UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Samuel Cunha Cotta

Um Estudo de Design Science Research (DSR) Utilizado em uma Abordagem de Inteligência Artificial na Redução de Dados em Bordas Computacionais

Juiz de Fora
2025

Samuel Cunha Cotta

Um Estudo de Design Science Research (DSR) Utilizado em uma Abordagem de Inteligência Artificial na Redução de Dados em Bordas Computacionais

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Mário Antônio Ribeiro Dantas

Coorientador: Prof. Dr. Marco Antônio Pereira Araújo

Juiz de Fora
2025

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Cotta, Samuel Cunha.
Um Estudo de Design Science Research (DSR) Utilizado em uma Abordagem de Inteligência Artificial na Redução de Dados em Bordas Computacionais / Samuel Cunha Cotta. -- 2025.
95 f. : il.

Orientador: Mário Antônio Ribeiro Dantas
Coorientador: Marco Antônio Pereira Araújo
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2025.

1. Autoencoder. 2. Borda Computacional. 3. Inteligência Artificial. 4. Compressão de Imagens. 5. Metodologia DSR. I. Dantas, Mário Antônio Ribeiro, orient. II. Araújo, Marco Antônio Pereira, coorient. III. Título.

Samuel Cunha Cotta

Um Estudo de Design Science Research (DSR) Utilizado em uma Abordagem de Inteligência Artificial na Redução de Dados em Bordas Computacionais

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Ciência da Computação.

Aprovada em 17 de novembro de 2025.

BANCA EXAMINADORA

Prof. Dr. Mario Antonio Ribeiro Dantas - Orientador

Universidade Federal de Juiz de Fora

Prof. Dr. Marco Antonio Pereira Araújo - Coorientador

Universidade Federal de Juiz de Fora

Profª. Dra. Bárbara de Melo Quintela

Universidade Federal de Juiz de Fora

Profª. Dra. Milena Faria Pinto

Centro Federal de Educação Tecnológica do Rio de Janeiro

Juiz de Fora, 03/11/2025.

Documento assinado eletronicamente por **Mario Antonio Ribeiro Dantas**, **Professor(a)**, em 03/12/2025, às 11:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Barbara de Melo Quintela**, **Professor(a)**, em 03/12/2025, às 11:09, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Marco Antonio Pereira Araujo**, **Professor(a)**, em 03/12/2025, às 21:52, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Milena Faria Pinto**, **Usuário Externo**, em 09/12/2025, às 13:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **2716801** e o código CRC **35B74626**.

Dedico este trabalho à família e amigos por todo suporte.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela saúde, força e oportunidade de chegar até aqui, enfrentando com serenidade e resiliência os desafios e aprendendo com cada um deles.

À minha família, base de tudo, pelo amor incondicional, paciência e compreensão nos momentos de ausência. Cada conquista aqui alcançada é também de vocês.

Aos meus orientadores, Prof. Dr. Mário Antônio Ribeiro Dantas e Prof. Dr. Marco Antônio Pereira Araújo, pela orientação técnica, discussões profundas e pela confiança depositada neste trabalho. A experiência e compromisso de vocês com a pesquisa foram fundamentais para o amadurecimento científico e para o desenvolvimento deste estudo.

Aos colegas, servidores e professores do Programa de Pós-Graduação em Ciência da Computação (PPGCC) da Universidade Federal de Juiz de Fora (UFJF), pelas contribuições nas disciplinas, pelas conversas inspiradoras e pelo ambiente de colaboração que tornaram esta jornada enriquecedora.

Agradeço também à equipe de suporte técnico e à infraestrutura computacional da UFJF, especialmente aos responsáveis pelos ambientes de processamento e servidores GPU utilizados neste trabalho da Rede Integrada de Pesquisa em Alta Velocidade (REPESQ), que viabilizaram os testes e simulações de grande porte realizados.

Por fim, a todos que, de alguma forma, contribuíram direta ou indiretamente para a realização deste trabalho de pesquisa, deixo meu reconhecimento e gratidão.

“We can only see a short distance ahead but we can see plenty there that needs to be done”. (Alan Turing)

RESUMO

O crescente volume de dados digitais, o pilar da Inteligência Artificial (IA), oriundos da borda computacional, especialmente em aplicações com Veículos Aéreos Não Tripulados (VANT) que capturam grandes quantidades de imagens em campo, impõe um desafio crítico: a necessidade de otimizar o processamento e a latência em dispositivos com severas restrições de hardware e energia. Visando analisar e comparar métodos de compressão de imagens baseados em IA para mitigar a latência na borda, este trabalho empregou a metodologia *Design Science Research* (DSR) em dois ciclos iterativos. A solução desenvolvida consistiu na implementação de modelos de *Autoencoder* (AE) de diferentes complexidades. Convencional, AE Variacional (VAE) e Penalizado por Redundância, treinados em ambiente de nuvem de alto desempenho utilizando imagens de VANT do dataset SARD-2. No Ciclo 1, os modelos foram comparados e otimizados quanto à qualidade de reconstrução e eficiência computacional, revelando que uma estrutura simples e otimizada é mais eficaz que arquiteturas excessivamente complexas. O AE Convencional Otimizado superou as variantes mais complexas, alcançando o melhor equilíbrio entre qualidade (PSNR = 20,71 dB; MS-SSIM = 0,9359) e tempo de processamento. No Ciclo 2, o modelo vencedor foi convertido e executado em ambiente de borda simulada por meio da plataforma OpenVINO, com simulação de hardware restrito e precisão FP32. A validação experimental demonstrou latência média de 21,2 ms e estabilidade temporal adequada para aplicações quase em tempo real, confirmando a viabilidade do modelo leve em dispositivos embarcados. Os resultados consolidam a tese de que a eficiência na borda depende da adequação estrutural do modelo ao hardware, e não apenas da sofisticação algorítmica. Este trabalho contribui como baseline metodológico replicável para compressão inteligente de imagens VANT, conciliando desempenho, sustentabilidade computacional e princípios da Ciência Aberta.

Palavras-chave: Autoencoder; Borda Computacional; Inteligência Artificial; Compressão de Imagens; Sustentabilidade Computacional; Metodologia DSR; Ciência Aberta.

ABSTRACT

The growing volume of digital data, the pillar of Artificial Intelligence (AI), originating from the computational edge, especially in applications with Unmanned Aerial Vehicles (UAVs) that capture large amounts of images in the field, poses a critical challenge: the need to optimize processing and latency on devices with severe hardware and power constraints. Aiming to analyze and compare AI-based image compression methods to mitigate latency at the edge, this work employed the Design Science Research (DSR) methodology in two iterative cycles. The solution developed consisted of implementing Autoencoder (AE) models of different complexities. Conventional, Variational Autoencoder (VAE), and Redundancy-Penalized Autoencoder, trained in a high-performance cloud environment using UAV images from the SARD-2 dataset. In Cycle 1, the models were compared and optimized for reconstruction quality and computational efficiency, revealing that a simple and optimized structure is more effective than overly complex architectures. The Optimized Conventional AE outperformed the more complex variants, achieving the best balance between quality (PSNR = 20.71 dB; MS-SSIM = 0.9359) and processing time. In Cycle 2, the winning model was converted and run in a simulated edge environment using the OpenVINO platform, with restricted hardware simulation and FP32 precision. Experimental validation demonstrated an average latency of 21.2 ms and adequate temporal stability for near real-time applications, confirming the viability of the lightweight model in embedded devices. The results consolidate the thesis that efficiency at the edge depends on the structural adequacy of the model to the hardware, and not only on algorithmic sophistication. This work contributes as a replicable methodological baseline for intelligent UAV image compression, reconciling performance, computational sustainability, and Open Science principles.

Keywords: Autoencoder; Edge Computing; Artificial Intelligence; Image Compression; Computational Sustainability; DSR methodology;Open Science.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AE	Autoencoders (Codificadores Automáticos)
CPU	Central Processing Unit (Unidade Central de Processamento)
dB	Decibels (Decibéis)
DCT	Discrete Cosine Transform (Transformada Discreta de Cosseno)
DAE	Deep Autoencoder (Autoencoder Profundo)
DR	Data Reduce (Redução de Dados)
DSR	Design Science Research (Pesquisa em Ciência do Design)
DSRM	Design Science Research Methodology (Metodologia de Pesquisa em Ciência do Design)
FP32	Floating Point 32-bit (Precisão em 32 bits)
FP16	Floating Point 16-bit (Precisão em 16 bits)
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico)
IA	Inteligência Artificial
IoT	Internet of Things (Internet das Coisas)
JPEG	Joint Photographic Experts Group (Conjunto de Especialistas em Fotografia)
LDA	Linear Discriminant Analysis (Análise Discriminante Linear)
ML	Machine Learning (Aprendizado de Máquina)
MPEG	Motion Picture Experts Group
ms	milisegundos
MSE	Mean Squared Error (Erro Quadrático Médio)
MS-SSIM	Multi-Scale Structural Similarity Index Measure (Medida do Índice de Similaridade Estrutural Multiescala)
ONXX	Open Neural Network Exchange (Troca de Redes Neurais Abertas)
OPENVINO	Open Visual Inference and Neural Network Optimization toolkit (Ferramentas Abertas para Inferência Visual e Otimização de Redes Neurais)
p95	Percentil 95
PCA	Principal Component Analysis (Análise de Componentes Principais)

PPGCC	Programa de Pós-Graduação em Ciência da Computação (UFJF)
PSNR	Peak Signal-to-Noise Ratio (Relação Sinal-Ruído de Pico)
RAM	Random Access Memory (Memória de Acesso Aleatório)
REPESQ	Rede Integrada de Pesquisa em Alta Velocidade
RF	Requisito Funcional
RNF	Requisito Não Funcional
SAR	Search and Rescue (Busca e resgate)
SARD 2	Search and Rescue Dataset (base de dados de busca e resgate)
SSIM	Structural Similarity Index Measure (Medida do Índice de Similaridade Estrutural)
TC	Taxa de Compressão
UFJF	Universidade Federal de Juiz de Fora
VAE	Variational Autoencoder (Autoencoder Variacional)
VRAM	Video Random Access Memory (Memória de Acesso Aleatório de Vídeo)
VANT	Veículos Aéreos Não Tripulados

SUMÁRIO

1. INTRODUÇÃO

1.1. MOTIVAÇÃO E CONTEXTUALIZAÇÃO

A explosão de dispositivos da *Internet of Things* (IoT) em cenários contemporâneos (Indústria 4.0, cidades inteligentes, agricultura, veículos autônomos, saúde eletrônica, energias renováveis, monitoramento de eventos climáticos e busca e resgate, SAR) tem gerado um volume massivo e heterogêneo de dados. Estima-se que, até 2025, existam 42 bilhões de dispositivos IoT conectados, produzindo 80 *zettabytes* de dados anualmente (UPADRISTA, 2021), volume este crescendo sem precedentes (MAFTEI et al. 2025). O desafio está na necessidade de manipulação, processamento e armazenamento eficiente desses registros, já que o envio direto dos dados para a nuvem pode causar tráfego excessivo, aumento de latência, consumo de banda e energia. Estratégias inovadoras, como *frameworks* baseados em *blockchain* e borda computacional, são exemplos de alternativas para superar essas limitações e garantir a escalabilidade e eficiência dos sistemas IoT (FAZELDEHKORDI & GRØNLI, 2022; BARBUTO et al. 2023; MAFTEI et al. 2025).

Dentre os diferentes tipos de dados gerados na IoT, as informações visuais, especialmente imagens, destacam-se pelo seu volume elevado e pelos desafios únicos de processamento, transmissão e armazenamento. Imagens provenientes de sensores, câmeras inteligentes, VANT e veículos autônomos constituem uma fração significativa do tráfego de dados em ambientes de borda, impactando diretamente a necessidade de soluções que atendam aos requisitos de baixa latência e alta eficiência operacional. Além disso, aplicações críticas como segurança pública, monitoramento ambiental e saúde digital exigem que a transmissão e o processamento de imagens e dados ocorram em tempo real ou quase tempo real, a depender da aceitação de pequenos atrasos (GOMES, 2021).

Neste contexto, os VANT ocupam posição de destaque. Esses sistemas capturam imagens em alta resolução e operam em cenários dinâmicos, remotos e frequentemente degradados, nos quais a conectividade é limitada e o processamento embarcado é restrito. Em aplicações como logística, monitoramento ambiental, busca e salvamento e fotogrametria, sistemas multi-robôs móveis devem possuir uma rede de comunicação confiável entre os veículos, garantindo que as

informações trocadas entre os nós tenham poucas perdas (RAMOS et al., 2023; GEGENAVA, 2025).

Nesses cenários, o envio bruto de imagens 1080p ou 2K para a nuvem resulta em tráfego elevado, consumo excessivo de energia e atrasos incompatíveis com operações quase em tempo real. Estudos recentes mostram que a latência de transmissão e processamento é um dos fatores que mais impactam a eficácia de missões com VANT, especialmente quando operando em redes 4G/5G instáveis ou enlaces de rádio de baixa capacidade (ZHANG et al., 2024; REDDI, 2025). Assim, métodos de compressão inteligente na borda tornam-se essenciais para viabilizar a operação contínua e eficiente dos VANT.

Portanto, a compressão de dados visuais em VANTs não é apenas uma etapa de otimização, mas um requisito operacional para garantir baixa latência, economia de banda e autonomia energética. É neste contexto que este trabalho se insere, investigando técnicas baseadas em AE para reduzir o volume de imagens geradas por esses veículos, preservando sua utilidade para tarefas críticas em ambientes de borda.

Além disso, a computação de borda surge como uma solução promissora para mitigar esses problemas, ao permitir que o tratamento do conteúdo ocorra próximo à sua origem (REDDI, 2025), ou seja, nos próprios dispositivos IoT. Essa abordagem possibilita a execução de tarefas de *data reduce* (DR) diretamente nos dispositivos de borda, diminuindo a necessidade de transferência excessiva para a nuvem (PIOLI et al., 2024). Além disso, esse ambiente computacional oferece suporte à mobilidade, distribuição geográfica, reconhecimento de localização e respostas rápidas, requisitos nem sempre atendidos pela nuvem (POWELL, DESINIOTIS & DEZFOULI, 2020). Soluções como borda e névoa computacional visam transferir o processamento e a inteligência para mais perto das fontes, evitando a dependência de servidores remotos (ZHOU et al., 2019; KOLAPO et al., 2024; UMEH, I. & UMEH, K. 2024). Embora a computação de borda represente uma solução promissora, é preciso considerar as restrições dos dispositivos de borda, que frequentemente operam com recursos limitados, como processadores de baixa potência e memória reduzida (KOLAPO et al., 2024; PIOLI et al., 2024).

Neste contexto de *big data* visual, uma alternativa é o uso de técnicas de DR visuais, que consiste em transformar um conjunto de registros visuais em um volume menor, mantendo sua qualidade e integridade antes da transmissão (PIOLI et al.,

2024). Essas técnicas são essenciais para otimizar o uso de banda, reduzir custos de armazenamento e possibilitar o processamento eficiente em dispositivos de borda, especialmente em aplicações que exigem respostas rápidas e baixo consumo de energia.

Com os avanços em computação de borda e da IA, a tecnologia IoT atinge um novo patamar. A integração de IA com a borda computacional, denominada inteligência de borda, permite a criação de aplicações mais inteligentes e eficientes (DENG et al., 2020) ou aprendizado de máquina na borda (REDDI, 2025), que para este trabalho são considerados sinônimos. Essa abordagem não só minimiza a latência e o volume de tráfego de conteúdo visual, como também viabiliza decisões em tempo real, melhora escalabilidade, privacidade e confiabilidade (BARBUTO et al., 2023). A execução de algoritmos de IA nesses dispositivos exige estratégias de otimização, como compressão de modelos e utilização de redes neurais leves, viabilizando o processamento local de informações visuais sem comprometer a autonomia energética.

Este trabalho se fundamenta em estudos recentes que analisam estratégias para DR em ambientes de borda e inteligência embarcada, evidenciando sua relevância atual (BARBUTO et al., 2023; PIOLI et al., 2024; PIOLI, 2025).

1.2. JUSTIFICATIVA

O avanço exponencial das aplicações de IA em cenários distribuídos, como IoT e os sistemas Ciber-Físicos, impõe a necessidade urgente de explorar arquiteturas de Computação de Borda (AHMAD et al., 2023; ANDRIULO et al., 2024; SHI et al., 2016). Essa abordagem é fundamental para que aplicações sensíveis à latência, como veículos autônomos e sistemas de monitoramento baseados em VANT, possam operar com a rapidez e a autonomia exigidas em missões críticas (DENG et al., 2020; RAMOS et al., 2023).

Apesar do volume expressivo de publicações na área, parte relevante dos trabalhos ainda apresenta limitações no que se refere à validação experimental em contextos reais ou próximos de cenários operacionais (BARBUTO et al., 2023; PIOLI et al., 2024). Essa lacuna é ainda mais evidente em dispositivos de borda com restrições severas de processamento, memória e energia (CAO et al., 2020; PIOLI JUNIOR, 2024), nos quais a necessidade de eficiência é crítica. O desafio se intensifica porque os dados, elemento fundamental para o treinamento e a inferência

dos modelos de IA, são produzidos diretamente na borda e representam grande parte do contexto situacional (KONG et al., 2022; PIOLI et al., 2025).

A evolução das redes de comunicação de próxima geração (5G e 6G) contribui para ampliar a capacidade de transmissão e o processamento distribuído, permitindo maior produção e manipulação de dados diretamente em dispositivos de borda (SINGH et al., 2024; KOLAPO et al., 2024; ADHIKARI; HAZRA, 2022). Essa expansão tecnológica reforça a necessidade de conduzir experimentos de IA capazes de reduzir dados visuais localmente, de forma eficiente e reprodutível, para que a infraestrutura emergente seja explorada de maneira adequada (BAO et al., 2023; HAMDAN et al., 2020). Assim, a promessa de baixa latência, resiliência e operação em tempo quase real depende diretamente da capacidade de adaptação dos modelos aos dispositivos restritos (DENG et al., 2020; PIOLI et al., 2025).

Neste contexto, o presente trabalho também se justifica pelo alinhamento direto com as demandas das missões de SAR, que dependem de VANT para captura contínua de imagens e tomada de decisão rápida. Esses cenários envolvem limitações severas de processamento embarcado, forte restrição energética e comunicação instável, o que torna a compressão inteligente um elemento essencial para reduzir o tráfego de dados sem comprometer a qualidade visual necessária para a detecção de vítimas ou objetos relevantes. Dessa forma, a combinação entre VANT, AE e uma arquitetura híbrida borda-nuvem surge como alternativa viável e necessária (RAMOS et al., 2023; SINGH; GILL, 2023; ZHOU et al., 2019).

Adicionalmente, operações SAR frequentemente ocorrem em ambientes críticos, com baixa visibilidade, interferências e necessidade de tomada de decisão quase imediata. No entanto, VANT utilizam *hardware* embarcado limitado, geralmente baseado em arquiteturas ARM, com pouca memória e autonomia energética reduzida. A transmissão de imagens de alta resolução intensifica o tráfego de dados e, quando realizada integralmente na nuvem, introduz latências que inviabilizam aplicações sensíveis ao tempo. Métodos tradicionais de compressão podem ainda suprimir detalhes essenciais à identificação de alvos relevantes, reforçando a necessidade de abordagens mais robustas e alinhadas ao contexto da borda (RAMOS et al., 2023; SINGH; GILL, 2023; ZHOU et al., 2019).

