminativa para ReID (YE et al., 2022). Essa hipótese é reforçada pela ausência de diferença arquitetural no PRID2011: como ambas as arquiteturas processam as mesmas imagens PNG pré-recortadas (modo *crops*), sem compressão JPEG intermediária, o fator de degradação da Híbrida é eliminado e ambas apresentam desempenho equivalente (Rank-1 de 2,5% vs. 2,0%). O insucesso compartilhado no PRID2011 confirma que, nesse *dataset*, a limitação é do modelo OSNet, não da arquitetura.

4.3.2 Saturação da fila no servidor Hybrid

A latência da arquitetura Híbrida, com média de 110 s e p95 de 328 s no limiar 0,40, é dominada pelo tempo de espera na fila do servidor, não pelo tempo de inferência individual. O modelo OSNet processa cada *crop* em aproximadamente 20 ms; entretanto, com quatro câmeras transmitindo simultaneamente, cada uma gerando milhares de detecções ao longo do vídeo, a taxa de chegada de *crops* excede a capacidade de processamento do servidor.

Esse desequilíbrio cria um acúmulo progressivo: as detecções chegam mais rápido do que são processadas, e cada nova detecção aguarda na fila atrás de todas as anteriores. Como resultado, a latência cresce de forma monotônica ao longo do tempo de execução, conforme ilustrado na Figura 4.1. O p95 de 328 s, mais de cinco minutos, significa que 5% das detecções aguardaram mais de cinco minutos entre o recebimento pelo servidor e a conclusão do *matching*. Em um cenário operacional de vigilância, essa latência torna o sistema incapaz de fornecer resultados em tempo real.

Essa limitação é estrutural da arquitetura Híbrida: a centralização do processamento OSNet no servidor cria um gargalo que escala linearmente com o número de câmeras. A adição de mais câmeras agravaria o problema, enquanto na *Edge-First*, cada câmera adicional processa seus próprios *embeddings* localmente, sem impactar a capacidade do servidor.

4.3.3 Trade-off entre as arquiteturas

A análise conjunta dos resultados revela um *trade-off* claro entre as arquiteturas, sintetizado a seguir:

a) A *Edge-First* oferece melhor acurácia (Rank-1 +11,1 p.p. no EPFL), latência cinco ordens de magnitude menor (1 ms vs. 110 s), banda 45% inferior (140 vs. 256 MB) e carga mínima no servidor (apenas *matching*).

b) A Híbrida oferece menor carga nas câmeras (CPU 3,7 vs. 6,9 núcleos, memória 616 vs. 782 MB), transferindo a complexidade computacional para o servidor.

A escolha entre as arquiteturas depende, portanto, das características do dispositivo de borda disponível. Em dispositivos com capacidade moderada de processamento, como placas ARM com acelerador de inferência (NPU/GPU embarcada), a *Edge-First* é inequivocamente superior em todas as dimensões avaliadas. Em dispositivos severamente limitados, incapazes de executar dois modelos de inferência simultaneamente com *frame rate* aceitável, a Híbrida pode ser a única opção viável, embora com as limitações de latência e acurácia demonstradas.

Nos experimentos realizados, executados em uma máquina de uso geral sem aceleração de hardware dedicada, a *Edge-First* demonstrou viabilidade prática: o aumento de 85% no uso de CPU e 27% na memória da câmera não comprometeu a operação do sistema, que processou todos os vídeos sem perda de *frames* (0,0%) em ambas as arquiteturas.

4.3.4 Latência total e impacto na escalabilidade

Para compreender o impacto prático das diferenças de latência, é necessário decompor a latência total do sistema em suas três parcelas constitutivas, desde a captura do quadro até a conclusão do *matching*:

1. **Latência de detecção ($L_{\mathbf{det}}$)**: tempo de inferência YOLO no nó. Comum a ambas as arquiteturas, tipicamente ≈20 ms por quadro no ambiente utilizado.

2. **Latência de extração ($L_{\mathbf{ext}}$)**: tempo de inferência OSNet para gerar o *embedding*. Na Híbrida, ocorre no servidor (≈20 ms por *crop*); na *Edge-First*, ocorre no nó (mesmo custo unitário, mas distribuído entre as câmeras).

3. **Latência de *matching* ($L_{\mathbf{match}}$)**: tempo de comparação vetorial contra a galeria. Negligível em ambas as arquiteturas (≈1 ms).