Apesar dos avanços em comunicação e infraestrutura, métodos tradicionais de compressão de imagens como JPEG, JPEG2000 ou esquemas baseados em transformadas, frequentemente apresentam limitações em cenários de borda, pois

podem degradar detalhes essenciais ou demandar etapas de processamento não otimizadas para *hardware* restrito. Nesse contexto, técnicas de Aprendizado Profundo (*Deep Learning*) têm ganhado destaque por sua capacidade de aprender representações compactas diretamente dos dados, adaptando-se ao conteúdo visual e às restrições da plataforma. Entre essas técnicas, os AE destacam-se como uma alternativa moderna e flexível para compressão, pois realizam a redução dimensional por meio de um encoder leve, capaz de ser executado em dispositivos embarcados, enquanto preservam características visuais relevantes para tarefas críticas. Assim, a escolha por AE decorre da necessidade de um mecanismo de compressão aprendido, adaptável ao *dataset* do VANT e compatível com as limitações computacionais da borda, o que justifica sua utilização como artefato central neste estudo (AZIZIAN; BAJIĆ, 2024; OLIVEIRA et al., 2021; BERAHMAND et al., 2024).

Dessa forma, este estudo propõe uma abordagem de validação experimental, fundamentada na metodologia DSR (HEVNER et al., 2004; PEFFERS et al., 2007), utilizando AE adaptados para execução em dispositivos de borda. O objetivo é mitigar desafios de latência e consumo de recursos diante do crescente volume de dados gerados localmente, fornecendo evidências práticas para a adoção de modelos leves e eficientes em cenários críticos (AZIZIAN; BAJIĆ, 2024; OLIVEIRA et al., 2021; BERAHMAND et al., 2024).

1.3. DEFINIÇÃO DO PROBLEMA

Apesar dos avanços recentes em técnicas DR visuais aplicadas a ambientes computacionais de borda, persistem lacunas significativas quanto à adaptação dessas abordagens às severas restrições de processamento, energia e latência desses dispositivos. Esse desafio torna-se ainda mais evidente em plataformas móveis como VANTs, que geram grandes volumes de imagens em alta resolução, mas operam com *hardware* embarcado limitado e conectividade variável.

Além disso, há escassez de estudos que realizem validações práticas em cenários heterogêneos e próximos da realidade operacional desses sistemas, especialmente no que se refere à eficácia das soluções de compressão diante de requisitos rígidos de latência para transmissão e reconstrução de conteúdo visual. Essa ausência de validação aplicada limita o avanço de técnicas realmente adequadas aos dispositivos de borda e às demandas de processamento distribuído.

1.4. QUESTÃO DE PESQUISA

Diante desse contexto, a seguinte questão central se destaca: como reduzir a latência durante a transmissão de imagens, por meio de métodos de compressão baseados em inteligência artificial, garantindo eficiência operacional e qualidade visual para aplicações sensíveis ao tempo?

1.5. OBJETIVOS

O objetivo geral deste trabalho é analisar e comparar métodos de compressão de imagens baseados em inteligência artificial para redução do tráfego de dados em ambientes de borda-nuvem. Para tanto, foram definidos 3 objetivos específicos:

- OE1: Sintetizar o estado da arte da compressão de imagens na borda, a partir de mapeamentos, revisões sistemáticas e estudos atuais, identificando tendências, limitações e lacunas.
- OE2: Implementar e adaptar modelos de AE para compressão de imagens, considerando restrições de dispositivos de borda.
- OE3: Avaliar experimentalmente o desempenho dos modelos quanto à latência, qualidade da reconstrução e eficiência de compressão.

1.6. HIPÓTESE DE PESQUISA

A literatura recente indica que arquiteturas de AE com maior sofisticação estrutural podem apresentar desempenho superior na compactação e reconstrução de imagens quando comparadas a AE convencionais otimizados. LAAKOM et al. (2024) demonstram que AE com penalização de redundância produzem representações latentes mais compactas e melhoram a fidelidade da reconstrução ao reduzir correlações indesejadas no gargalo. De modo semelhante, OLIVEIRA et al. (2021) evidenciam que modelos VAE alcançam melhor equilíbrio entre taxa de compressão e qualidade visual em cenários embarcados, aproximando-se das restrições de operação encontradas em sistemas móveis. Além disso, ZHU (2024), ao comparar diversas arquiteturas de AE, mostra que modelos mais complexos tendem a oferecer maior qualidade de reconstrução em relação a versões convencionais.

Diante dessas evidências, a hipótese central deste trabalho é a de que VAE e AE com penalização por redundância apresentam desempenho superior em compressão de imagens, em termos de qualidade de reconstrução e eficiência, quando comparados a um AE convencional otimizado, especialmente em ambientes de borda com restrições computacionais, como aqueles encontrados em aplicações com VANT.

Esta hipótese será testada por meio da construção, experimentação e avaliação quantitativa de diferentes arquiteturas de AE em dois ciclos da metodologia DSR. As métricas de avaliação incluem PSNR (*Peak Signal-to-Noise Ratio*), SSIM (*Structural Similarity Index*) e o MS-SSIM), taxa de compressão e latência, em ambientes computacionais distintos (nuvem e borda), simulando cenários operacionais com VANT.

1.7. CONTRIBUIÇÕES ESPERADAS

O estudo propõe consolidar o conhecimento recente sobre compressão de dados visuais em bordas computacionais*,* organizando tendências, desafios e lacunas da literatura. Pretende-se formalizar processos por meio da metodologia DSR, promovendo amadurecimento dos artefatos e geração de conhecimento científico. Serão adaptados e avaliados modelos de *AE* para compressão de imagens em cenários de borda, considerando restrições reais ou simuladas de dispositivos. O protocolo experimental prioriza a análise de latência, além de métricas tradicionais, respondendo a lacunas identificadas. Todos os experimentos e *pipelines* estão disponibilizados publicamente no GitHub[1], incentivando a Ciência Aberta. Por fim, espera-se fornecer recomendações práticas para aplicações reais de *borda* que demandam baixa latência e eficiência computacional, uma vez que o *deep* AE (DAE) reduz significativamente a dimensionalidade dos dados, diminuindo o tráfego de comunicação e o custo de processamento sem perda relevante de informação.

1.8. ORGANIZAÇÃO DO TRABALHO

Os conceitos teóricos e trabalhos relacionados deste estudo estão no Capítulo 2. No capítulo 3, por sua vez, é apresentada a metodologia seguida. Além

[1] Disponível em: https://github.com/samuelccotta/sar_autoencoders. Acesso em: 18 out. 2025

disso, o Capítulo 4 mostra os resultados experimentais relacionados aos experimentos em nuvem e de simulação na borda computacional. Finalmente, o Capítulo 5 conclui esta pesquisa, resumindo este trabalho e delineando futuros estudos.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos, fundamentos e avanços tecnológicos que embasam esta pesquisa, contextualizando o cenário atual das tecnologias aplicadas. Nele são abordados os princípios de IoT, seus componentes e desafios, o ambiente computacional de nuvem e de borda, as técnicas de DR visuais e a evolução para o uso de inteligência artificial nesse contexto, com foco nos AE. Cabe destacar a existência na literatura do ambiente de névoa, que é relevante, mas foi suprimido para fins de simplificação da solução, mantendo a arquitetura borda-nuvem como foco da solução. Esse referencial fundamenta as motivações teóricas e práticas do trabalho, fornecendo suporte para a análise crítica das soluções propostas e para o desenvolvimento do *pipeline* experimental aplicado à compressão de imagens em ambientes de borda computacional.

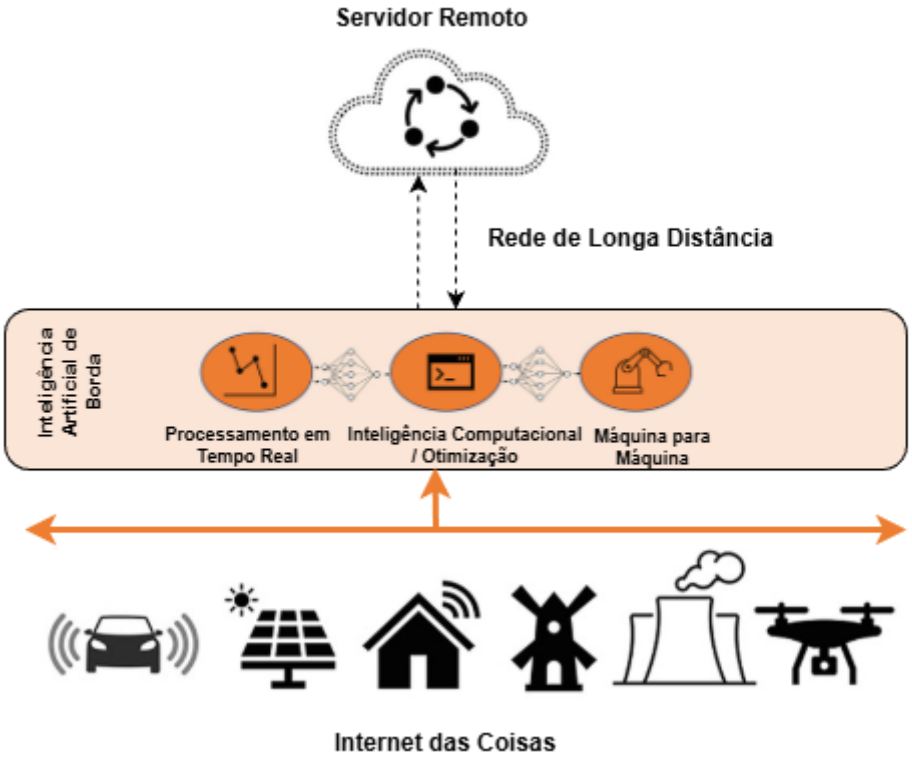
2.1. INTERNET DAS COISAS (IoT)

IoT pode ser definida como uma rede global de dispositivos inteligentes interconectados, capazes de coletar, processar e compartilhar dados automaticamente, utilizando diferentes protocolos e tecnologias de comunicação. O conceito central da IoT envolve a integração de sensores, atuadores e sistemas computacionais, permitindo aplicações em áreas como cidades inteligentes, saúde, agricultura e automação industrial (DIN et al., 2018; FURSTENAU et al., 2020; MANSOUR et al., 2023). A arquitetura clássica da IoT é geralmente estruturada em camadas, sendo o modelo de três camadas (percepção, rede e aplicação) o mais tradicional, enquanto revisões recentes destacam arquiteturas mais complexas, como as de cinco camadas, que incluem processamento em borda e camadas de suporte à segurança e gerenciamento (FURSTENAU et al., 2020; BANIJAMALI et al., 2020; MANSOUR et al., 2023).

Tendências atuais apontam para a adoção de tecnologias emergentes como inteligência artificial, computação em nuvem, 5G/6G e *blockchain*, que ampliam a escalabilidade, eficiência e segurança dos sistemas IoT (DIN et al., 2018; BANIJAMALI et al., 2020; MANSOUR et al., 2023). No entanto, entre os desafios arquiteturais estão a interoperabilidade, segurança, privacidade, confiabilidade, restrições energéticas e ausência de padrões comuns, exigindo adaptações conforme o cenário de aplicação (NIKOUI et al., 2020; BANIJAMALI et al., 2020). A

convergência entre IoT e computação em nuvem, bem como a adoção de arquiteturas orientadas a serviços (SOA) e microserviços, são tendências destacadas para garantir escalabilidade, automação e tomada de decisão autônoma (BANIJAMALI et al., 2020; RAZZAQ, 2020). A Figura 1 ilustra alguns dispositivos IoT, demonstrando seu papel na disponibilização dos dados para que a borda computacional possa realizar as inferências utilizando técnicas de inteligência artificial que serão abordadas em capítulos seguintes.

Figura 1 – Exemplo de dispositivos Internet das Coisas no contexto de Inteligência Artificial de Borda



Fonte: Adaptado de SINGH, GILL (2023).

2.2. AMBIENTE COMPUTACIONAL DE NUVEM

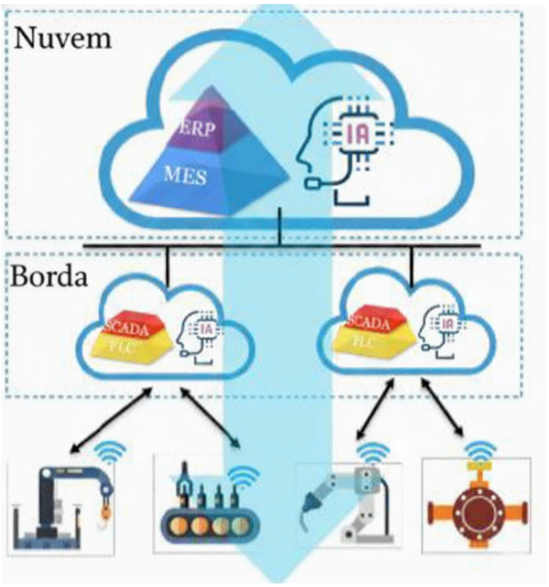
A computação em nuvem é um modelo computacional flexível que fornece serviços como servidores, armazenamento, bancos de dados, redes e *software* sob demanda via internet, eliminando a necessidade de infraestrutura física local e possibilitando escalabilidade dinâmica (MELL & GRANCE, 2011; AMAJUOYI et al., 2024). Desde a consolidação dos serviços de infraestrutura em nuvem na última década, seu uso se expandiu para suportar aplicações que demandam

processamento intensivo e análise de grandes volumes de dados (AMAJUOYI et al., 2024).

Nos últimos anos, a integração da computação em nuvem com abordagens emergentes, como borda e névoa computacional, tem sido central para superar limitações relacionadas à latência e segurança, especialmente nas aplicações IoT (ANDRIULO et al., 2024). A crescente adoção de *machine learning* e inteligência artificial na nuvem tem impulsionado melhorias significativas na análise de dados, eficiência operacional e automação adaptativa dos recursos computacionais (WANG et al., 2024).

Contudo, mesmo com esses avanços, persistem desafios cruciais como latência, privacidade, segurança dos dados, gerenciamento eficiente de recursos e o risco de dependência de fornecedores (*vendor lock-in*), que ainda são amplamente discutidos e motivam investigações para garantir maior robustez e adoção segura da computação em nuvem (ALSHAREEF, 2023). A Figura 2 ilustra como a IA se distribui entre nuvem e borda em arquiteturas modernas. Na nuvem, ficam as tarefas de maior custo computacional, como treinamento e otimização dos modelos. Já a borda executa as inferências próximas às fontes de dados, reduzindo latência e tráfego. A seta central representa esse fluxo: modelos treinados na nuvem são enviados para execução na borda. Essa organização reflete a abordagem adotada neste trabalho, que utiliza a nuvem para treinar os Autoencoders e a borda para realizar as inferências em dispositivos restritos.

Figura 2 – Exemplo de arquitetura *borda-nuvem, Internet das Coisas* e IA



Internet das Coisas, nuvem e borda computacional habilitado para indústria de sistemas de sistemas de automação

Fonte: Adaptado de NOVIK (2025).

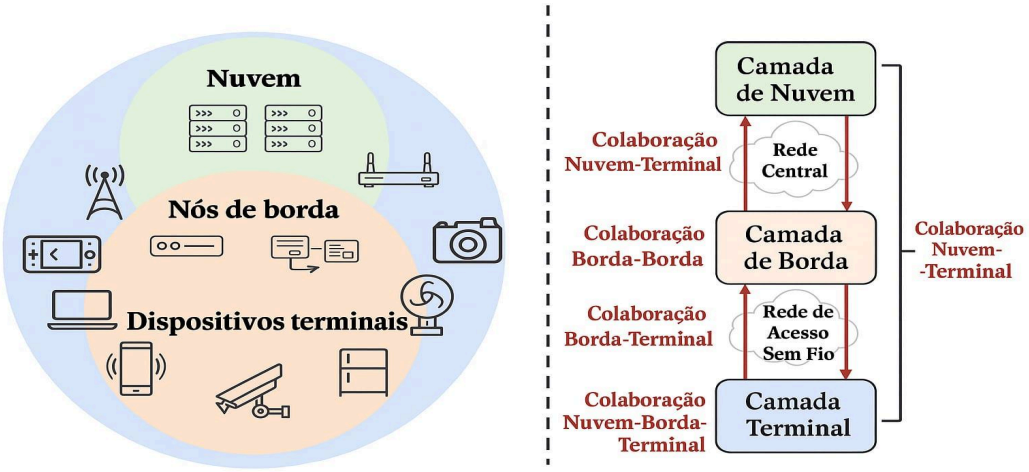
2.3. AMBIENTE COMPUTACIONAL DE BORDA

O ambiente computacional de borda, conhecido como borda computacional, refere-se a uma arquitetura distribuída que integra recursos de computação, armazenamento e rede próximos à fonte dos dados, como dispositivos IoT, *gateways* e servidores locais. Essa abordagem visa reduzir a latência, aumentar a eficiência no processamento de dados e melhorar a privacidade e a segurança, evitando o envio de grandes volumes de dados para *data centers* distantes na nuvem (SHI et al., 2016; CAO et al., 2020). Ambientes de borda são fundamentais para aplicações sensíveis ao tempo, como cidades inteligentes, veículos autônomos, realidade aumentada e monitoramento industrial, onde a resposta rápida é essencial (SHI et al., 2016; QIU et al., 2020; SULIEMAN et al., 2022).

Modelos e arquiteturas como computação de borda móvel, nuvem pequena e computação de névoa são adotados para atender demandas de mobilidade, escalabilidade e gerenciamento de recursos (HAMDAN et al., 2020; SULIEMAN et al., 2022). O ambiente de borda enfrenta desafios como alocação eficiente de tarefas, consumo de energia, segurança, privacidade e integração com tecnologias emergentes, incluindo inteligência artificial e *blockchain* (QIU et al., 2020; SINGH et al., 2022; SULIEMAN et al., 2022). Kong et al. (2022) destacam que o ambiente de

borda é peça-chave para a Internet de Tudo, ampliando o escopo de aplicações e exigindo soluções inovadoras para migração de serviços, implantação de nós de borda e integração com tecnologias como *digital twin* e 6G. Por fim, a computação de borda complementa a computação em nuvem, promovendo cooperação entre ambos para otimizar desempenho e qualidade dos serviços em aplicações modernas (HAMDAN et al., 2020; SULIEMAN et al., 2022). A estrutura da computação de ponta é geralmente dividida em três camadas: camada terminal, camada de borda e camada de nuvem, conforme ilustrado na Figura 3.

Figura 3 – Exemplo de arquitetura de borda



Fonte: Adaptado de KONG (2022).

2.4. REDUÇÃO DE DADOS EM BORDAS COMPUTACIONAIS: CONCEITOS E TÉCNICAS

A explosão de dispositivos conectados e sistemas embarcados resultou em desafios significativos para o tratamento e a transmissão de grandes volumes de dados em tempo quase real, especialmente em aplicações críticas como veículos autônomos, VANT, sensores industriais. A arquitetura de borda, ou computação de borda, visa distribuir parte do processamento para próximo das fontes de dados, reduzindo a dependência da nuvem e otimizando o uso de largura de banda e recursos computacionais locais.(SHI et al., 2016; DENG et al., 2020; KONG et al., 2022; KOLAPO et al., 2024).

Entre as técnicas tradicionais de DR, destacam-se algoritmos clássicos de compressão como JPEG, JPEG2000 e MPEG para imagens e vídeos. O algoritmo JPEG, adotado desde a década de 1990, realiza compressão com perdas através da transformação Discrete Cosine Transform (DCT) e quantização, mantendo boa qualidade visual em taxas moderadas de compressão (THAI; COGRANNE, 2019; SABZAVI; GHADERI, 2024). Já o JPEG2000 introduz a transformação *wavelet*, permitindo compressão ainda mais eficiente, progressiva e com melhor preservação de detalhes em taxas elevadas (LAWSON; ZHU, 2002; MA et al., 2020). Estas técnicas fazem parte de muitos processos embarcados, devido à sua eficiência, baixo custo computacional e ampla disponibilidade em bibliotecas e *hardware* dedicados (THAI; COGRANNE, 2019; LAWSON; ZHU, 2002).

Paralelamente, métodos de redução de dimensionalidade como a *Principal component Analysis* (PCA) e a *Linear Discriminant Analysis* (LDA) são frequentemente empregados para a compactação de grandes conjuntos de dados lineares, sendo úteis na filtragem de informações redundantes (REDDY et al., 2020; JIMÉNEZ-NARVÁEZ et al., 2023). Outras abordagens, como filtros de quantização e técnicas de amostragem, complementam o arsenal tradicional para redução de dados, cada qual com *trade-offs* entre fidelidade e consumo computacional (BEN SAAD; BEFERULL-LOZANO; ISUFI, 2020; ZHAO et al., 2024).

Entretanto, pesquisas recentes apontam que abordagens tradicionais de compressão enfrentam limitações importantes quando aplicadas a aplicações modernas de borda computacional. Estudos evidenciam que a relação entre taxa de compressão e qualidade dos dados pode ser significativamente degradada diante de restrições severas de banda e energia, tornando as soluções clássicas menos eficazes nesse cenário dinâmico (ALSHARIF et al. 2025; JÚNIOR et al. 2021) Segundo, algoritmos clássicos geralmente não se adaptam de forma dinâmica ao contexto ou ao conteúdo dos dados, o que pode limitar a eficiência quando comparados a abordagens aprendidas (PIOLI et al., 2024; BARBUTO et al., 2023). Por fim, as limitações de *hardware* de microcontroladores, matrizes de portas programáveis em campo e sistemas embarcados exigem compressão eficaz sem comprometer o desempenho da aplicação (DENG et al., 2020; ZHOU et al., 2019).

Há exceções importantes, como o ARCog-NET (RAMOS et al. 2024), uma arquitetura cognitiva avançada para VANTs em sistemas cooperativos, que integrou processamento distribuído no paradigma borda-névoa-nuvem e foi avaliado em um

cenário realista de inspeção de turbinas eólicas, apresentando resultados de simulação que mostram que o ARCog-NET reduz a latência, aumenta a taxa de transferência de dados e melhora a eficácia operacional. Todavia, o foco do ARCog-NET não está na construção de artefatos DSR, análise quantitativa multifatorial e ciência aberta, mas sim no comportamento colaborativo do enxame e na distribuição hierárquica de tarefas cognitivas. Assim, embora valida processamento embarcado em VANTs, ele não aborda diretamente a problemática tratada nesta dissertação.

Essas limitações motivam a adoção de técnicas baseadas em aprendizagem profunda, como os AE, que aprendem representações não lineares diretamente dos dados e podem ser ajustados especificamente para o domínio das imagens aéreas capturadas por VANT. Diferentemente dos métodos clássicos com transformações fixas, AE extraem características relevantes de maneira adaptativa, permitindo compactação mais eficiente sob restrições de banda, energia e processamento típicas da computação de borda. Evidências recentes demonstram que variantes como VAE e modelos com penalização de redundância produzem representações mais compactas e com maior fidelidade quando comparadas às abordagens tradicionais (OLIVEIRA et al., 2021; LAAKOM et al., 2024; ZHU, 2024; TENG et al., 2025). Essa combinação de adaptabilidade, eficiência e compatibilidade com ambientes embarcados fundamenta sua escolha neste trabalho.

2.5. APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL NA REDUÇÃO DE DADOS NA BORDA COMPUTACIONAL

A ascensão da IA, sobretudo aprendizado de máquina (ML) e redes neurais profundas, trouxeram novas perspectivas para a DR em ambientes de borda, especialmente com o uso de redes neurais profundas para compressão de dados visuais. Entre as técnicas promissoras está o uso de AE, redes neurais desenhadas para aprender representações compactas de entrada, sintetizando informações relevantes em um espaço latente reduzido. Dentre os modelos de AE disponíveis na literatura foram escolhidos para esse trabalho: convencional, variacional e com penalidade de redundância.

AE são redes neurais artificiais auto supervisionadas, utilizadas para aprender representações compactas e eficientes de dados, sendo amplamente aplicados em tarefas como compressão e detecção de anomalias. O AE convencional emprega

camadas convolucionais para capturar padrões espaciais em imagens, permitindo uma codificação eficiente e preservando características locais relevantes (LI et al., 2023; BERAHMAND et al., 2024). Já o VAE é um modelo generativo que aprende uma distribuição probabilística no espaço latente, possibilitando a geração de novas amostras e melhorando a capacidade de generalização; VAE têm apresentado desempenho superior em tarefas de reconstrução e classificação de imagens, especialmente quando combinados com arquiteturas convolucionais (CHEN et al., 2020; YU et al., 2021; LI et al., 2023; BERAHMAND et al., 2024).

Além disso, abordagens que penalizam a redundância no espaço latente, como a maximização da informação mútua entre variáveis latentes e entradas, buscam tornar as representações aprendidas mais informativas e compactas, reduzindo a redundância e promovendo maior eficiência na codificação (YU et al., 2021). Essas estratégias são particularmente relevantes em cenários que exigem compressão eficiente e alta capacidade de generalização, como aplicações de computação de borda e análise de grandes volumes de dados. Dessa forma, a escolha entre AE convencionais, variacionais ou penalizados por redundância depende dos objetivos específicos do problema e das características dos dados envolvidos.

Além dos AE, técnicas complementares vêm sendo aplicadas para aprimorar modelos de IA na borda computacional:

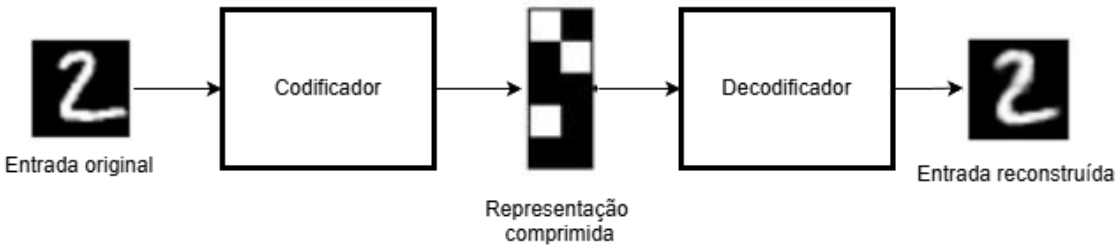
- Quantização pós-treinamento, reduzindo a precisão dos pesos da rede, permitindo rodada eficiente em *hardware* restrito.
- *Pruning* e regularização, diminuindo o número de parâmetros e simplificando arquiteturas para performance embarcada.
- Conversão para formatos otimizados, como *TensorFlow Lite* e *Open Neural Network Exchange* (ONXX), facilitando a implantação na borda.

A literatura nacional e internacional (BARBUTO et al., 2023; PIOLI et al., 2024; LAAKOM et al., 2024; TENG et al., 2025), demonstra ganhos expressivos no uso de IA embarcada para compressão: definição dinâmica das taxas de compressão, adaptação ao contexto operacional (ex.: prioridade para regiões de interesse em missões SAR) e manutenção da qualidade perceptual mesmo sob altas taxas de compactação. Trabalhos como Ramos et al. (2023) abordam a simulação e avaliação do uso de AE de aprendizado profundo para compressão de imagens em sistemas com múltiplos VANT, destacando ganhos de desempenho como aumento

de velocidade no processamento e envio de imagens comprimidas, além de boa acurácia e qualidade de reconstrução das imagens.

Ainda há desafios complexos: generalização dos modelos frente a múltiplos domínios de dados, estabilidade em *hardware* heterogêneo, e comparabilidade transparente com padrões clássicos. O trabalho atual busca superar parte dessas lacunas ao implementar, otimizar e avaliar experimentalmente diferentes arquiteturas de AE, analisando seu desempenho em ambientes simulados de nuvem e borda, e explorando o impacto de diferentes níveis de compressão, requisitos computacionais e potenciais aplicações práticas em contextos reais. A Figura 4, ilustra a arquitetura fundamental de um AE, destacando o processo de compressão da entrada via encoder, geração de uma representação latente (*bottleneck*) e subsequente reconstrução pelo decoder (CHOLLET, 2016).

Figura 4 – Estrutura básica de um *AE*



Fonte: Adaptado de CHOLLET, 2016.

Para uma melhor compreensão dos elementos que compõem os modelos utilizados neste trabalho de pesquisa, apresenta-se uma síntese conceitual dos elementos que compõem os autoencoders empregados neste estudo. O codificador é responsável por extrair características relevantes da imagem por meio de camadas convolucionais sucessivas, comprimindo a informação e reduzindo sua dimensionalidade (CHOLLET, 2016; OLIVEIRA et al., 2021). O decodificador realiza o processo inverso, reconstruindo a imagem original a partir da representação comprimida, operação dependente da qualidade do espaço latente, onde estão codificados os atributos essenciais que preservam a semântica do dado (LAAKOM et al., 2024; BERAHMAND et al., 2024). As funções de ativação, como *Rectified Linear Unit* (ReLU) e Sigmoid, introduzem não linearidade ao modelo, tornando possível a aprendizagem de relações complexas entre pixels (GOODFELLOW;

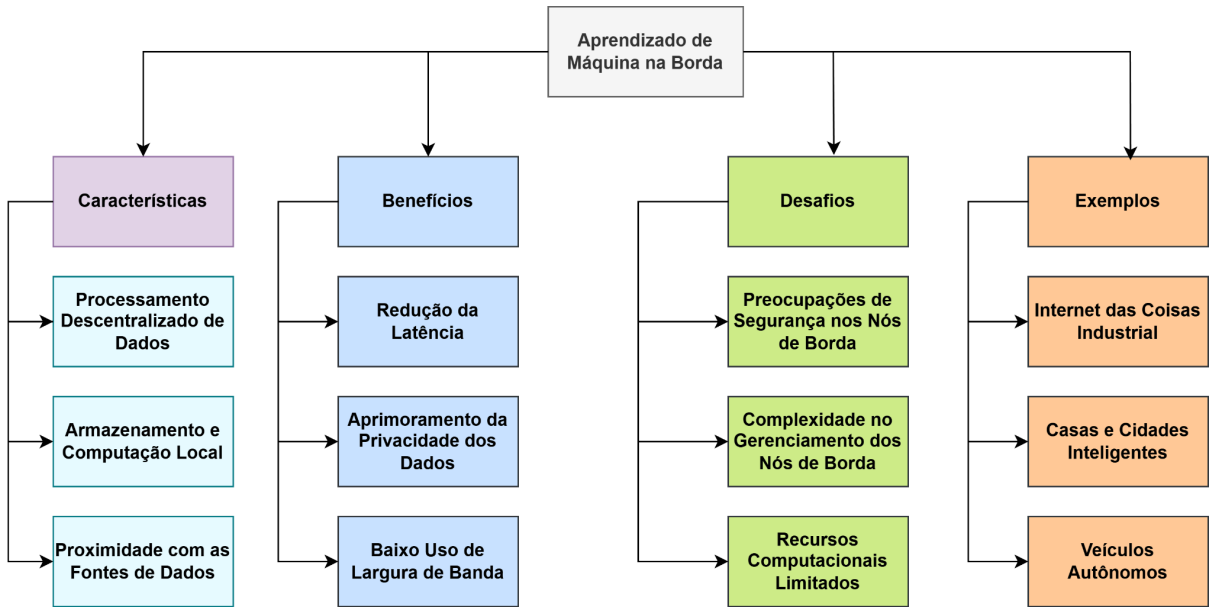
BENGIO; COURVILLE, 2016; CHOLLET, 2016). Estratégias de regularização, como penalização L1, *dropout* ou controle de redundância, reduzem sobreajuste e promovem generalização do modelo em cenários reais de inferência embarcada (AZIZIAN; BAJIĆ, 2024; OLIVEIRA et al., 2021). Por fim, métricas como MSE, PSNR, SSIM e MS-SSIM são amplamente utilizadas para mensurar fidelidade visual entre imagem original e reconstruída, equilibrando erro numérico e percepção estrutural, fator crucial em aplicações VANT/SAR onde a preservação de detalhes pode determinar o sucesso da operação (RAMOS et al., 2023; SINGH; GILL, 2023; ZHOU et al., 2019).

Considerando que a eficiência desses modelos depende diretamente do ambiente em que são executados, torna-se necessário compreender como a computação de borda afeta sua performance, escalabilidade e aplicabilidade. Nesse sentido, compreender tais dimensões auxilia não apenas na contextualização teórica do modelo adotado, mas também na interpretação dos resultados experimentais que serão apresentados no capítulo 4. Ao considerar a execução de modelos de inteligência artificial em dispositivos de borda é fundamental compreender as dimensões que influenciam o desempenho desses sistemas e o equilíbrio entre seus benefícios e desafios. Conforme apresentado por Reddi (2025), a Figura 5 sintetiza quatro eixos principais: características, benefícios, desafios e exemplos de aplicação.

Entre as características, destacam-se o processamento descentralizado de dados, o armazenamento e a computação locais e a proximidade das fontes de dados, que reduzem a dependência da nuvem. Dentre os benefícios, sobressaem-se a redução da latência, o aumento da privacidade dos dados e o menor uso de largura de banda, fatores essenciais para aplicações críticas. Em contrapartida, os desafios envolvem questões de segurança nos nós de borda, complexidade na gestão distribuída e limitações de recursos computacionais.

Por fim, a figura ilustra exemplos representativos, como Internet das Coisas Industrial, casas e cidades inteligentes e veículos autônomos, contextos em que a computação de borda possibilita maior autonomia e processamento local em tempo real.

Figura 5 – Dimensões do Aprendizado de Máquina na Borda



Fonte: Adaptado de REDDI (2025).

2.6. TRABALHOS RELACIONADOS

A redução e compressão de dados visuais em ambientes de VANT e borda computacional tem sido amplamente investigada devido às restrições de processamento e comunicação desses cenários. As propostas baseadas em AE se destacam por oferecer compactação aprendida e adaptativa, superando limitações de métodos tradicionais. Diversos estudos recentes exploram arquiteturas e abordagens variadas, cada qual contribuindo com diferentes perspectivas e desafios.

Ramos et al. (2023) analisam AE convolucionais e modelos profundos para compressão de imagens em redes multi-VANT em cenários de vigilância/SAR, enfatizando métricas de qualidade visual e aspectos de latência no sistema distribuído. O estudo evidencia ganhos no processamento distribuído com o uso de AE, porém não avalia arquiteturas com penalização explícita de redundância nem realiza otimizações estruturais voltadas a restrições de dispositivos de borda, lacunas abordadas diretamente neste trabalho de pesquisa.

O trabalho ARCog-NET proposto por Ramos et al. (2024), por sua vez apresenta uma arquitetura cognitiva avançada para enxames de VANT, integrando processamento distribuído no paradigma borda–névoa–nuvem e avaliando cenários realistas, como inspeção de turbinas eólicas e missões de monitoramento/SAR. Os autores demonstram reduções relevantes de latência e ganhos na eficiência

operacional do enxame a partir do particionamento adequado de funções entre os diferentes níveis da arquitetura. Porém, apesar de sua relevância para aplicações embarcadas, o ARCog-NET não foca em compressão visual baseada em AE, tampouco discute otimizações estruturais em gargalos latentes ou validação quantitativa da reconstrução, como proposto nesta dissertação. Dessa forma, ele se posiciona como um trabalho complementar, reforçando a importância de artefatos de compressão eficientes para o fluxo de dados dentro da arquitetura cognitiva.

Laakom et al. (2024), investigam AE penalizados por redundância no gargalo latente via termo de perda baseado em covariâncias pareadas, demonstrando que a redução de correlações resulta em representações mais compactas e informativas, com ganhos em reconstrução e classificação. Contudo, o estudo mantém foco teórico-experimental em *datasets* padrão (ex.: MNIST, CIFAR-10) e não valida tais modelos em cenários operacionais com restrições de VANT ou SAR, lacuna diretamente abordada nesta dissertação.

Marchenko et al. (2024) realizam uma análise abrangente de algoritmos de compressão de imagens via redes neurais, comparando diferentes arquiteturas e funções de custo. Embora forneçam visão geral sobre desempenho em reconstrução, os autores não investigam restrições de *hardware* em ambientes de borda nem avaliam latência operacional, aspectos centrais desta pesquisa.

Bao et al. (2023) propõem um AE segmentado para compressão de imagens em redes de sensores sem fio (WSN), obtendo bons resultados de compactação em cenários com restrições de transmissão. Entretanto, a ênfase recai mais sobre eficiência energética do que sobre inferência embarcada ou latência operacional, aspectos centrais para aplicações com VANT como as deste estudo.

Oliveira et al. (2021) utilizam VAE de complexidade reduzida para compressão on-board de imagens de satélite, destacando sua viabilidade computacional em *hardware* embarcado e superando padrões como CCSDS 122.0-B. Embora o cenário seja análogo ao de VANT por restrições semelhantes, os autores não comparam arquiteturas AE penalizadas por redundância nem otimizações adicionais de AE convencionais, lacunas abordadas neste trabalho.

Yamazaki et al. (2022) investigam compressão neural de *deep features* guiada por otimização de taxa-distorção, contribuindo para o entendimento dos *trade-offs* perceptuais em *pipelines* neurais. Essa abordagem complementa os

objetivos deste estudo ao reforçar a relevância de métricas estruturais como SSIM e MS-SSIM em avaliações de qualidade.

Zhu (2024) compara diversas arquiteturas de AE e demonstra que modelos mais complexos tendem a alcançar maior fidelidade visual, reforçando discussões relevantes para a hipótese central desta pesquisa. Entretanto, o autor não examina a adequação dessas arquiteturas a *hardware* embarcado, ponto essencial desta dissertação

Bhagat et al. (2024) apresentam análises de AE para extração de *features* em classificação de imagens, demonstrando ganhos em reconhecimento visual. Contudo, o foco em tarefas de classificação em nuvem não aborda a compactação para ambientes restritos de borda, como os desta pesquisa.

Barbuto et al. (2023) discutem os desafios de integrar IA embarcada à computação de borda, enfatizando latência, carga de rede e eficiência energética em uma meta-revisão sistemática. Embora não realizem experimentos com AE, reforçam a necessidade de soluções enxutas para restrições reais, validando a motivação deste estudo.

Qiu et al. (2020) e Sulieman et al. (2022) investigam arquiteturas de borda para aplicações sensíveis a tempo, destacando questões de escalabilidade, alocação de tarefas e impacto da latência, aspectos essenciais ao contexto estudado nesta dissertação.

Teng et al. (2025) ampliam a compreensão sobre compressão baseada em deep learning para imagens de inspeção por VANT, avaliando estratégias de otimização e impacto em métricas perceptuais (PSNR, SSIM, MS-SSIM). Ainda assim, o contexto de inferência embarcada em VANT e comparações estruturais entre modelos complexos e otimizados não são explorados, aspectos centrais deste trabalho de pesquisa.

Pioli et al. (2024, 2025) fornecem revisões sistemáticas e *frameworks* conceituais sobre redução inteligente de dados e inteligência de borda, identificando limitações em validações experimentais de compressão neural em cenários reais ou próximos da operação. Essa lacuna é diretamente tratada neste trabalho.

Em suma, embora cada estudo contribua para o avanço da compressão inteligente de dados visuais, nenhum deles integra, de forma simultânea, três arquiteturas de AE, otimização estrutural, avaliação quantitativa multifatorial, metodologia DSR e execução simulada na borda, como realizado nesta dissertação.

Essa convergência reforça a originalidade, relevância prática e rigor científico do artefato proposto. A Tabela 1 apresenta uma síntese dos trabalhos relacionados identificando com um X, caso o estudo esteja explicitamente vinculado aos assuntos desta pesquisa, mostrando visualmente as semelhanças e diferenças para este trabalho.

Tabela 1 – Síntese dos trabalhos relacionados

Autor/Ano	DR	Borda	IA	DSR	AE	VANT/SAR	Ciência Aberta
Ramos et al. (2023)	X	X	X		X	X	X
Ramos et al. (2024)		X	X			X	X
Laakom et al. (2024)	X		X		X		X
Marchenko et al. (2024)	X		X		X		
Bao et al. (2023)	X	X	X		X		
Oliveira et al. (2021)	X	X	X		X		X
Yamazaki et al. (2022)	X		X		X		X
Zhu (2024)	X		X		X		
Bhagat et al. (2024)			X		X		X
Barbuto et al. (2023)	X	X	X				X
Qiu et al. (2020) Sulieman et al. (2022)	X	X	X				X
Teng et al. (2025)	X	X	X			X	
Pioli et al. (2024, 2025)	X	X					X
Este estudo	X	X	X	X	X	X	X

Fonte: elaborado pelo autor (2025).