A latência total percebida é, portanto:

$$L_{\text{total}} = L_{\text{det}} + L_{\text{transmissão}} + L_{\text{fila}} + L_{\text{ext}} + L_{\text{match}} \tag{4.1}$$

Na *Edge-First*, L_{ext} ocorre no nó (antes da transmissão) e $L_{\text{fila}} \approx 0$ porque o servidor processa apenas o *matching* trivial. Na Híbrida, L_{ext} e L_{fila} ocorrem no servidor, e L_{fila} domina a latência total por acúmulo progressivo.

O impacto na escalabilidade é direto: ao adicionar câmeras ao sistema, cada uma gera detecções que devem ser processadas. Na arquitetura Híbrida, o servidor acumula a carga de inferência OSNet de todas as câmeras, fazendo com que a latência de fila cresça linearmente com o número de nós ($L_{\text{fila}} \propto N_{\text{câmeras}}$). Com quatro câmeras, a latência média já atinge 110 s; com oito câmeras, o acúmulo dobraria, tornando o sistema inviável para operação em tempo real.

Na arquitetura *Edge-First*, a adição de câmeras não impacta a carga do servidor, pois cada nó processa seus próprios *embeddings* localmente. O servidor executa apenas o *matching* vetorial, cuja complexidade é $O(n)$ em relação ao número de identidades na galeria, não ao número de câmeras. Assim, a latência do servidor permanece praticamente constante ($\approx$1 ms) independentemente do número de nós, conferindo à *Edge-First* uma escalabilidade significativamente superior. O custo dessa escalabilidade é transferido para os dispositivos de borda, que devem possuir capacidade computacional suficiente para executar dois modelos de inferência simultaneamente (YOLO + OSNet).

4.3.5 Influência do limiar no comportamento da galeria

Os resultados evidenciam que o limiar de similaridade não afeta apenas a acurácia de forma direta, mas modula o comportamento dinâmico da galeria de identidades, com efeitos opostos nos dois *datasets* avaliados.

No EPFL Terrace, onde até 7 pessoas são observadas continuamente por 4 câmeras, limiares baixos ($\tau = 0{,}20$) fragmentam as identidades: o *matching* conservador falha em associar observações de ângulos ou iluminações diferentes da mesma pessoa, criando múltiplas entradas na galeria para um mesmo indivíduo. Limiares mais altos ($\tau = 0{,}40$) favorecem a fusão dessas observações, resultando em representações mais completas e melhores métricas de ReID.

No PRID2011, onde 200 identidades *probe* são comparadas com 749 identidades *gallery*, muitas visualmente similares, o sistema apresenta desempenho próximo ao acaso em todos os limiares avaliados (Rank-1 $\leq$ 2,5%). Nesse cenário, a variação do limiar não produz efeito

significativo sobre a acurácia, pois a limitação dominante é a baixa separabilidade dos *embeddings* do modelo `osnet_x1_0` (GAP de 0,062), que impede a discriminação efetiva entre identidades independentemente do limiar escolhido.

Essa assimetria sugere que a escolha do limiar ideal depende fortemente das características do cenário de implantação: número de indivíduos simultâneos, diversidade visual entre eles e duração da observação. Em sistemas de produção, uma abordagem promissora, indicada como trabalho futuro, seria a adaptação dinâmica do limiar com base no tamanho corrente da galeria e na distribuição de similaridades das detecções recentes.

4.3.6 Limitação do modelo OSNet no PRID2011

Conforme evidenciado na Tabela 4.2, o sistema não consegue vincular a mesma pessoa entre câmeras no PRID2011: o Rank-1 não ultrapassa 2,5% em nenhum dos limiares avaliados, valor próximo ao acaso. Para investigar a causa dessa limitação, foi conduzida uma análise de separabilidade dos *embeddings* extraídos pelo modelo `osnet_x1_0` utilizado nos experimentos.

A análise comparou a similaridade cosseno média entre observações da mesma pessoa em câmeras diferentes (similaridade intra-pessoa) com a similaridade entre pessoas distintas (similaridade inter-pessoa), utilizando 20 identidades do PRID2011 e 5 imagens por pessoa por câmera. A diferença entre essas médias, denominada GAP de separabilidade, indica a capacidade do modelo de distinguir indivíduos no espaço de *embeddings*. Foram avaliadas cinco variantes do OSNet, incluindo versões com AIN, treinadas em diferentes *datasets* de ReID.