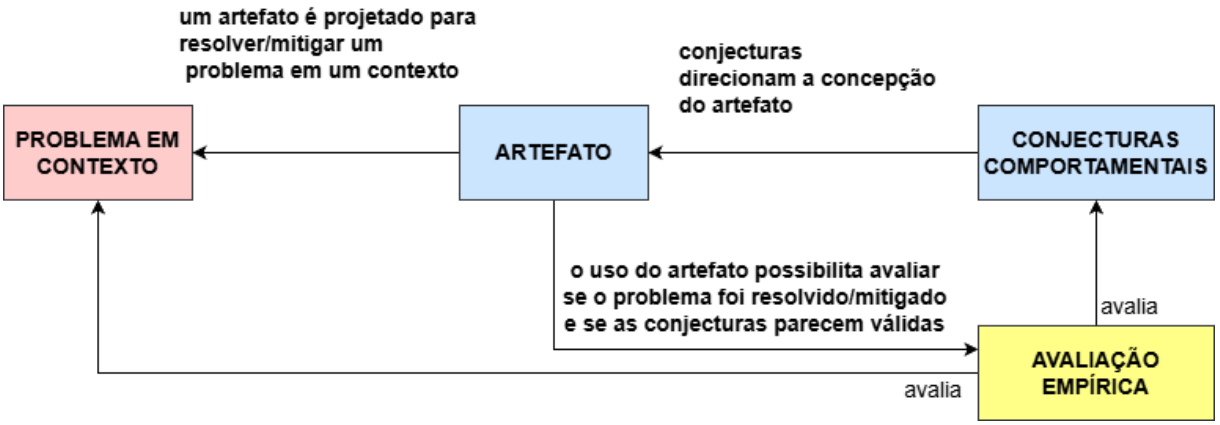
3. METODOLOGIA

3.1. FUNDAMENTAÇÃO E TIPO DE PESQUISA

A metodologia deste trabalho segue a abordagem DSR (HEVNER et al., 2004; PEFFERS et al., 2007), cuja essência está na construção e avaliação de artefatos tecnológicos que solucionam problemas relevantes de forma científica. Conforme Pimentel et al. (2020), a DSR propõe um ciclo iterativo de *design*, avaliação e reflexão, articulando rigor metodológico com relevância prática. Essa abordagem orienta a criação de artefatos que no caso desta pesquisa, modelos de AE para compressão de imagens que são avaliados quanto à sua efetividade e eficiência em contextos reais ou simulados de borda computacional.

O modelo de Pimentel et al. (2020) apresenta de forma clara os elementos centrais e as inter-relações entre o rigor científico, a relevância prática e o ciclo de projeto, que fundamentam o presente estudo. A Figura 6 ilustra esses elementos, destacando como a DSR se estrutura em torno da construção e avaliação de artefatos científicos aplicados a problemas reais.

Figura 6 – Elementos centrais do modelo-DSR



Fonte: Adaptado de PIMENTEL et al. (2020).

Diante dos objetivos propostos, esta pesquisa está estruturada em dois ciclos de DSR, buscando garantir tanto rigor metodológico quanto relevância prática. A abordagem segue as orientações metodológicas: identificação do problema, definição de objetivos, projeto e desenvolvimento, demonstração, avaliação e comunicação, proposta por Peffers et al. (2007) para a *Design Science Research Methodology* (DSRM).

O delineamento da pesquisa é classificado como experimental, com abordagem quantitativa e comparativa, fundamentado na reprodução de experimentos e na análise de métricas de desempenho. Para a sustentação teórica e a contextualização conceitual da relevância do tema, foi realizada uma pesquisa bibliográfica seletiva e crítica. Esta baseou-se em revisões sistemáticas sobre inteligência de borda e DR em sistemas distribuídos (BARBUTO et al., 2023; PIOLI et al., 2024), que destacam a necessidade de soluções eficientes e inteligentes de processamento em borda computacional. Além disso, referências específicas recentes de estudos experimentais sobre compressão e arquiteturas de AE foram utilizadas para embasar o desenho experimental, tais como: Ramos et al. (2023), Laakom et al. (2024), Zhu (2024) e Teng et al. (2025).

O artefato proposto neste trabalho é inovador porque integra modelos de AE para compressão de imagens, previamente validados em nuvem, a uma abordagem de otimização capaz de operar em dispositivos de borda com severas restrições de processamento, memória e energia. Por essa pesquisa, a literatura carece de validações experimentais que considerem a latência e o desempenho de tais modelos em ambientes reais de *borda* computacional, especialmente no contexto de imagens oriundas de dispositivos IoT. Essa integração, aliada à avaliação comparativa e à documentação de um *pipeline* replicável, busca responder às lacunas identificadas nas revisões do tema.

3.2. DSR CICLO 1 - AMBIENTE DE NUVEM

3.2.1. DEFINIÇÕES

A etapa inicial do ciclo seguiu uma abordagem de estudo secundário (KITCHENHAM et al., 2007), caracterizada pela análise, síntese e integração de resultados de pesquisas já publicadas, especialmente revisões sistemáticas, mapeamentos e artigos experimentais recentes como BARBUTO et al. (2023), RAMOS et al. (2023), PIOLI et al. (2024), LAAKOM et al. (2024), ZHU (2024) e TENG et al. (2025). O objetivo central foi reunir o conhecimento consolidado sobre compressão e DR visuais na borda, identificar tendências, lacunas e recomendações práticas presentes na literatura, e fundamentar o desenho e avaliação dos artefatos deste trabalho de pesquisa. A opção por um estudo secundário é justificada pela maturidade do campo e pelo grande volume de pesquisas já consolidadas,

permitindo uma análise crítica baseada em evidências direcionando esforços para aspectos experimentais.

Com base nos elementos centrais do modelo-DSR, proposto por Pimentel et al. (2020), para este ciclo 1, foram identificados Problema de Contexto, a Conjectura e os Artefatos, respectivamente.

- Quais são as versões dos modelos de AE convencional, variacional e de penalização por redundância, com melhor resultado nas métricas, PSNR, MS-SIM, SSIM e latência, de acordo com as imagens de entrada do *dataset* SARD2 (GEGENAVA, 2025)?
- O uso de um servidor virtual de nuvem do REPESQ com disponibilidade de GPU, é suficiente para treinar os modelos e produzir resultados satisfatórios.
- Modelos funcionais de AE*:* convencional, variacional e de penalização por redundância.

3.2.2. LATÊNCIA, QUALIDADE DE RECONSTRUÇÃO E MÉTRICAS DE AVALIAÇÃO

Com base no levantamento da literatura anterior, foram definidas as métricas a serem exploradas neste trabalho de pesquisa. No contexto de aplicações com VANT, foco deste estudo, a eficiência de transmissão de imagens sugere-se uma avaliação multifatorial.

A latência, segundo Tanenbaum et al. (2021), pode ser entendida como o atraso total entre o envio de uma requisição e o recebimento de sua resposta, sendo composta pelos tempos de processamento, transmissão, propagação e enfileiramento. Em contextos de inteligência de borda, ela corresponde ao tempo decorrido entre a entrada e a saída de um modelo executado na borda (REDDI, 2025).

Para avaliar o desempenho das arquiteturas de compressão, utilizam-se métricas objetivas como o PSNR, SSIM e o MS-SSIM, que mensuram, respectivamente, a intensidade do ruído introduzido, a similaridade estrutural entre imagem original e a reconstruída e a similaridade em múltiplas escalas. Estas fornecem uma avaliação mais robusta da qualidade da imagem (RAMOS et al., 2023). Por sua vez, a taxa de compressão (TC), definida como a razão entre o tamanho do arquivo original e do arquivo comprimido e o tempo de processamento

por imagem, também são parâmetros centrais em estudos comparativos (SUBBURAJ & BHAVANA, 2024; ZHU, 2024). A combinação dessas métricas permite uma análise multifatorial da eficiência de cada modelo em contextos práticos.

3.2.3. REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS LEVANTADOS

De acordo com os estudos analisados, foram definidos os requisitos funcionais e não funcionais para o *pipeline* de compressão a ser implementado, considerando restrições reais de processamento, energia e latência.

Fundamentado na análise de revisões e artigos recentes, o *pipeline* desenvolvido neste trabalho de pesquisa foi orientado pelos seguintes requisitos, que refletem desafios, tendências e demandas frequentemente apontados na literatura para soluções práticas em inteligência na borda. Foram levantados os seguintes requisitos, sendo os funcionais (RF), o que o sistema deve fazer e os não funcionais (RNF) como o sistema deve se comportar:

- (RF1) Receber e pré-processar imagens capturadas por dispositivos de borda;
- (RF2) Comprimir e reconstruir imagens por meio de modelos de *AE*;
- (RF3) Mensurar as métricas de latência, PSNR, SSIM, MS-SSIM e taxa de compressão;
- (RF4) Permitir adaptação dos modelos para ambientes com diferentes restrições de *hardware*;
- (RNF1) Executar em *hardware* com memória e processamento restritos ou simulados (ex: Raspberry Pi, placas ARM);
- (RNF2) Minimizar o consumo energético do *pipeline*;
- (RNF3) Documentação e replicabilidade do experimento, disponibilizando o código em repositório público;
- (RNF4) Flexibilidade para ajuste de parâmetros conforme o contexto experimental.

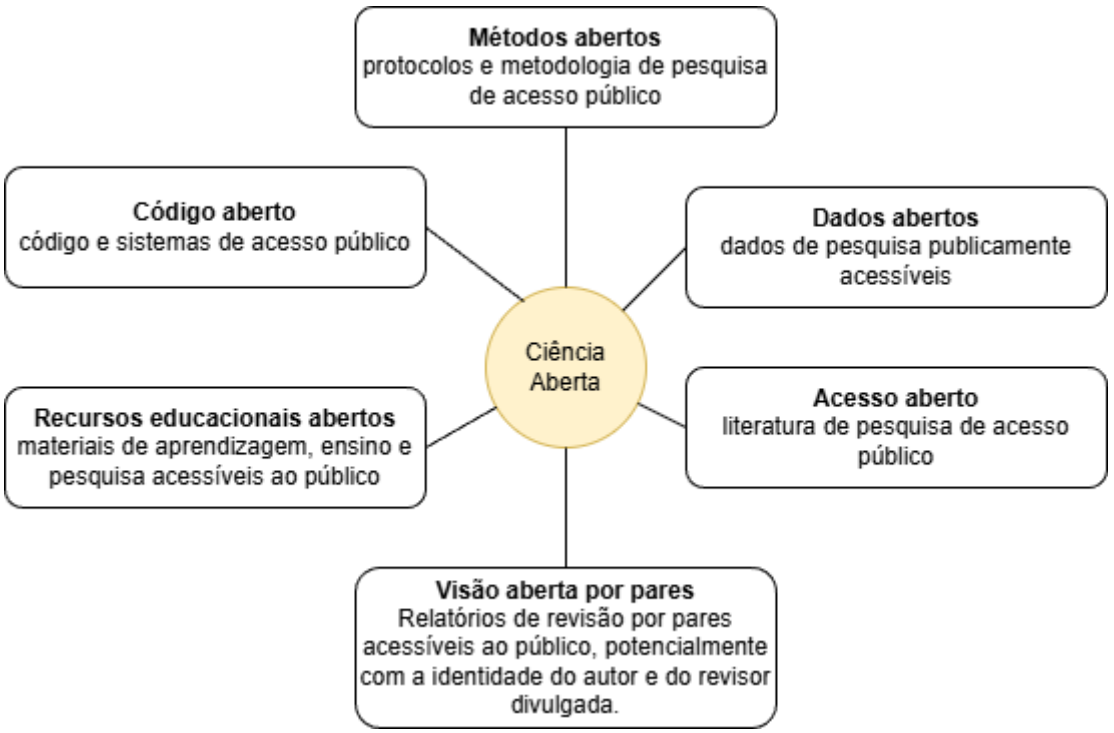
Tais requisitos asseguram que o artefato desenvolvido não só seja tecnicamente viável, mas também relevante para cenários práticos de *borda computacional*, atendendo a desafios identificados na literatura.

O requisito RNF3 foi identificado, porque alguns estudos encontrados não disponibilizam os modelos ou códigos fontes em repositórios públicos além disso vai ao encontro do que preconiza a ciência aberta que será detalhada a seguir.

A Ciência Aberta preconiza princípios como transparência, colaboração, inclusividade e acesso aberto em todas as etapas do processo científico. Isso inclui o compartilhamento aberto de dados, métodos, códigos e resultados, a adoção de revisões abertas por pares, o uso de repositórios digitais, a publicação em acesso aberto e a promoção da participação ampla de diferentes atores na pesquisa (BERTRAM et al., 2023; DEZHINA, 2023). A transparência é considerada um princípio central, orientando políticas e práticas para garantir maior acesso, responsabilidade acadêmica e reprodutibilidade dos resultados científicos (LEONELLI, 2023; ROMERO, 2025).

Além disso, ela busca superar desafios como vieses, falta de replicabilidade e competitividade excessiva, promovendo uma cultura de colaboração, diversidade, justiça e sustentabilidade na produção do conhecimento (DEZHINA, 2023). Organizações internacionais têm papel fundamental no desenvolvimento de políticas, infraestrutura e modelos de comunicação científica abertos, destacando a importância de dados abertos, revisão aberta, métodos transparentes e inclusão social (BERTRAM et al., 2023). Essas características são ilustradas na Figura 7.

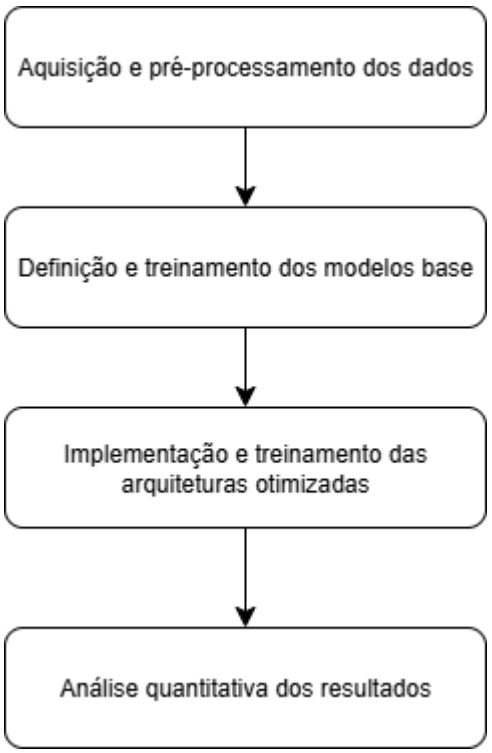
Figura 7 – Núcleo dos princípios da Ciência Aberta



Fonte: Adaptado de BERTRAM et al., 2023.

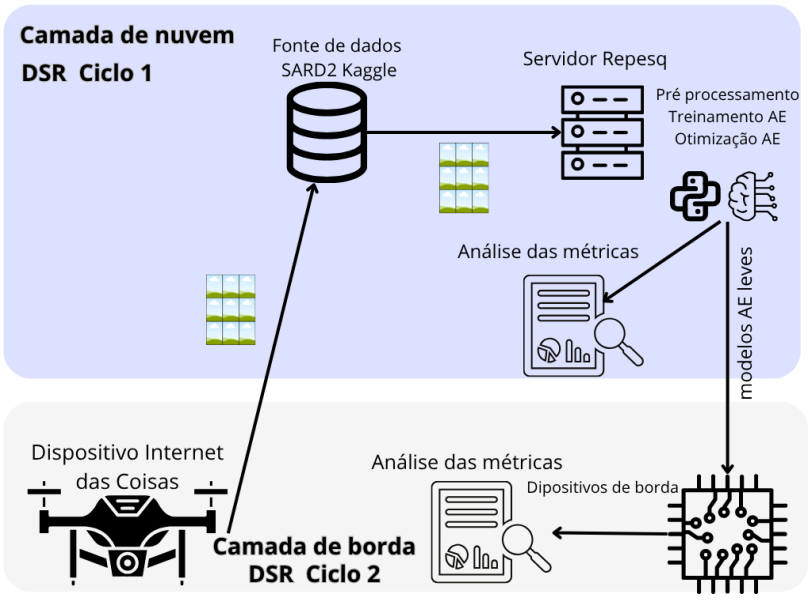
Com base nesses requisitos, foi desenhado o *pipeline* do ciclo 1 que corresponde ao tratamento inicial dos dados e treinamento dos modelos e otimizações que foram experimentados em um ambiente computacional de nuvem. As Figuras 8 e 9 ilustram o fluxo arquitetural completo da solução, detalhando as responsabilidades de cada camada, desde a aquisição na borda até o *deploy* final do modelo otimizado na borda computacional.

Figura 8 – *Pipeline* metodológico para avaliação comparativa das arquiteturas de AE para o ciclo 1



Fonte: elaborado pelo autor (2025).

Figura 9 – Arquitetura da Solução de Compressão AE em Ambientes Híbridos Borda-Nuvem



Fonte: elaborado pelo autor (2025).

As imagens utilizadas neste estudo foram obtidas do *dataset* SARD-2 (GEGENAVA, 2025), composto por registros aéreos de alta resolução (1920×1080 pixels) capturados por VANT em cenários simulados de busca e salvamento. O conjunto disponibiliza imagens organizadas em treino, validação e teste, totalizando 1.386 imagens na base de treinamento, 396 na validação e 196 no teste. A Figura 10, apresenta um exemplo típico de imagem do *dataset*, ilustrando a natureza aérea e o tipo de cena processado pelos modelos avaliados.

Figura 10 – Exemplo de imagem do *dataset* SARD-2 utilizado neste trabalho

Fonte: GEGENAVA (2025).

Após a aquisição dos dados, foram copiadas do ambiente Kaggle[2], foram para o servidor virtual da REPESQ[3] da Universidade Federal de Juiz de Fora (UFJF). Embora o próprio Kaggle, tenha um ambiente de desenvolvimento, assim como o Google Colab, onde o experimento é possível de ser realizado, a escolha do uso do deste servidor, se deu por limitação de tempo de uso das Unidades de Processamento Gráficos (GPUs) do ambiente gratuito do Kaggle, que fornece 30 horas semanais. Os modelos de AE foram codificados e otimizados dentro do REPESQ coletando as análises do ciclo 1 (nuvem) e, após esse processamento, o

[2] Disponível em: https://www.kaggle.com/datasets/nikolasgegenava/sard-2-search-and-rescue-dataset-extra-classes. Acesso em: 20 jun. 2025.
[3] Disponível em: https://www.repesq.ufjf.br/ Acesso em 03 set. 2025

modelo com melhor resultado foi transferido para o ambiente de borda, que representa microcontroladores com *software* embarcado para as análises do ciclo 2 (borda). Os detalhes de cada processo, são apresentados nos itens a seguir.

3.2.4. AMBIENTE COMPUTACIONAL DE NUVEM UTILIZADO

A implementação dos experimentos iniciais, contemplando o treinamento dos modelos e avaliação dos resultados, foi realizada em ambiente virtual de nuvem, dentro da REPESQ da UFJF, que possui maior capacidade computacional, em relação aos ambientes computacionais de borda, sendo exigência para o treinamento dos modelos. Essa escolha se deve também, pela indisponibilidade inicial de infraestrutura de borda com recursos computacionais e de rede restritos, como largura de banda limitada, latência variável e processamento local reduzido, características típicas desses ambientes (ZHANG et al., 2024). Assim, no ciclo 2 borda, a inferência será direcionada para ambientes de borda, visando eficiência computacional e redução de tráfego de dados, conforme abordado na literatura relacionada (YAMAZAKI et al., 2022; BAO et al., 2023; AZIZIAN e BAJIĆ, 2024).

Devido às limitações do ambiente de nuvem, como a impossibilidade de controlar a largura de banda e a latência de rede de forma realista, a latência real de transmissão não pôde ser mensurada; por isso, utilizou-se o tempo de compressão e reconstrução como *proxy* da latência. A Tabela 2 apresenta um resumo com as configurações do ambiente para facilitar a reprodutibilidade, de acordo com o que preconiza a Ciência Aberta, dos experimentos, as demais bibliotecas utilizadas (Tensorflow, Keras, Numpy, etc) podem ser consultadas diretamente no *notebook* disponível no Github (COTTA, 2025).

Tabela 2 – Configurações do ambiente de nuvem usado para treinar os modelos REPESQ

Sistema Operacional	**Processador (CPU)**	**Memória RAM**	**Armazenamento**	**GPU**	**Linguagem/Ambiente**
22.04.5 LTS x86_64	4 CPUs físicas e 4 CPUs lógicas	32 Gb	100gb	GPU: 1x NVIDIA A30 24 GBs VRAM	Python 3.10.12 / Jupyter Colab

Fonte: elaborado pelo autor com base nas configurações do servidor (2025).

3.2.5. AQUISIÇÃO E PRÉ PROCESSAMENTO DOS DADOS

Para a etapa inicial de aquisição e pré-processamento dos dados, a coleta dos dados foi realizada a partir do *dataset* SARD 2 “*Search and Rescue Dataset, Extra Classes*”, disponibilizado por Nikolas Gegenava (GEGENAVA, 2025), sob licença MIT, dentro do site Kaggle.

O *dataset* SARD-2 foi selecionado por atender de forma precisa aos requisitos deste estudo, que demanda imagens reais capturadas por VANT em cenários de SAR, alinhando-se diretamente ao contexto de aplicações em borda computacional. O conjunto oferece imagens de alta resolução (1920×1080 px), diversidade de ambientes e variações de movimento humano, permitindo analisar a compressão em condições próximas às enfrentadas por VANT em operações críticas. Além disso, sua distribuição sob licença aberta (MIT) viabiliza a reprodutibilidade dos experimentos e está em conformidade com os princípios de Ciência Aberta, adotados ao longo desta dissertação. O balanceamento entre qualidade, volume de dados e facilidade de pré-processamento torna o SARD-2 especialmente adequado para experimentos envolvendo treinamento em ambiente de nuvem e posterior execução em dispositivos de borda com recursos computacionais limitados.

Não foram considerados outros *datasets* para este estudo porque a pesquisa adotou critérios específicos de elegibilidade:

- alinhamento temático com imagens capturadas por VANT em cenários de SAR
- disponibilidade pública e licença aberta que permitisse reprodutibilidade
- quantidade de imagens adequada ao treinamento em ambiente de nuvem sem necessidade de particionamento adicional
- resolução suficiente para avaliar técnicas de compressão
- facilidade de pré-processamento para adaptação aos requisitos de *borda* computacional.

D*atasets* disponíveis na literatura focam em tarefas distintas, como detecção de objetos, vigilância urbana, segmentação semântica ou captura por câmeras terrestres, não atendendo às necessidades específicas deste estudo. Dessa forma,

o SARD-2 se mostrou o conjunto mais aderente ao problema investigado e suficientemente abrangente para os experimentos propostos.

Este conjunto de dados contém imagens de alta resolução capturadas por VANT em ambientes reais, com encenações simuladas de emergências, disponibilizadas em conjuntos de treino, validação e teste com múltiplas classes de movimento humano. O conjunto de dados coletados possui 1386 imagens na base de treinamento, 196 de teste e 396 de validação.

Para garantir consistência e comparabilidade entre as diferentes arquiteturas, foram selecionadas amostras balanceadas entre as classes e cada imagem foi redimensionada e normalizada conforme os requisitos dos modelos.

O *dataset* SARD 2 é composto por imagens em alta resolução (1920x1080 px), exigiu uma sequência de etapas de pré-processamento para adequação ao cenário de borda computacional com restrições de *hardware* e memória.

Primeiramente, as imagens foram redimensionadas (*downscaling*) para a dimensão de entrada de 128×128px (com 3 canais de cor), totalizando uma dimensão de entrada de (128×128×3). Esta escolha é fundamental para simular as severas restrições de sustentabilidade computacional, memória e energia esperadas em sistemas embarcados de VANT. Tal resolução é compatível com a faixa de dimensões adotadas em estudos de compressão aprendida e cenários restritivos (OLIVEIRA et al., 2021; LAAKOM et al., 2024; WANG et al., 2024) indicando redução de custo computacional sem comprometer de forma crítica a análise comparativa entre modelos. Testes exploratórios confirmaram a escolha de 128×128 px, pois resoluções maiores, como 256x256px, agravaram o consumo de memória no treinamento em nuvem e não proporcionaram ganhos de qualidade que justificassem o aumento da resolução das imagens.

Em seguida, o ordenamento dos canais de cor foi ajustado. As imagens lidas foram convertidas do padrão BGR (Azul, Verde, Vermelho), comum em bibliotecas de visão computacional, para o padrão RGB (Vermelho, Verde, Azul), que é o formato esperado pela maioria dos *frameworks de* aprendizado profundo (utilizando-se a função COLOR_BGR2RGB).

Por fim, os dados foram submetidos à Normalização (Min-Max). Este procedimento realizou-se pela divisão dos valores de pixel por 255. A normalização garante que os dados de entrada se distribuam uniformemente no intervalo [0,1], evitando a saturação das funções de ativação e o problema do *vanishing gradient,*

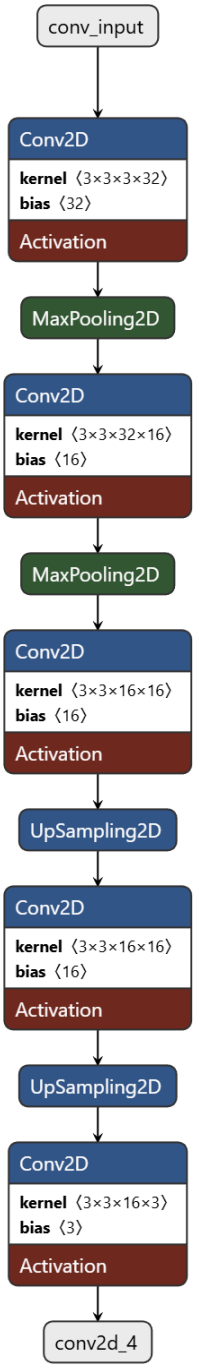
quando os gradientes das funções de ativação, como a sigmóide, tornam-se muito pequenos à medida que são propagados de volta pelas camadas, fazendo com que as atualizações dos pesos nas camadas iniciais sejam quase nulas (WANG et al. 2022). Como resultado, a rede aprende muito lentamente ou até para de aprender, dificultando o treinamento de redes profundas. Tal ajuste é crucial para garantir a estabilidade e a velocidade de convergência do treinamento da rede neural.

3.2.6. DEFINIÇÃO DOS MODELOS BASE

Três arquiteturas distintas de AE foram escolhidas e avaliadas: convencional, variacional e penalizada por redundância. Cada uma delas foi implementada a partir de princípios consolidados na literatura e ajustada para a tarefa de compressão de imagens VANT.

- AE Convencional: o modelo convencional é composto por um encoder com camadas convolucionais e operações de *max pooling*, responsáveis por reduzir progressivamente a dimensionalidade espacial enquanto preservam padrões estruturais relevantes. O *decoder* realiza o processo inverso por meio de *upsampling* e convoluções, reconstruindo a imagem a partir do mapa latente comprimido. A otimização é conduzida pela função de perda *Mean Squared Error* (MSE). A Figura 11 apresenta a arquitetura completa do modelo convencional

Figura 11 – Arquitetura do AE Convencional (Modelo Base)



Fonte: elaborado pelo autor usando o *software* Netron[4] (2025).

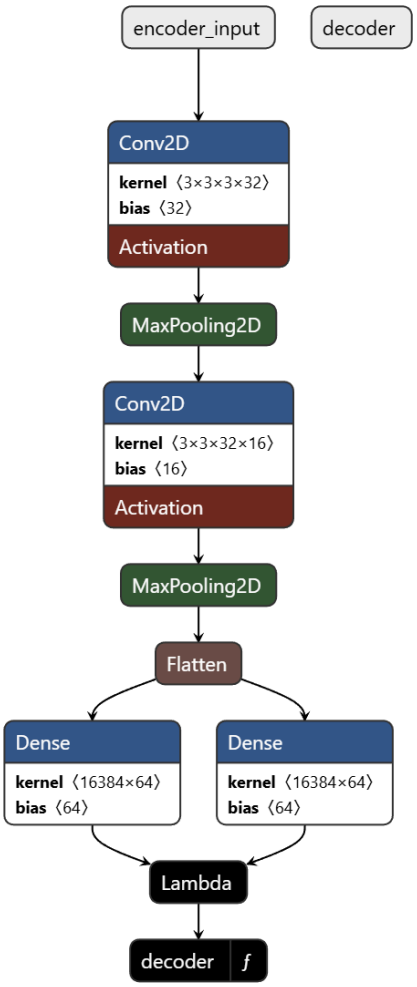
- VAE: a Figura 12 apresenta a arquitetura do VAE Base, utilizado neste trabalho. O *encoder* é composto por duas camadas convolucionais seguidas de operações de *max pooling*, responsáveis por reduzir a dimensionalidade espacial da

[4] Disponível em: https://github.com/lutzroeder/netron Acesso em: 10 out. 2025

imagem enquanto extraem características relevantes. Em seguida, o mapa de características é achatado (*flatten*) e projetado em duas camadas densas que estimam os parâmetros estatísticos do espaço latente: o vetor de médias μ e o vetor dos logaritmos das variâncias log σ^2

Esses dois vetores são combinados por meio de uma camada Lambda que implementa o *reparameterization trick*, permitindo a geração de um vetor latente z de forma diferenciável. Essa etapa é fundamental para que o VAE aprenda não apenas uma codificação comprimida, mas também uma distribuição contínua no espaço latente, favorecendo generalização e regularização. O vetor z resultante é então encaminhado ao *decoder*, que reconstrói a imagem a partir dessa representação comprimida. A figura evidencia claramente a separação conceitual entre *encoder*, espaço latente probabilístico e *decoder*, ressaltando o papel da amostragem estocástica na formação do vetor latente utilizado na reconstrução.

Figura 12 – Arquitetura do VAE (Modelo Base)



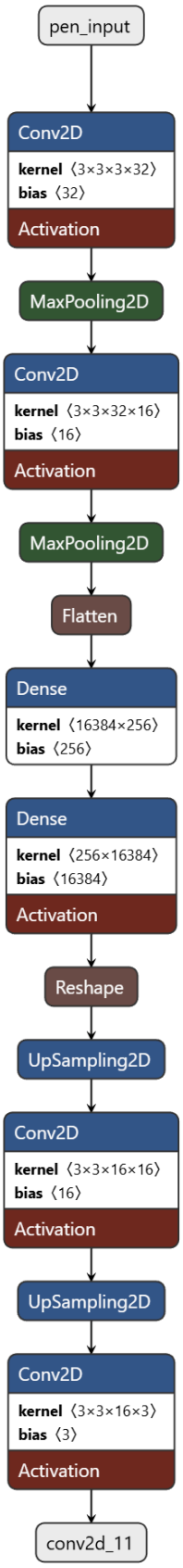
Fonte: elaborado pelo autor usando o *software* Netron (2025).

- AE Penalizado por Redundância: a Figura 13 apresenta a arquitetura do AE Penalizado por Redundância (Modelo Base) utilizado neste estudo. O encoder é composto por duas camadas convolucionais seguidas de operações de *max pooling*, responsáveis por reduzir progressivamente a resolução espacial da imagem enquanto preservam características estruturais relevantes. Após essa etapa, o mapa de características é achatado (*flatten*) e projetado em uma camada densa com 256 unidades, sobre a qual é aplicada uma regularização L1. Essa penalização induz esparsidade no vetor latente, estimulando o modelo a eliminar informações redundantes e a manter apenas as características mais relevantes para a reconstrução.

A etapa de decodificação inicia com uma camada densa de expansão, que reconstitui o volume latente para o formato espacial original do encoder (32×32×16). Em seguida, são aplicadas operações de *upsampling* e convoluções, restaurando gradualmente a dimensão espacial até atingir o formato final da imagem reconstruída. A última camada convolucional, com ativação *sigmoid*, produz a saída no domínio [0,1], adequada para representações normalizadas de intensidade.

A figura evidencia, portanto, o fluxo completo de compressão e reconstrução, destacando o papel central da penalização L1 em promover representações latentes mais compactas e eficientes, característica especialmente relevante para cenários de transmissão de dados em ambientes de borda com banda limitada

Figura 13 – Arquitetura do AE Penalizado por Redundância (Modelo Base)



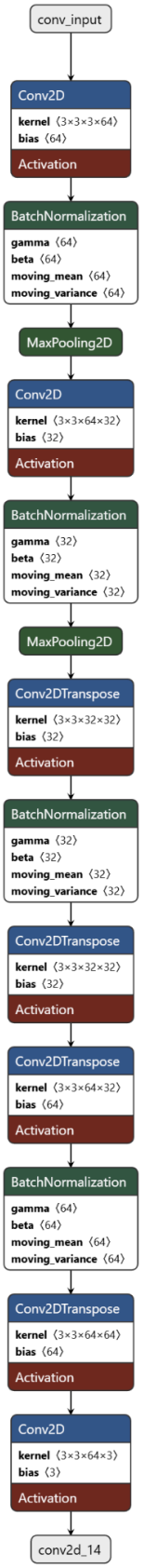
Fonte: elaborado pelo autor usando o *software* Netron (2025).

Após a avaliação das arquiteturas base observou-se a necessidade de aprimorar estabilidade, capacidade representacional e desempenho computacional. Dessa forma, foram desenvolvidas versões otimizadas das três arquiteturas, apresentadas na Seção 3.2.7.

3.2.7. ARQUITETURAS OTIMIZADAS

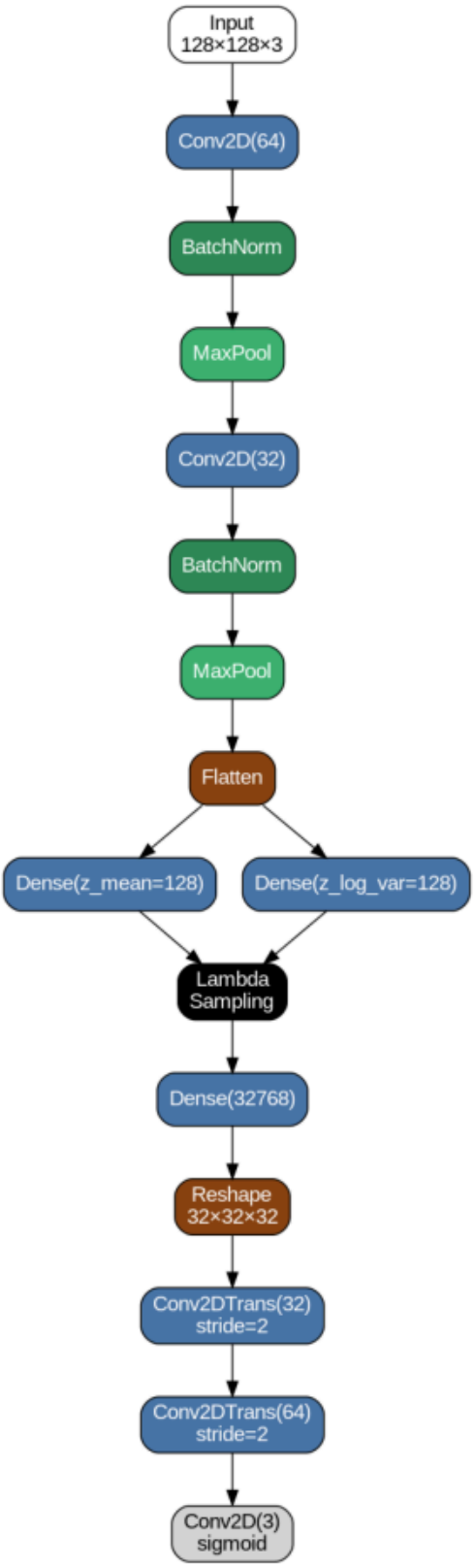
As Figuras 14, 15 e 16 apresentam, respectivamente, as versões otimizadas dos modelos Convencional, Variacional e Penalizado por Redundância. As otimizações incorporam ajustes estruturais como aumento de filtros, inclusão de camadas Batch Normalization, alteração da dimensão latente e substituição de operações de *upsampling*, de modo a melhorar estabilidade de treinamento e qualidade de reconstrução. Mais detalhes das otimizações são abordados na seção 4.1.1.

Figura 14 – Arquitetura do AE Convencional (Modelo Otimizado)



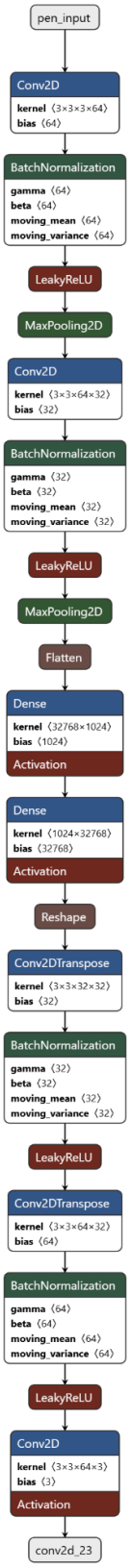
Fonte: elaborado pelo autor usando o *software* Netron (2025).

Figura 15 – Arquitetura do VAE (Modelo Otimizado)



Fonte: elaborado pelo autor (2025).

Figura 16 – Arquitetura do do AE Penalizado por Redundância (Modelo Otimizado)



Fonte: elaborado pelo autor usando o *software* Netron (2025).

3.2.8. IMPLEMENTAÇÃO E TREINAMENTO DOS MODELOS

Os experimentos foram implementados em linguagem de programação Python, utilizando o *Jupyter Lab* e *frameworks* como TensorFlow e Keras, simulando cenários compatíveis com borda computacional e adaptando arquiteturas conforme as restrições identificadas.

Foram implementadas as arquiteturas de AE (convencional, VAE e com penalização de redundância latente), treinamento até convergência e avaliação experimental. Os hiperparâmetros iniciais estão detalhados na Tabela 3. Os resultados são analisados com base em métricas encontradas na literatura, taxa de compressão, PSNR, SSIM, MS-SSIM e latência de processamento, e apresentados em gráficos e tabelas para comparação quantitativa entre as arquiteturas no próximo item. Para a reprodutibilidade completa do estudo, a implementação detalhada dos modelos, incluindo a configuração de todos os hiperparâmetros comuns (ex: *batch size*, otimizador e número de épocas), está disponível no código fonte público (COTTA, 2025).

Tabela 3 – Hiperparâmetros iniciais usado para treinar os modelos

Hiperparâmetro	Convencional	VAE	Penalização por redundância
Formato da entrada	128x128x3	128x128x3	128x128x3
Arquitetura codificador	Conv2D 32 -> Pool Máximo-> Conv2D 16 -> Pool Máximo->Conv2D 16	Conv2D 32 -> Pool Máximo-> Conv2D 16 -> Pool Máximo-> Achatar	Conv2D 32 -> Pool Máximo-> Conv2D 16 -> Pool Máximo
Camadas extras	Aumento de resolução-> Conv2D 16 -> Aumento de resolução	Achatar-> Densa (u. logvar) -> Lambda (amostragem)	Achatar ->Densa 256 (L1 = 10^{-5})
Espaço Latente	Gargalo Espacial (implícito)	Densa 64 + Amostragem estocástica	Densa 256 com L1
Decodificador	Aumento de resolução-> Conv2D 16 -> Aumento de resolução-> Conv2D 32	Densa -> Remodelar -> Aumento de resolução->Conv2D 32	Densa -> Remodelar -> Aumento de resolução ->Conv2D 32
Função de ativação	ReLU (intermediária) + Sigmóide (saída)	ReLU (intermediária) + Sigmóide (saída)	ReLU (intermediária) + Sigmóide (saída)
Regularização	-	-	L1 (10^{-5}) na camada Densa
Dimensão Latente/ Penalização	Implícito via agrupamento	64	256
Otimizador	Estimativa de Momento Adaptativa (lr=10^{-3})	Estimativa de Momento Adaptativa (lr=10^{-3})	Estimativa de Momento Adaptativa (lr=10^{-3})
Função de perda	Erro Quadrático Médio	Erro Quadrático Médio	Erro Quadrático Médio

Fonte: elaborado pelo autor (2025).