Tabela 4.7 – Análise de separabilidade dos *embeddings* no PRID2011 para cinco variantes OSNet

Modelo (domínio de treino)	**Sim. intra**	**Sim. inter**	**GAP**
`osnet_x1_0` (ImageNet)	0,679	0,617	0,062
`osnet_ain` (CUHK03)	0,722	0,538	0,183
`osnet_ain` (DukeMTMC-reID)	0,712	0,526	0,186
`osnet_ain` (Market-1501)	0,689	0,512	0,177
`osnet_ain` (MSMT17)	0,682	0,514	0,168

Nota: Similaridade cosseno média sobre 20 identidades, 5 imagens por pessoa por câmera. GAP = similaridade intra-pessoa − similaridade inter-pessoa. Valores maiores indicam melhor capacidade de discriminação. As variantes AIN utilizam *Adaptive Instance Normalization* (ZHOU et al., 2021).

Fonte: Elaborado pelo autor.

A Tabela 4.7 revela que o modelo `osnet_x1_0`, utilizado em todos os experimentos,

apresenta um GAP de apenas 0,062 no PRID2011. Isso significa que a similaridade média entre uma pessoa e ela mesma vista por outra câmera (0,679) é apenas marginalmente superior à similaridade com pessoas completamente diferentes (0,617). Com uma margem tão estreita, qualquer variação de pose, iluminação ou resolução entre as câmeras pode inverter o *ranking*, fazendo com que o sistema atribua a detecção a uma identidade incorreta.

Em contraste, as variantes com AIN (OSNet-AIN), propostas por Zhou et al. (ZHOU et al., 2021), alcançam GAP entre 2,7 e 3,0 vezes maior (0,168 a 0,186), demonstrando capacidade de discriminação significativamente superior no PRID2011. A técnica AIN normaliza as estatísticas de estilo por instância, tornando o modelo mais robusto a variações de domínio como mudanças de iluminação e contraste entre câmeras.

Entretanto, substituir o modelo nos experimentos não foi considerado viável, pois invalidaria a comparação direta com os resultados já obtidos no EPFL Terrace, onde o `osnet_x1_0` apresentou desempenho adequado (Rank-1 de 67,6%). No EPFL Terrace, as câmeras compartilham condições visuais similares (mesma cena, iluminação uniforme, ângulos próximos), e o GAP de 0,062 mostrou-se suficiente para a tarefa.

4.3.7 Dependência do modelo em relação ao domínio

A análise de separabilidade evidencia uma conclusão importante para sistemas de ReID em produção: não existe um modelo universal de extração de *embeddings* que funcione adequadamente em qualquer cenário. O desempenho do modelo depende da correspondência entre as características visuais do ambiente de implantação e as condições do *dataset* de treinamento.

No caso do `osnet_x1_0`, treinado com *features* de propósito geral (ImageNet), o modelo gera representações suficientes para cenários com baixa variação inter-câmera (EPFL Terrace), mas insuficientes quando as câmeras apresentam diferenças significativas de ângulo, iluminação e resolução (PRID2011). As variantes OSNet-AIN, treinadas especificamente em *datasets* de ReID (Market-1501, DukeMTMC, CUHK03, MSMT17), incorporam mecanismos de normalização que atenuam essas variações de domínio, resultando em *embeddings* mais discriminativos.

Essa dependência reforça a necessidade, em implantações práticas, de selecionar ou adaptar o modelo de extração de características ao domínio específico de operação, considerando as condições visuais das câmeras instaladas. A avaliação de separabilidade realizada neste

trabalho pode servir como metodologia de diagnóstico para essa seleção.

5 CONCLUSÃO E TRABALHOS FUTUROS

5.1 Conclusões

Este trabalho teve como objetivo comparar duas arquiteturas de processamento distribuído para ReID em redes de múltiplas câmeras: a arquitetura Híbrida, na qual os nós de borda executam apenas a detecção (YOLO) e delegam a extração de *embeddings* (OSNet) e o *matching* ao servidor central; e a arquitetura *Edge-First*, na qual os nós executam tanto a detecção quanto a extração de *embeddings*, transmitindo ao servidor apenas o vetor de representação para o *matching* com a galeria global.

Os experimentos foram conduzidos em dois *datasets* públicos, EPFL Terrace (vídeo contínuo, 4 câmeras, *pipeline* completo) e PRID2011 (*crops* pré-recortados, 2 câmeras, *matching* isolado), com uma grade de quatro limiares de similaridade (0,20; 0,30; 0,35; 0,40), totalizando 16 configurações experimentais. A avaliação abrangeu tanto a acurácia de reidentificação (Rank-1, Rank-5, mAP) quanto o desempenho do sistema (latência, banda, CPU, memória).