A Tabela 3 apresenta um resumo de hiperparâmetros adotados para cada uma das arquiteturas iniciais de AE avaliadas neste trabalho. Para o *encoder*, utilizaram-se duas camadas convolucionais seguidas de operações de *MaxPooling*, enquanto no *decoder* a reconstrução ocorre através de camadas de *Upsampling* e Conv2D, estratégia que favorece a extração e recuperação de características visuais essenciais. O espaço latente, que sintetiza a informação comprimida, é modelado de forma implícita no AE convencional (via *pooling*), explicitamente em uma camada *Dense* com 64 neurônios no VAE (acompanhada de amostragem estocástica) e com

256 neurônios no modelo penalizado, o qual também incorpora regularização L1 (10^{-5}) para induzir esparsidade. As funções de ativação empregadas foram ReLU nas camadas intermediárias e Sigmoid na camada de saída, adequadas para imagens normalizadas. Para a otimização, foi utilizado o algoritmo *Adaptive Moment Estimation (*Adam) com taxa de aprendizado de 10^{-3} e a função de perda MSE (erro quadrático médio), que mede a diferença global entre as imagens original e reconstruída.

A escolha de hiperparâmetros como taxa de aprendizado, tamanho do lote, número de camadas e tamanho das entradas deve ser guiada por critérios empíricos, considerando limitações computacionais e o objetivo de evitar *overfitting*, que é um problema comum em aprendizado de máquina em que um modelo se ajusta de forma excessivamente precisa aos dados de treinamento, em vez de aprender os padrões e relacionamentos subjacentes aos dados, ele memoriza o ruído e as flutuações irrelevantes, levando a um mau desempenho com novos dados.

Parâmetros como *batch sizes,* variam comumente entre 32 e 256 e taxas de aprendizado entre 0,01 e 0,0001. Além disso, não há um conjunto universal de hiperparâmetros ideais, e o ajuste fino é geralmente feito empiricamente para cada tarefa (GOODFELLOW et al., 2016). Todos os modelos foram treinados até a convergência da função de perda, utilizando o otimizador Adam e *early stopping* para evitar o *overfitting*.

Os hiperparâmetros adotados, como taxa de aprendizado entre 10^{-2} e 10^{-5}, tamanhos de *batch* variando de 16 a 256, otimizador Adam, função de ativação ReLU e dimensões de entrada, seguem recomendações amplamente reconhecidas na literatura contemporânea (LAAKOM et al., 2024; ZHU, 2024).

A escolha do Adam justifica-se mesmo em cenários com grandes volumes de dados, devido à sua eficiência de convergência em redes profundas e sua robustez em ambientes não convexos, conforme demonstrado em aplicações de engenharia preditiva e séries temporais complexas (Chen et al., 2025; Bouhanch, 2025).

Já a função de ativação ReLU permanece uma escolha padrão em autoencoders convolucionais, inclusive em modelos aplicados a ambientes industriais e sistemas de diagnóstico embarcados, por sua estabilidade numérica, simplicidade e eficiência computacional (Chae et al., 2025).

Essas escolhas foram ajustadas com base nas restrições computacionais disponíveis e nos objetivos experimentais, promovendo uma boa relação entre desempenho, tempo de treinamento e reprodutibilidade.

3.2.9. AMBIENTE COMPUTACIONAL DE BORDA SIMULADO

A tabela 4 abaixo, apresenta os parâmetros que foram utilizados para simulação de um ambiente computacional de borda. Esses também podem ser verificados no repositório público[5] deste trabalho de pesquisa. O desafio não é só o modelo, mas a pilha de *software* de implantação (*deployment stack*). A otimização ONXX/OpenVINO foi crucial, conforme apresentado no capítulo de resultados.

Tabela 4 – Ambiente computacional de borda simulado com OPENVINO[6]

Parâmetro	Valor Simulado	Descrição/Efeito
Dispositivo	CPU (ou GPU / MYRIAD)	Simula execução em *hardware* de borda (ARM, CPU etc.)
Precisão numérica	FP32, pois FP16 falhou	Reduz memória e latência com leve perda de precisão
Resolução de entrada	128×128×3	Compatível com restrições de VANT e sistemas embarcados
Latência simulada	1.2–2.5× da latência base	Adiciona jitter para simular carga variável
Energia estimada	lat_ms × 0.0025 mJ	Proporcional à latência (restrição energética)
Aquecimento	5 execuções	Estabiliza caches antes da medição
Execuções	30 rodadas	Calcula média, p95 e mínima da latência
Normalização	[0,1] (Min–Max)	Evita saturação e acelera convergência
Log de resultados	resultados_edge.csv	Registra métricas PSNR, SSIM, MS-SSIM e latência

Fonte: elaborado pelo autor (2025).

[5] Disponível em https://github.com/samuelccotta/sar_autoencoders. Acesso em 18 out. 2025.
[6] Disponível em: https://github.com/openvinotoolkit/openvino Acesso em: 12 jan. 2025

As análises quantitativas dos resultados serão discutidos no capítulo 4 à luz das lacunas apontadas pela literatura, subsidiando a definição de oportunidades para o ciclo seguinte.

3.3. DSR CICLO 2 - AMBIENTE DE BORDA

Com base nas limitações e oportunidades do ciclo anterior, foram ajustados os requisitos para refletir as restrições mais severas de dispositivos de borda, incluindo consumo energético, memória e capacidade de processamento.

Os modelos foram otimizados para execução local em ambientes de borda usando OpenVINO e o formato ONNX, empregando técnicas como compressão de modelos, quantização e arquiteturas mais leves, com o objetivo de minimizar a latência e viabilizar o processamento embarcado. Essas otimizações tornam-se necessárias porque dispositivos de borda possuem restrições severas de memória, processamento e energia, o que limita a execução eficiente de modelos tradicionais de AE (KONG et al., 2022; HAMDAN et al., 2020). A conversão para ONNX, aliada ao uso de quantização, podas e arquiteturas compactas, reduz significativamente o tamanho do modelo e o custo computacional, diminuindo o tempo de inferência e permitindo processamento em tempo quase real sem dependência da nuvem (LAAKOM et al., 2024; ZHU, 2024; REDDI et al., 2025). Além disso, em cenários críticos como VANT/SAR, onde há instabilidade de comunicação e decisões rápidas são essenciais, a execução local contribui para maior resiliência, autonomia operacional e confiabilidade do sistema.

Desta forma**,** uma nova rodada experimental foi conduzida, buscando aferir, sempre que possível, a latência eficiência dos modelos embarcados, além das métricas convencionais.

Os resultados do ciclo otimizado foram comparados com os do ciclo inicial e confrontados com recomendações da literatura, buscando evidenciar ganhos, limitações e implicações práticas para aplicações reais.

Todo o *pipeline* experimental, configurações de modelos e código-fonte foram devidamente documentados e publicados em repositório público[7], conforme prática recomendada em DSR e Ciência Aberta, para fomentar a replicação e extensão por outros pesquisadores (COTTA, 2025).

[7] Disponível em https://github.com/samuelccotta/sar_autoencoders. Acesso em 18 out. 2025.

4. RESULTADOS EXPERIMENTAIS

4.1. CICLO 1 - AMBIENTE DE NUVEM

4.1.1. AVALIAÇÃO EMPÍRICA

Os experimentos realizados, confirmaram a conjectura e permitiram avaliar os artefatos, que para este estudo são os modelos de AE, na tarefa de compressão e reconstrução de imagens provenientes de VANT em cenários de SAR. Os resultados quantitativos das métricas analisadas, são apresentados na Tabela 5, sendo ilustrados comparativamente na Figura 11. Esses indicadores foram escolhidos não só pelo valor técnico, mas pela relevância direta para missões reais de SAR, em que cada milissegundo de latência e cada ganho de fidelidade visual podem significar uma resposta mais rápida e eficaz em campo. Os resultados dos modelos iniciais (Base), indicaram necessidade de melhorias e otimizações até se chegar ao modelo final que, os resultados serão apresentados a seguir.

Tabela 5 – Resultados Comparativos para as arquiteturas avaliadas

Modelo	**PSNR**	**SSIM**	**MS-SSIM**	**Tempo (s)**	**Resumo de parâmetros de Otimização**
Convencional Base	17.66	0.54	0.86	0.25	---
VAE Base	15.01	0.13	0.44	0.26	---
Redundância Base	12.28	0.06	0.21	0.27	---
Convencional Otimizado FINAL	20.71	0.80	0.93	0.26	*filter_max*: 64 *Batch Normalization* *kernel_regularize*r = l2(10^{-4}) *Conv2DTranspose* Arquitetura assimétrica
VAE Otimizado FINAL	13.40	0.07	0.26	0.28	beta=0.01, L=128
Redundância Otimizado FINAL	14.21	0.09	0.33	0.28	l1_reg=10^{-6}, L=512

Fonte: elaborado pelo autor (2025).

A diferença de tempo observada no VAE pode ser explicada pela própria estrutura probabilística desse modelo. Diferentemente do AE convencional, o VAE precisa estimar os vetores de média (μ) e desvio padrão (σ) e realizar a etapa de *latent sampling* por meio do *reparameterization trick*, conforme apresentado por CHEN et al. (2020), YU et al. (2021) e OLIVEIRA et al. (2021). Essa operação envolve cálculos adicionais e a criação de tensores intermediários, elevando o custo computacional do encoder. Além disso, o VAE possui duas projeções latentes (μ e σ), aumentando o número de operações em comparação ao AE convencional, como também destacado por BERAHMAND et al. (2024). Dessa forma, a maior latência observada para o VAE é coerente com as diferenças estruturais da arquitetura.

As otimizações dos modelos se concentraram em três pilares: capacidade/estabilidade, regularização estrutural e ajuste do espaço latente. O parâmetro *filter_max*: 64 dobrou a capacidade do *encoder* para extrair características da imagem. A introdução de *BatchNormalization* (normalização em lotes) e *kernel_regularizer*=l2(10^{-4}) aumentou a estabilidade do treinamento e preveniu o *overfitting*. No *decoder*, o uso de *Conv2DTranspose* em uma arquitetura assimétrica (*encoder* e *decoder* com profundidades diferentes), permitiu a reconstrução de imagens com maior fidelidade.

Nos modelos específicos, o ajuste de beta=0,01 no VAE priorizou a qualidade de reconstrução sobre a suavidade do espaço latente, enquanto a combinação de um baixo fator de penalidade l1_reg=10^{-6} com uma grande Dimensão Latente (L=512) no modelo de Redundância foi necessária para tentar manter alguma qualidade de imagem, mitigando a perda de informação imposta pela penalidade de redundância.

No treinamento o aumento do número de épocas para 800 foi o mais assertivo para os modelos, foram utilizados os *callbacks Learning Rate Scheduler e EarlyStopping* para ajustar dinamicamente a taxa de aprendizado e interromper o treinamento quando não houvesse mais melhoria na validação, prevenindo *overfitting* e otimizando o tempo computacional.

Esses hiperparâmetros foram ajustados com base em suas funções específicas em cada arquitetura: no VAE, o parâmetro beta controla o equilíbrio entre a fidelidade da reconstrução e a regularização do espaço latente (quanto maior o beta, mais suave e generalizado o espaço, porém com perda de detalhes visuais). Já a dimensão latente (L) define a capacidade de representação do *bottleneck,*

valores maiores tendem a capturar mais variabilidade, mas aumentam o custo computacional.

Por fim, no modelo com penalização de redundância, o coeficiente “l1_reg” atua como fator de esparsidade, induzindo o modelo a eliminar redundâncias no vetor latente. Esses ajustes visam encontrar o ponto de equilíbrio entre qualidade perceptiva e eficiência de compressão, conforme observa-se nas Tabelas 5 e 7. A tabela 6, demonstra os valores de perda de cada modelo.

Tabela 6 – *Loss* final de cada modelo

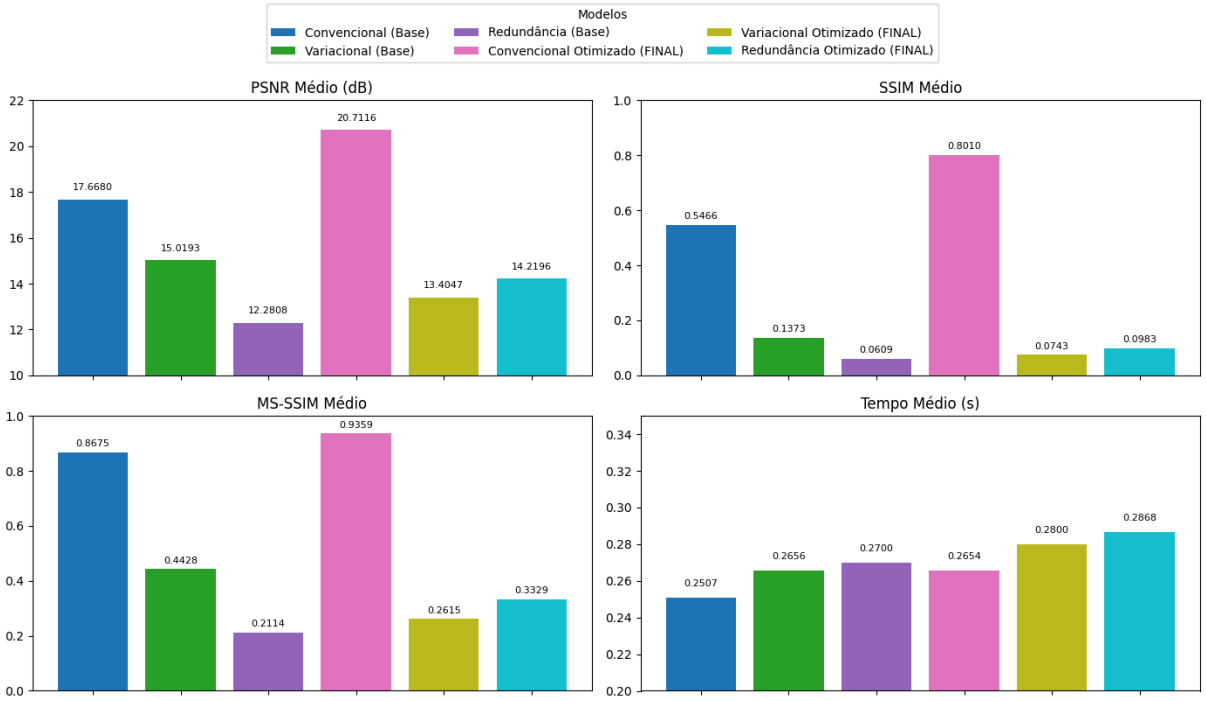
Modelo	**Loss final (treino)**
Convencional Base	≈ 0,019
Convencional Otimizado	≈ 0,009
VAE Base	≈ 0,028
VAE Otimizado	≈ 500
Redundância Base	≈ 0,059
Redundância Otimizado	≈ 0,037

Fonte: elaborado pelo autor (2025).

Os valores de SSIM dos modelos base, em especial o SSIM ≈ 0,06 da arquitetura com penalização de redundância, são compatíveis com reconstruções quase aleatórias e indicam que esses modelos não aprenderam uma representação latente útil para o *dataset*, caracterizando falha de treinamento e/ou configuração excessivamente compressiva do gargalo. Isso torna essencial documentar explicitamente a verificação de convergência: a loss dos modelos base estaciona em patamares relativamente altos (por volta de 0,028 no Convencional, ≈0,032 no Variacional e ≈0,038 no Redundância), enquanto o Convencional Otimizado alcança *loss* em torno de 0,009, o que explica o salto de SSIM para a faixa de 0,8 e confirma que a iteração do ciclo DSR corrigiu o problema. As curvas de treinamento mostram que nos modelos base a perda desce rápido e depois se estabiliza com diferença pequena entre treino e validação, sugerindo um regime de *underfitting* (capacidade insuficiente ou gargalo muito restrito) mais do que *overfitting* clássico, ao passo que nos modelos otimizados as curvas continuam decrescendo até um platô mais baixo, com boa sobreposição entre loss de treino e validação, o que sustenta o argumento

de que houve convergência adequada apenas após as otimizações estruturais e de regularização aplicadas no ciclo 1. Os *plots* de cada histórico de treinamento estão disponíveis no repositório público deste estudo (COTTA, 2025).

Figura 17 – Comparação gráfica das métricas entre os modelos base e otimizados



Fonte: elaborado pelo autor (2025).

No experimento apresentado na Figura 17, cada modelo foi executado uma única vez por imagem do conjunto de teste, totalizando *N* execuções, onde *N* corresponde ao número de amostras da base de teste. Como a latência é medida individualmente por imagem, o valor reportado na figura representa a média agregada sobre todas essas execuções, o que fornece uma estimativa estável do comportamento do modelo em condições normais de avaliação.

A robustez dessa média decorre do fato de que a avaliação é realizada sobre um conjunto extenso de amostras, o que reduz a influência de variações pontuais. Além disso, os experimentos foram conduzidos em um ambiente controlado (mesmo *hardware*, mesma carga de GPU/CPU, ausência de outros processos interferindo), minimizando a variabilidade externa.

Optou-se por não apresentar o desvio padrão porque, neste contexto específico, a latência de inferência dos modelos convolucionais é quase determinística, com baixa variabilidade entre execuções quando comparada à

variabilidade entre arquiteturas. Estudos anteriores sobre compressão neural e avaliação em borda mostram comportamento similar, com variação mínima entre execuções sucessivas em *hardware* dedicado (KONG et al., 2022; LAAKOM et al., 2024).

Observa-se que, para o modelo Convencional Otimizado FINAL, o MS-SSIM (0,93) supera o SSIM (0,80), invertendo a relação usual MS-SSIM ≤ SSIM. Essa inversão, embora contraintuitiva, é coerente com imagens SAR ruidosas e *downscaling* agressivo: o MS-SSIM, que pondera múltiplas escalas de análise, captura melhor a preservação de texturas speckle em escalas finas, enquanto o SSIM (escala única) penaliza perdas de bordas globais. No Ciclo 2 (borda), a relação se normaliza ligeiramente (SSIM=0,82; MS-SSIM=0,96), confirmando que a otimização OpenVINO preserva estruturas multi-escala mesmo sob restrições computacionais.

A avaliação comparativa dos modelos de AE revelou diferenças significativas entre as arquiteturas-base e suas versões otimizadas, especialmente em termos de qualidade de reconstrução e eficiência de processamento. Vale destacar, que antes de se chegar aos resultados, foram testados os seguintes parâmetros que foram descartados dos modelos VAE e Redundância, conforme exposto na Tabela 7. Além disso, testes com a função de perda combinada (MSE + SSIM) e o uso de (Keras Tuner/Optuna) foram explorados na busca de melhores hiperparâmetros, mas não foi considerado na análise final, devido a dificuldades de salvamento e carregamento dos modelos VAE para esse contexto e *dataset*. Todos os resultados estão no repositório do experimento dentro da pasta *results*.

Tabela 7 – Resultado dos parâmetros testados nos modelos VAE e de Redundância descartados.

Modelo	PSNR	SSIM	MS-SSIM	Tempo (s)	Resumo parâmetros de Otimização
VAE Otimizado	14.8372	0.1244	0.4182	0.2738	beta=0.1,L=128
VAE Otimizado	13.3675	0.0746	0.2604	0.2797	beta=0.01,L=256
Redundância Otimizado	14.3642	0.1052	0.3515	0.2743	l1_reg=10^{-6},L=512
Redundância Otimizado	12.2730	0.0605	0.2110	0.2764	l1_reg=10^{-7},L=512
Redundância Otimizado	14.2196	0.0983	0.3329	0.2868	l1_reg=10^{-6},L=1024

Fonte: elaborado pelo autor (2025).

As métricas de tempo foram obtidas a partir da média de 50 execuções consecutivas, após estabilização das rotinas TensorFlow. Assim, eventuais variações transitórias (*jitter*) foram amortecidas, garantindo consistência estatística entre os valores comparados.

4.1.2. ANÁLISE EM QUALIDADE DE RECONSTRUÇÃO

O desempenho em qualidade de imagem é dominado pela arquitetura Convencional Otimizada FINAL. O modelo alcançou os melhores resultados em todas as métricas de qualidade: PSNR: 20,7116 dB (significativamente superior aos demais), SSIM: 0,8010. MS-SSIM: 0,9359.

O valor de PSNR = 20,71 dB, obtido para o modelo Convencional Otimizado Final, está dentro da faixa esperada para reconstrução de imagens SAR do conjunto SARD-2, caracterizadas por alto ruído de speckle e contraste não linear. Estudos recentes em compressão e reconstrução de imagens SAR com AE relatam valores de PSNR tipicamente na faixa de 22 dB a 24 dB para arquiteturas otimizadas, como demonstrado por Cardona-Mesa et al. (2025), que analisaram cerca de 240 arquiteturas de autoencoders e identificaram desempenhos representativos nessa

faixa para reconstrução e redução de *speckle* em imagens SAR.

Para imagens de satélite ópticas, Sri et al. (2025) reportaram desempenho médio superior em PSNR, em torno de 25 dB, com SSIM moderado, refletindo a menor presença de ruído e a maior linearidade de contraste nesses dados em comparação com SAR. Portanto, embora o PSNR obtido para SAR seja numericamente inferior ao de imagens ópticas, ele representa uma reconstrução estruturalmente satisfatória, especialmente quando corroborado pelos índices elevados de SSIM (0,8010) e MS-SSIM (0,9359), que indicam preservação perceptual significativa mesmo sob restrições de tempo real e processamento embarcado.

Em contraste, os modelos VAE e Redundância (tanto nas versões base quanto nas otimizadas) apresentaram uma qualidade de reconstrução notavelmente inferior. O modelo Redundância (Base) obteve o desempenho inferior, com PSNR de apenas 12,2808 dB, indicando que, embora a penalização da redundância no manifold latente reduza correlações, ela comprometeu severamente a fidelidade da reconstrução neste cenário. A otimização implementada na arquitetura Convencional (aumentando o PSNR de 17,6680 dB para 20,7116 dB) foi a que apresentou o ganho de qualidade mais expressivo entre os modelos avaliados.

Ainda assim, para aplicações embarcadas em VANTs, valores de PSNR abaixo de aproximadamente 25 dB tendem a indicar perda perceptível de qualidade, de modo que o desempenho atual deve ser considerado preliminar para esse contexto. A qualidade obtida é aceitável para inspeção humana exploratória, especialmente quando combinada com índices estruturais como SSIM e MS-SSIM, mas há margem clara para evolução em estudos complementares, por exemplo, mitigando efeitos do *downscaling* e refinando a arquitetura e os hiperparâmetros.

Trabalhos conduzidos em condições distintas de *dataset*, resolução e taxa de *bits* reportam PSNR absolutos mais elevados, por exemplo, ≈40 dB em cenários específicos (RAMOS et al. 2023). No presente estudo, o objetivo não é maximizar PSNR absoluto, mas avaliar o equilíbrio entre qualidade e custo computacional sob restrições típicas de borda (resolução 128×128, forte compressão, inferência com foco em latência). Assim, a contribuição reside no ganho relativo entre arquiteturas dentro do mesmo protocolo experimental e na manutenção de latência competitiva, fatores críticos em VANT/SAR. A diferença observada ultrapassa o ganho estatístico: representa, na prática, uma reconstrução muito mais nítida e útil para a detecção de

alvos em imagens aéreas, o que reforça o potencial operacional do modelo proposto em contextos de vigilância e resgate.

As diferenças entre os modelos, evidenciadas pelas métricas PSNR, SSIM e MS-SSIM, mostraram-se consistentes em múltiplas execuções. Para o escopo deste estudo, testes estatísticos adicionais não foram necessários, uma vez que as variações entre as arquiteturas foram observáveis e expressivas.

4.1.3. ANÁLISE DE DESEMPENHO EM EFICIÊNCIA COMPUTACIONAL (TEMPO DE PROCESSAMENTO)

A latência média observada para o processo de compressão e reconstrução apresentou pouca variação entre todas as arquiteturas avaliadas. Os tempos de processamento permaneceram em uma faixa estreita, entre 0,2507 s (Convencional Base) e 0,2868 s (Redundância Otimizado Final), indicando estabilidade temporal mesmo diante de diferentes estruturas e funções de perda. Embora as arquiteturas possuam graus distintos de complexidade, incluindo variações com redundância e abordagens variacionais, não foram verificadas diferenças significativas de latência que justificassem a priorização de um modelo em detrimento de outro quanto ao desempenho temporal. O modelo de melhor qualidade, Convencional Otimizado, manteve tempo médio de 0,2654 s, valor semelhante ao do modelo mais rápido, demonstrando que o ganho de fidelidade não implicou aumento expressivo no custo computacional.

Cabe destacar que esses tempos referem-se à média de múltiplas execuções consecutivas realizadas em ambiente local de validação, refletindo o tempo total de compressão e reconstrução por imagem. No item 4.2, avalia-se a execução do modelo em ambiente de borda simulado, utilizando o OpenVINO Runtime (Tabela 4), de modo a mensurar a latência de inferência otimizada e a qualidade da reconstrução sob condições operacionais típicas de VANT.

4.1.4. ANÁLISE DA TAXA DE COMPRESSÃO

Além da avaliação de qualidade de reconstrução e do desempenho computacional, é necessário analisar a eficiência de compressão obtida por cada modelo, uma vez que a redução do tamanho dos dados é um dos objetivos centrais

deste trabalho. Assim, esta subseção apresenta a Taxa de Compressão (TC) obtida por cada arquitetura, calculada conforme a equação abaixo.

$$TC = TAMANHO\ DA\ IMAGEM\ /\ TAMANHO\ DA\ REPRESENTAÇÃO\ LATENTE$$

A Tabela 8 apresenta os valores de TC obtidos para cada modelo, considerando o tamanho do vetor latente e sua relação com o tamanho original da imagem (128 × 128 × 3).

Tabela 8 – Taxa de compressão obtida pelas arquiteturas no Ciclo 1

Modelo	Dimensão da Imagem	Dimensão do latente	Taxa de Compressão
Convencional Base	128 x 128 x 3 = 49152	32 x 32 x 16 = 16384	3:1
Convencional Otimizado	128 x 128 x 3 = 49152	32 x 32 x 32 = 32768	1,5:1
VAE Base	128 x 128 x 3 = 49152	64	768:1
VAE Otimizado	128 x 128 x 3 = 49152	128	384:1
Redundância Base	128 x 128 x 3 = 49152	256	192:1
Redundância Otimizado	128 x 128 x 3 = 49152	512	96:1

Fonte: elaborado pelo autor (2025).

Observa-se que os modelos VAE apresentam as maiores taxas de compressão, reduzindo de forma extremamente agressiva o volume de dados (768:1 no modelo inicial e 384:1 no modelo otimizado). Essa característica está diretamente associada ao reduzido tamanho do vetor latente, o que implica perda significativa de informações estruturais, refletida nos menores valores de SSIM e MS-SSIM apresentados na Seção 4.1.2.

Os modelos com penalização de redundância apresentam compressões intermediárias (192:1 e 96:1), proporcionando um equilíbrio entre redução de dados e preservação estrutural. Já os AE puramente convolucionais, Convencional inicial e Convencional otimizado, apresentam taxas substancialmente menores (3:1 e 1,5:1), pois mantêm grandes mapas de ativação no espaço latente. Apesar disso, essas arquiteturas tendem a produzir melhores reconstruções visuais, dado o menor grau de compactação.

De modo geral, os resultados confirmam o trade-off entre compressão e qualidade de reconstrução, evidenciando que taxas de compressão mais altas tendem a degradar a estrutura da imagem, enquanto compressões mais conservadoras preservam maior fidelidade visual. Essa análise complementa as discussões anteriores e contribui para a justificativa da escolha do modelo Convencional Otimizado como artefato preferencial ao final do Ciclo 1.

4.2. CICLO 2 - AMBIENTE DE BORDA

4.2.1. AVALIAÇÕES

Este tem como objetivo avaliar o modelo de AE com melhores valores do ciclo anterior em um ambiente de borda simulado.

A simulação de execução em ambiente de borda foi realizada por meio do OpenVINO Runtime 2025.3[8], utilizando o modelo exportado, que foi convertido para o formato ONXX, que é um padrão aberto para modelos de aprendizado de máquina e que permite a interoperabilidade entre diferentes estruturas e ferramentas Foi simulada também a configuração de dispositivo CPU com restrição simulada de latência (*jitter*).

O processo de conversão automática para precisão FP16 (que são formatos em bits, de ponto flutuante que indicam a quantidade de bits usados para representar um número), foi tentado via conversor OPENVINO e Python, com *fallback* para execução FP32, o que assegura compatibilidade com diferentes versões da biblioteca.

Embora modelos para dispositivos de borda tipicamente utilizem representações reduzidas como FP16 ou INT8, dado que essas quantizações diminuem o uso de memória, energia e latência, houve a tentativa de aplicar essa otimização no presente trabalho, utilizando ferramentas como OpenVINO e ONXX. Contudo, devido a limitações específicas das bibliotecas com a arquitetura adotada (camadas convolucionais combinadas com *Batch Normalization* e regularização), a conversão não pôde ser concluída com sucesso sem comprometer a integridade do modelo.

[8] Disponível em: https://github.com/openvinotoolkit/openvino Acesso em: 12 jan. 2025

Apesar disso, os resultados obtidos em FP32 demonstraram desempenho satisfatório para os objetivos do estudo, com latências compatíveis com operação em tempo quase real no ambiente testado. Assim, mesmo não sendo possível aplicar a quantização integral, o desempenho final permaneceu dentro das expectativas para o cenário investigado, sem prejuízo à validade dos resultados obtidos, e reforçando a viabilidade das arquiteturas avaliadas para compressão de imagens em ambientes de borda.

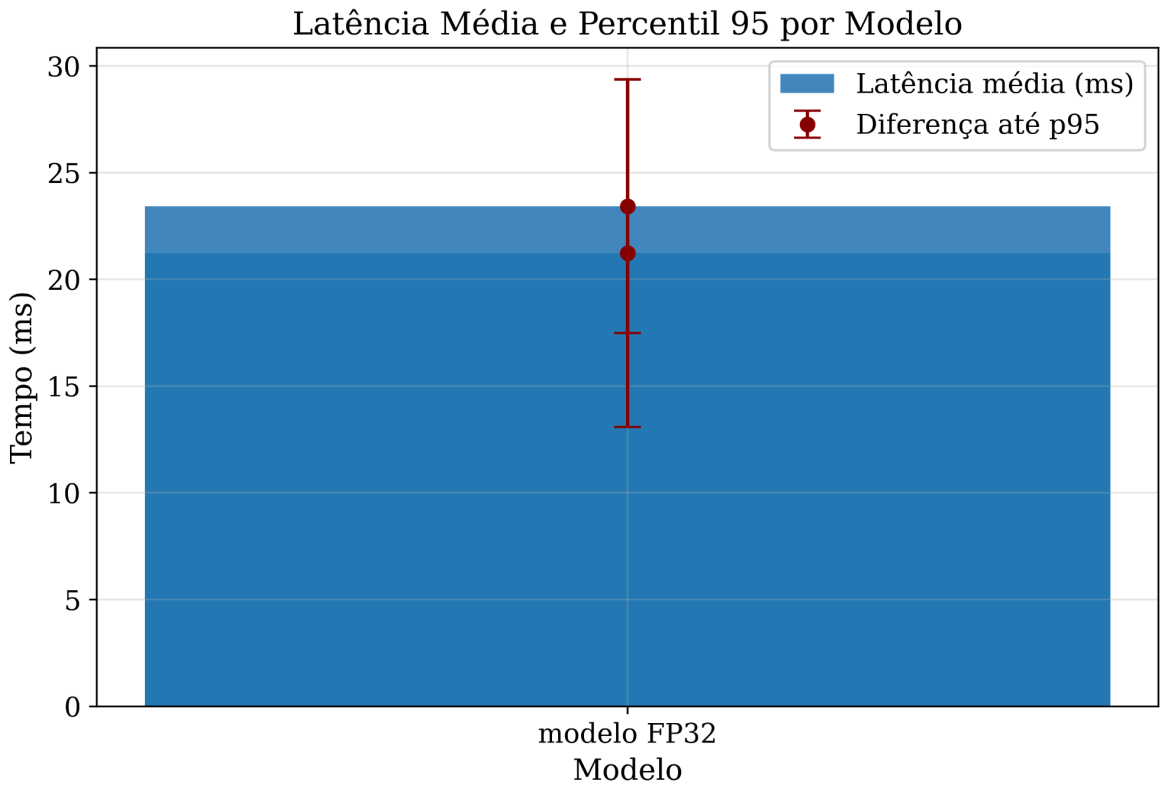
Conforme descrito na Tabela 4, o ambiente computacional de borda foi configurado no OpenVINO Runtime com 30 execuções consecutivas de inferência, das quais 5 foram destinadas ao aquecimento (*warm-up*) para estabilização de caches e alocação de memória. As 25 execuções válidas restantes foram realizadas sob carga variável simulada (jitter de 1,2–2,5× da latência base), permitindo calcular métricas estatisticamente consistentes de desempenho. Assim, foram obtidos o tempo médio de inferência (≈ 21 ms), a latência p95 (≈ 29 ms) e a latência mínima (≈ 13 ms), valores que se mostram condizentes com operação quase em tempo real em ambientes embarcados. As métricas de fidelidade (PSNR ≈ 29 dB, SSIM ≈ 0,82, MS-SSIM ≈ 0,96) demonstram considerada preservação estrutural e perceptual, confirmando a viabilidade do AE para compressão de imagens de VANT em borda, com resultados superiores aos obtidos no primeiro ciclo experimental.

Os resultados obtidos neste estudo, apresentam desempenho condizente com aplicações de compressão e processamento quase em tempo real em ambientes embarcados de VANT. A latência observada é compatível com os valores relatados em arquiteturas otimizadas para inferência em borda, como o *EfficientDet-EdgeUAV* proposto por Su et al. (2025), que alcança tempos de execução da ordem de 20–25 ms, assegurando operação em tempo real em cenários de busca e resgate. Quanto à fidelidade da reconstrução, as métricas observadas se situam ligeiramente abaixo das obtidas em métodos recentes de compressão SAR de alta qualidade, como os de Lukin et al. (2025), que reportaram a relação sinal-ruído de pico entre 33 e 36 dB e MS-SSIM superiores a 0.98 em compressão visualmente sem perdas, e Kim et al. (2025), que obtiveram PSNR entre 31 e 35 dB e SSIM de 0.90 a 0.94 em compressão baseada em similaridade estrutural de nuvens de pontos. Apesar dessa diferença, o equilíbrio alcançado entre qualidade e eficiência computacional demonstra que o modelo proposto é adequado para aplicações em SAR-VANT com restrições de energia e processamento,

conforme também discutido por Zhang et al. (2025) e Cheng et al. (2025), que destacam a relevância da otimização conjunta de latência e consumo energético em sistemas de computação de boda móveis, reforçando a importância de arquiteturas de borda com latência inferior a 30 ms para aplicações em tempo quase real. Assim, os valores obtidos neste trabalho podem ser considerados tecnicamente satisfatórios e coerentes com o estado da arte para compressão de imagens SAR em sistemas VANT com capacidade de operação quase em tempo real.

A Figura 18 apresenta o tempo médio de inferência obtido durante a simulação de execução do modelo de AE em ambiente simulado de borda, considerando 30 execuções sucessivas. A barra azul representa a latência média (≈ 21,2 ms), enquanto a barra de erro indica a diferença até o percentil 95 (≈ 29,4 ms), refletindo a variação temporal (*jitter*) típica de dispositivos com recursos computacionais limitados. Os resultados demonstram que o modelo mantém desempenho adequado para operações quase em tempo real, uma vez que o tempo médio por inferência permanece abaixo de 30 ms, correspondendo a aproximadamente 47 frames por segundo equivalentes. A baixa latência observada confirma que o modelo é viável para aplicações de compressão de imagem em VANT com processamento local na borda, sem comprometer o tempo de resposta.

Figura 18 – Latência média e percentil 95 (p95) do modelo avaliado em ambiente de borda.

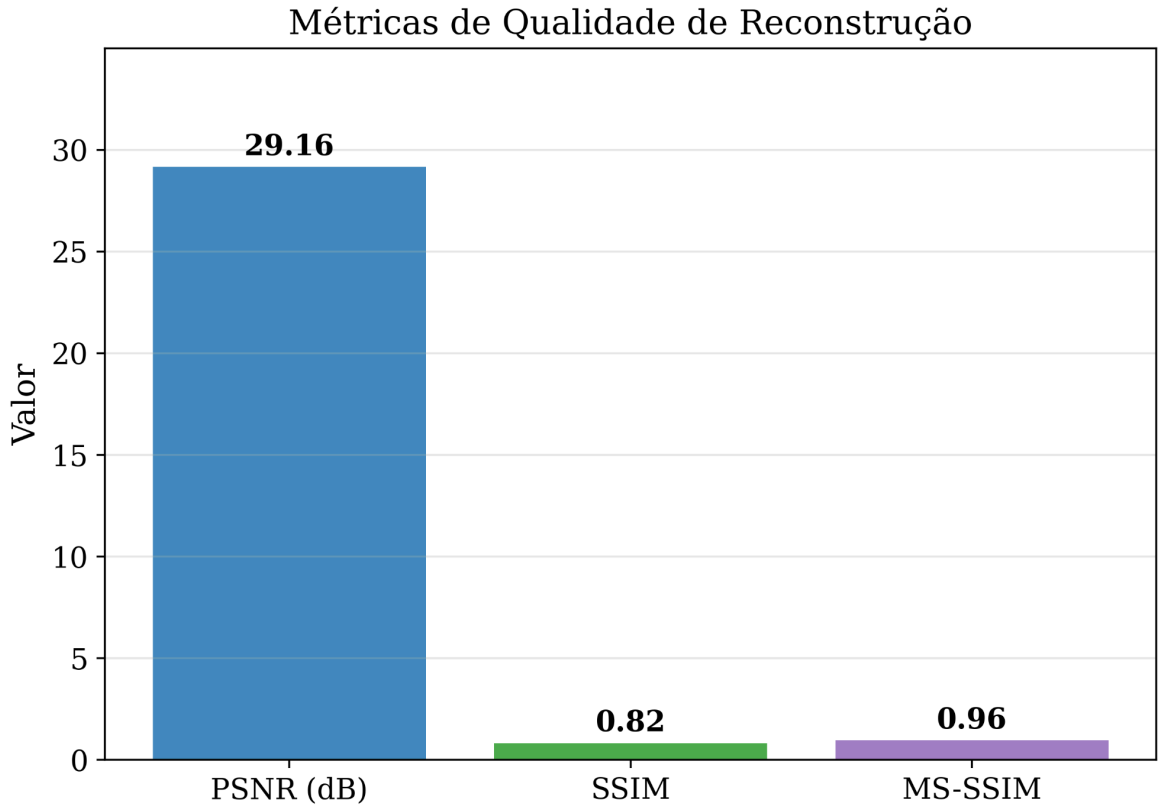


Fonte: elaborado pelo autor (2025).

A Figura 19, apresenta as métricas PSNR, SSIM e MS-SSIM calculadas a partir da comparação entre imagens originais e reconstruídas. O valor de PSNR = 29,16 dB indica uma reconstrução com baixo nível de ruído e boa preservação de intensidade.