Os resultados permitem responder à pergunta de pesquisa formulada: a arquitetura *Edge-First* é superior à Híbrida nas dimensões de desempenho avaliadas para o cenário experimental considerado. No EPFL Terrace, a *Edge-First* superou a Híbrida em Rank-1 por 11,1 pontos percentuais (67,6% vs. 56,5%, limiar 0,40). No PRID2011, ambas as arquiteturas apresentaram desempenho próximo ao acaso (Rank-1 $\leq$ 2,5%), limitação atribuída à baixa capacidade de discriminação do modelo `osnet_x1_0` nesse domínio visual, e não à arquitetura de processamento. Em termos de desempenho, a diferença é expressiva: a latência de ReID foi reduzida de aproximadamente 110 segundos para 1 milissegundo, cinco ordens de magnitude, e o consumo de banda diminuiu 45,2% (140,1 MB vs. 255,8 MB para quatro câmeras).

A vantagem de acurácia da *Edge-First* foi atribuída à preservação da qualidade do *crop*: o *embedding* é extraído diretamente da imagem original em memória, sem a degradação introduzida pela compressão JPEG durante a transmissão na arquitetura Híbrida. A vantagem de latência decorre da eliminação do gargalo de processamento no servidor: na Híbrida, o modelo OSNet centralizado não consegue acompanhar a taxa de chegada de *crops* de quatro câmeras simultâneas, gerando acúmulo progressivo na fila de processamento.

O custo dessa superioridade é o maior consumo de recursos nos nós de borda: aumento de 85% no uso de CPU e 27% na memória, em relação à Híbrida. Entretanto, nos experimentos

realizados, esse aumento não comprometeu a operação do sistema, que processou todos os vídeos sem perda de *frames*.

Os objetivos específicos do trabalho foram atendidos: (i) o sistema multi-câmera foi implementado com duas arquiteturas funcionais; (ii) o protocolo de comunicação UDP com formato de pacote próprio demonstrou-se eficaz para ambas as abordagens; (iii) a avaliação quantitativa em dois *datasets* permitiu uma comparação rigorosa de acurácia e desempenho; e (iv) a análise de sensibilidade ao limiar de similaridade revelou comportamentos assimétricos entre os *datasets*, evidenciando a importância da calibração desse parâmetro para cada cenário de implantação.

5.2 Limitações

Os resultados apresentados devem ser interpretados à luz das seguintes limitações:

a) **Ambiente controlado:** os experimentos foram conduzidos com vídeos pré-gravados transmitidos em rede local (*localhost*), sem latência, perda de pacotes ou variabilidade de largura de banda reais. Em redes Wi-Fi com congestionamento, a taxa de perda de datagramas UDP poderia afetar ambas as arquiteturas de forma diferente.

b) **Ausência de aceleração de hardware:** os nós de borda foram emulados em uma máquina de uso geral, sem GPU dedicada ou acelerador NPU. Em dispositivos embarcados reais, o desempenho do OSNet no nó poderia ser significativamente diferente, tanto positiva quanto negativamente, dependendo do hardware disponível.

c) **Limiar estático:** o limiar de similaridade foi fixo durante cada experimento. A análise demonstrou que o valor ótimo depende das características do cenário (número de indivíduos, diversidade visual, duração da observação), sugerindo que um limiar adaptativo seria mais robusto em condições variáveis.

d) **Escala dos *datasets*:** o EPFL Terrace contém no máximo 7 pessoas simultâneas e o PRID2011 possui 200 identidades *probe*. *Datasets* de maior escala (MARS, Market-1501) permitiriam avaliar o comportamento do sistema com centenas ou milhares de identidades, testando os limites das políticas de evicção da galeria.

e) **Modelo de detecção:** o YOLOv12n opera com limiar de confiança de 0,7 e processa um a cada cinco *frames*. Configurações mais agressivas poderiam aumentar a taxa

de detecções, e, consequentemente, a carga sobre a galeria e a rede, alterando potencialmente o equilíbrio entre as arquiteturas.

5.3 Trabalhos Futuros

Com base nos resultados obtidos e nas limitações identificadas, as seguintes direções são propostas para trabalhos futuros:

a) **Avaliação em hardware embarcado real:** implantar a arquitetura *Edge-First* em dispositivos como Raspberry Pi 5 com acelerador NPU ou NVIDIA Jetson Nano, medindo o impacto da aceleração de hardware sobre a latência de inferência e o consumo energético nos nós.

b) **Adaptação dinâmica do limiar de similaridade:** desenvolver um mecanismo que ajuste o limiar automaticamente com base no tamanho corrente da galeria e na distribuição de similaridades das detecções recentes, buscando equilibrar fragmentação e fusão de identidades sem intervenção manual.

c) **Testes em cenários com ruído de rede:** avaliar o sistema em redes Wi-Fi reais, com perda de pacotes, congestionamento e latência variável, para verificar a robustez das duas arquiteturas em condições operacionais mais próximas do mundo real.

d) **Validação em *datasets* de maior escala:** executar os experimentos com *datasets* como MARS (1.261 identidades, 20.478 sequências) e Market-1501 (1.501 identidades, 32.668 imagens), avaliando a escalabilidade da galeria e das políticas de evicção.

e) **Triangulação de posição multi-câmera:** integrar informações geométricas das câmeras para estimar a posição 3D dos indivíduos, seguindo abordagens similares às implementadas em plataformas como o NVIDIA DeepStream. Essa funcionalidade permitiria não apenas reidentificar, mas também localizar indivíduos no espaço físico, ampliando as possibilidades de aplicação em segurança e gestão de espaços.

f) **Criptografia de transporte:** implementar DTLS (*Datagram Transport Layer Security*) sobre UDP para proteger os *embeddings* transmitidos na *Edge-First*, garantindo que dados biométricos não possam ser interceptados em trânsito.

REFERÊNCIAS

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2005. p. 886–893.

DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: **International Conference on Learning Representations (ICLR)**. [s.n.], 2021. Disponível em: <https://arxiv.org/abs/2010.11929>.

FEI, L.; HAN, B. Multi-object multi-camera tracking based on deep learning for intelligent transportation: A review. **Sensors**, MDPI, v. 23, n. 8, p. 3852, 2023.

FLEURET, F. et al. Multicamera people tracking with a probabilistic occupancy map. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.l.: s.n.], 2008. v. 30, n. 2, p. 267–282.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016.

HE, K. et al. Deep residual learning for image recognition. In: **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

HE, L. et al. Fastreid: A pytorch toolbox for general instance re-identification. In: **ACM International Conference on Multimedia**. [S.l.: s.n.], 2023. p. 4321–4329.

HIRZER, M. et al. Person Re-Identification by Descriptive and Discriminative Classification. In: **Proc. Scandinavian Conference on Image Analysis (SCIA)**. [S.l.: s.n.], 2011.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017. Disponível em: <https://arxiv.org/abs/1704.04861>.

KHAN, S. M.; SHAH, M. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2006. p. 133–146.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2012.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998.

LIN, T.-Y. et al. Microsoft coco: Common objects in context. In: **European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2014. p. 740–755.

LIU, Y. et al. Privacy-preserving video surveillance on edge devices using deep learning. **IEEE Transactions on Industrial Informatics**, v. 16, n. 5, p. 3592–3602, 2020.

REDMON, J. et al. You only look once: Unified, real-time object detection. In: **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 779–788.

REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. **arXiv preprint arXiv:1804.02767**, 2018. Disponível em: <https://arxiv.org/abs/1804.02767>.

REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: **Advances in Neural Information Processing Systems (NeurIPS)**. [S.l.: s.n.], 2015. p. 91–99.

SATYANARAYANAN, M. The emergence of edge computing. **Computer**, IEEE, v. 50, n. 1, p. 30–39, 2017.

SHI, W. et al. Edge computing: Vision and challenges. **IEEE Internet of Things Journal**, v. 3, n. 5, p. 637–646, 2016.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014. Disponível em: <https://arxiv.org/abs/1409.1556>.

SUN, Y. et al. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In: **European Conference on Computer Vision (ECCV)**. Munich, Germany: [s.n.], 2018.

SZEWCZYK, P. W. et al. Survey of visual surveillance technologies for public safety. **Sensors**, v. 20, n. 24, p. 7181, 2020.

TIAN, Y.; YE, Q.; DOERMANN, D. Yolov12: Attention-centric real-time object detectors. **arXiv preprint arXiv:2502.12524**, 2025. Disponível em: <https://arxiv.org/abs/2502.12524>.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: **IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2001. p. 511–518.

WU, Y. et al. Unsupervised graph association for person re-identification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 43, n. 5, p. 1652–1668, 2021.

YE, M. et al. Deep learning for person re-identification: A survey and outlook. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 6, p. 2872–2893, 2022.

ZHENG, L. et al. Person re-identification: Past, present and future. **arXiv preprint arXiv:1610.02984**, 2016. Disponível em: <https://arxiv.org/abs/1610.02984>.

ZHOU, K. et al. Learning generalisable omni-scale representations for person re-identification. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 44, n. 10, p. 5056–5069, 2021.

ZHOU, K. et al. Omni-scale feature learning for person re-identification. In: **IEEE/CVF International Conference on Computer Vision (ICCV)**. Seoul, Korea: [s.n.], 2019.