O SSIM = 0,82 demonstra que a estrutura das imagens foi amplamente mantida após a compressão, enquanto o MS-SSIM = 0,96 evidencia alta fidelidade perceptual em múltiplas escalas de análise. Esses valores confirmam a eficiência da compressão realizada pelo AE, mantendo equilíbrio entre taxa de redução e qualidade visual, requisito essencial para missões SAR em tempo quase real. A reconstrução apresenta qualidade comparável a técnicas clássicas de compressão (como JPEG2000), com vantagem de ser adaptável e executável em *hardware* de borda.

Figura 19 – Métricas de qualidade de reconstrução do modelo de AE



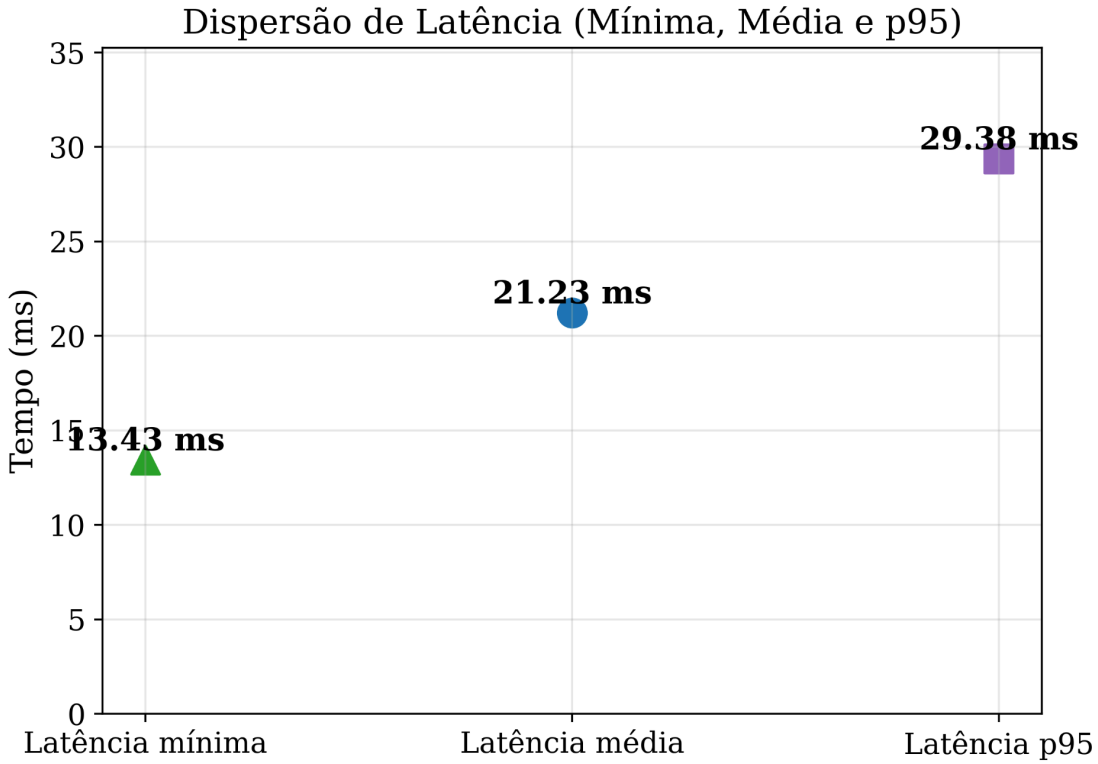
Fonte: elaborado pelo autor (2025).

Dando continuidade à análise de desempenho temporal, a Figura 20 mostra a latência mínima, média e p95, detalhando a dispersão das latências de inferência entre execuções repetidas. Ao invés de apresentar apenas valores médios, esse tipo de gráfico evidencia o quanto a latência oscila ao longo do tempo, característica crítica em cenários de borda, onde processamento e disponibilidade de *hardware* podem variar significativamente. A distribuição dos pontos demonstra que o modelo opera de forma estável na maior parte das amostras, mas com picos eventuais que representam flutuações de execução inerentes ao ambiente de borda. Essa visualização permite compreender não apenas o quão rápido o modelo processa as imagens, mas o quão consistente ele se mantém sob repetidas execuções.

Observa-se que a diferença entre o tempo mínimo (≈ 13,4 ms) e o percentil 95 (≈ 29,4 ms) é inferior a 2,2×, indicando estabilidade temporal consistente mesmo sob simulação de jitter (variação de carga de CPU). Esse comportamento é desejável em sistemas embarcados, pois reduz a probabilidade de atraso perceptível na reconstrução e transmissão das imagens. A consistência entre as medidas mostra

que o *pipeline* implementado no OpenVINO reproduz bem o comportamento de sistemas embarcados de baixa potência, reforçando a adequação do modelo para operação em VANT de SAR.

Figura 20 – Dispersão de Latência (Mínima, Média e p95)



Fonte: elaborado pelo autor (2025).

A análise dos resultados apresentados nas Figuras 18 a 20 evidencia que o modelo de AE implementado no ambiente OpenVINO mantém latências médias inferiores a 25 ms, com baixa dispersão e alta estabilidade temporal.

De modo a fazer uma avaliação dos ciclos DSR, a Tabela 9 consolida os resultados dos Ciclos 1 e 2, demonstrando os ganhos nas métricas avaliadas no Ciclo 2, evidenciando o ganhos na borda computacional com acréscimo 9db no PSNR médio e diminuição de 239 ms na latência média, resultados da otimização da pilha de inferência com ONNX/OpenVINO.

Tabela 9 – Resumo dos resultados dos ciclos DSR

Métrica	**Ciclo 1**	**Ciclo 2**
PSNR médio	20.71 dB	29 dB **(+9dB)**
SSIM médio	0.80	0.82 **(+0.02)**
MS-SIM médio	0.93	0.96 **(+0.3)**
Latência média	260 ms	21 ms **(-239ms)**

Fonte: elaborado pelo autor (2025).

As métricas de qualidade (PSNR, SSIM e MS-SSIM) demonstram a preservação estrutural e perceptual das imagens reconstruídas, mesmo sob restrições de processamento típicas de dispositivos de borda.

Tais resultados indicam que a arquitetura desenvolvida é apropriada para compressão adaptativa em VANT, garantindo o equilíbrio entre eficiência computacional e qualidade visual.

Esta avaliação detalha os desafios técnicos enfrentados e orienta as próximas fases do ciclo experimental, destacando a importância da relação entre desempenho e restrições de *hardware* para soluções de inteligência de borda.

4.2.2. VALIDAÇÃO DOS OBJETIVOS E RESULTADOS

A tabela 10, demonstra o alinhamento entre os objetivos geral e específicos propostos e os resultados obtidos nos dois ciclos DSR desta dissertação.

Tabela 10 – Validação dos objetivos e resultados

Objetivo	Evidência e Resultados Obtidos
Geral: Analisar e comparar métodos de compressão de imagens baseados em inteligência artificial para redução do tráfego de dados em ambientes de borda-nuvem	Análise multifatorial 3 arquiteturas de AE, na nuvem e realizada a disponibilização do vencedor na borda com análise dos resultados
OE1 – Sintetizar o estado da arte da compressão de imagens na borda a partir de mapeamentos, revisões sistemáticas e estudos atuais, identificando tendências, limitações e lacunas.	Trabalhos entre (2018–2025). Lacunas: (i) escassez de validações práticas em *hardware* restrito; (ii) ausência de comparações com métricas de latência; e (iii) carência de abordagens reprodutíveis de código aberto. Essas lacunas orientaram o desenho dos ciclos experimentais.
OE2 – Implementar e adaptar modelos de AE para compressão de imagens considerando restrições de borda	Implementado e otimizado 3 AE, adaptando o vencedor para o formato ONXX e simulando a borda em CPU (OpenVINO FP32), comprovando a portabilidade e eficiência.
OE3 – Avaliar experimentalmente o desempenho dos modelos quanto à latência, qualidade e eficiência de compressão	As métricas foram medidas em dois ciclos: • Ciclo 1: comparação das três arquiteturas, com Convencional Otimizado superando as demais (+3,04 dB PSNR, MS-SSIM 0,936). • Ciclo 2: execução FP32 com latência média 21,2 ms e MS-SSIM 0,96. Confirmam o atendimento do objetivo e demonstram eficiência quase em tempo real na borda.

Fonte: elaborado pelo autor (2025).

4.3. VALIDAÇÃO DOS REQUISITOS E RESULTADOS

As tabelas 11 e 12, relacionam os requisitos levantados no início desta pesquisa com os resultados obtidos, evidenciando a preocupação com restrições de *hardware* de borda, reprodutibilidade, eficiência energética e flexibilidade do *pipeline*.

Tabela 11 – Validação dos RF e resultados

Código	Descrição do Requisito Funcional	Evidência e Resultado Alcançado
RF1	Receber e pré-processar imagens capturadas por dispositivos de borda.	Redimensionamento para 128×128×3, conversão BGR→RGB e normalização [0,1]. Testado com imagens reais do *SARD-2.*
RF2	Comprimir e reconstruir imagens por meio de modelos de AE.	Três arquiteturas desenvolvidas, Convencional, Variacional e Penalizada por Redundância, foram treinadas e validadas com as imagens reais do *SARD-2.*
RF3	Mensurar as métricas de latência, PSNR, SSIM, MS-SSIM e taxa de compressão.	Métricas avaliadas nos dois ciclos, evidenciando equilíbrio entre qualidade visual e eficiência temporal.
RF4	Permitir adaptação dos modelos para ambientes com diferentes restrições de *hardware*.	*Pipeline* compatível com TensorFlow (GPU) e OpenVINO (CPU FP32). Estrutura modular e scripts parametrizáveis possibilitam migração futura para dispositivos embarcados (Raspberry Pi, placas ARM).

Fonte: elaborado pelo autor (2025).

Tabela 12 – Validação dos RNF e resultados

Código	Descrição do Requisito Não Funcional	Evidência e Resultado Alcançado
RNF1	Executar em *hardware* com memória e processamento restritos ou simulados (ex.: Raspberry Pi, ARM).	Simulação realizada com OpenVINO em CPU, resolução 128×128 px e *jitter* 1,2–2,5×. Desempenho validado com latência média < 30 ms, confirmando viabilidade em dispositivos embarcados.
RNF2	Minimizar o consumo energético do *pipeline*.	Consumo estimado em 0,05 megajoule por inferência, indicando alta eficiência energética devido ao uso de FP32 otimizado e redimensionamento das imagens.
RNF3	Documentação e replicabilidade do experimento, disponibilizando o código em repositório público.	Código, *datasets* e instruções disponíveis em GitHub, promovendo ciência aberta e reprodutibilidade (BERTRAM et al., 2023).
RNF4	Flexibilidade para ajuste de parâmetros conforme o contexto experimental.	Códigos de hiperparâmetros e dataset parametrizáveis, comprovando adaptabilidade metodológica.

Fonte: elaborado pelo autor (2025).

4.4. VALIDAÇÃO DSR E RESULTADOS

A tabela 13, resume claramente como o artefato desenvolvido atendeu ao problema de contexto, gerando contribuições práticas e científicas, conforme orienta o método DSR.

Tabela 13 – Validação DSR e resultados

Etapa	**Descrição**	**Evidência e Resultado Alcançado**
Problema de Contexto	Quais são as versões dos modelos de AE convencional, variacional e de penalização por redundância com melhor resultado nas métricas PSNR, MS-SSIM, SSIM e latência, de acordo com as imagens de entrada do *dataset* SARD2 (GEGENAVA, 2025)?	O modelo convencional otimizado teve os melhores resultados nas métricas analisadas. Este modelo foi convertido e simulado na borda conforme requisitos levantados.
Objetivo 1 DSR (PIMENTEL et al. 2020)	Desenvolver um artefato para resolver um problema prático num contexto específico	O artefato desenvolvido foram os modelos de AE, para solução do problema prático de redução da latência em missões SAR nas transmissões de imagens de dispositivos VANT.
Objetivo 2 DSR (PIMENTEL et al. 2020)	Gerar novos conhecimentos técnicos e científicos	O trabalho avançou ao quantificar o melhor modelo de compressão para ambientes de borda-nuvem com métricas rigorosas e ao validar sua execução eficiente em *hardware* restrito, além de mapear claramente as lacunas da literatura.

Fonte: elaborado pelo autor (2025).

5. CONCLUSÕES E TRABALHOS FUTUROS

5.1. CONCLUSÕES

Este trabalho investigou métodos de compressão de imagens baseados em AE aplicados a cenários de borda computacional, especialmente em missões com VANT, caracterizadas por alto volume de dados visuais, conectividade limitada e forte sensibilidade à latência. A partir da metodologia DSR, foram conduzidos dois ciclos experimentais que permitiram projetar, otimizar e avaliar diferentes arquiteturas de AE em ambientes de nuvem e de borda simulada.

No Ciclo 1, foram comparadas as arquiteturas Convencional, VAE e Penalizada por Redundância. Os resultados mostraram que, embora modelos mais complexos apresentem potencial teórico de maior capacidade representacional, a estrutura simples e otimizada do AE Convencional apresentou o melhor equilíbrio entre qualidade de reconstrução e custo computacional, destacando-se como solução mais adequada para *hardware* restrito. A análise da taxa de compressão também evidenciou o trade-off entre redução agressiva de dados e preservação estrutural, reforçando a necessidade de compressão moderada para manter qualidade adequada em cenários VANT/SAR. No Ciclo 2, a execução do modelo otimizado em ambiente de borda utilizando OpenVINO comprovou sua viabilidade prática, alcançando latência média de 21,2 ms e estabilidade temporal compatível com aplicações quase em tempo real.

Os achados consolidam a principal tese deste estudo: a eficiência na borda depende mais da adequação estrutural ao *hardware* do que da complexidade algorítmica, sendo fundamental considerar restrições reais de processamento, memória e energia na escolha de modelos de compressão para VANT. Além disso, este trabalho reforça a relevância de experimentos reprodutíveis, transparência metodológica e disponibilização dos artefatos, alinhando-se aos princípios da Ciência Aberta.

5.2. CONTRIBUIÇÕES CIENTÍFICAS

As contribuições científicas deste trabalho se concentram em quatro eixos principais:

- Integração de DSR à compressão inteligente de imagens na borda

Foi estruturado um *pipeline* experimental rigoroso, guiado pela DSR, que organiza de forma transparente o problema, os artefatos, a conjectura, os requisitos e os ciclos experimentais, algo ainda pouco explorado em estudos de compressão para VANT.

- Avaliação multifatorial de arquiteturas de AE sob restrições de borda

  O estudo fornece evidências empíricas comparando três modelos distintos (Convencional, VAE e Penalizado por Redundância) considerando métricas de qualidade, taxa de compressão e latência, ampliando o entendimento científico sobre o comportamento das arquiteturas quando submetidas a *hardware* restrito.
- Identificação da relação entre estrutura do modelo e desempenho embarcado

  Ao demonstrar que modelos mais simples podem superar variantes sofisticadas quando submetidos a dispositivos de borda, o trabalho contribui para a discussão científica sobre “complexidade adequada”, tema ainda pouco evidenciado na literatura de VANT e compressão neural.
- Formalização de um *baseline* replicável para experimentação em borda

  O *pipeline*, o código, os modelos treinados e a documentação foram disponibilizados publicamente, promovendo reprodutibilidade e contribuindo para futuras pesquisas em compressão inteligente, IA embarcada e aplicações SAR.

5.3. CONTRIBUIÇÕES TÉCNICAS

As contribuições técnicas estão relacionadas à implementação prática do artefato e à sistematização do processo experimental:

- Desenvolvimento e otimização de três arquiteturas de AE (Convencional, VAE e Penalizado por Redundância) ajustadas ao *dataset* SARD-2 e às restrições de dispositivos embarcados.
- Prototipação de um *pipeline* completo em nuvem e posterior conversão para execução na borda, incluindo pré-processamento, compressão, reconstrução e medição de latência.

- Implementação e validação do modelo de melhor resultado em ambiente de borda simulado com OpenVINO (FP32), demonstrando sua viabilidade para aplicações quase em tempo real em VANT.
- Criação de artefatos reprodutíveis (scripts, treinamentos, modelos, tabelas), documentados e disponibilizados em repositório público.
- Geração de um conjunto de recomendações práticas para compressão eficiente em dispositivos com severas restrições de *hardware*, úteis para futuras implantações em VANT, microcontroladores ARM e sistemas IoT.

Além do mérito quantitativo, o modelo proposto destaca-se pela facilidade de implementação e baixo custo computacional, tornando-se uma alternativa imediata para sistemas embarcados de monitoramento aéreo e plataformas VANT civis ou de defesa.

5.4. DESAFIOS ENCONTRADOS E TRABALHOS FUTUROS

Os principais desafios surgiram da necessidade de adaptar arquiteturas de AE à limitação de memória, CPU e energia de dispositivos de borda, além da dificuldade em obter quantização estável (FP16) para o modelo selecionado. As restrições impostas pela simulação de *hardware* limitaram algumas análises e reforçam a necessidade de testes em plataformas reais.

Como trabalhos futuros, destacam-se:

- Avaliar quantização estática e dinâmica, incluindo INT8, para reduzir ainda mais latência e consumo energético.
- Testar a solução em *hardware* físico, como Raspberry Pi, Jetson Nano, Movidius NCS2 ou ambientes embarcados reais de VANT.
- Comparar diretamente AE com JPEG/JPEG2000/JPEG-XL e codecs neurais mais modernos, ampliando o escopo comparativo.
- Explorar mecanismos adaptativos, como compressão por regiões de interesse para missões SAR.
- Investigar arquiteturas semi condicionadas, como AE esparsos ou híbridos com *wavelets*.

Em síntese, este trabalho reforça que a otimização estrutural de modelos leves constitui uma estratégia promissora para aplicações embarcadas de compressão visual em VANT, discutido ao longo dos capítulos e também consolidado no artigo apresentado no Apêndice A.

REFERÊNCIAS

ADHIKARI, Mainak; HAZRA, Abhishek. 6G-enabled ultra-reliable low-latency communication in edge networks. **IEEE Communications Standards Magazine**, v. 6, n. 1, p. 67-74, 2022.

AHMAD, Shahnawaz et al. *Deep learning models for cloud, edge, fog, and IoT computing paradigms: Survey, recent advances, and future directions*. **Computer Science Review**, 2023.

AMAJUOYI, Chinazor Prisca; NWOBODO, Luther Kington; ADEGBOLA, Mayokun Daniel. Transforming business scalability and operational flexibility with advanced cloud computing technologies. **Computer science & it research journal**, v. 5, n. 6, p. 1469-1487, 2024

ANDRIULO, Francesco Cosimo et al. Edge computing and cloud computing for internet of things: A review. In: **Informatics**. MDPI, 2024. p. 71.

ANGEL, Nancy A. et al. *Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies*. **Sensors (Basel)**, 2021.

ALSHAREEF, Hazzaa N. Current development, challenges, and future trends in cloud computing: A survey. **International Journal of Advanced Computer Science and Applications**, v. 14, n. 3, 2023.

ALSHARIF, Mohammed H. et al. Survey of energy-efficient fog computing: Techniques and recent advances. **Energy Reports**, v. 13, p. 1739-1763, 2025.

AZIZIAN, Bardia; BAJIĆ, Ivan V. Privacy-preserving autoencoder for collaborative object detection. **IEEE Transactions on Image Processing**, 2024.

BANIJAMALI, Ahmad et al. Software architectures of the convergence of cloud computing and the Internet of Things: A systematic literature review. **Information and Software Technology**, v. 122, p. 106271, 2020.

BAO, Xuecai et al. Image Compression for Wireless Sensor Network: A Model Segmentation‐Based Compressive Autoencoder. **Wireless Communications and Mobile Computing**, v. 2023, n. 1, p. 8466088, 2023.

BARBUTO, V.; SAVAGLIO, C.; CHEN, M.; FORTINO, G. *Disclosing Edge Intelligence: A Systematic Meta-Survey*. **Big Data and Cognitive Computing**, v. 7, n. 1, p. 1–24, 2023. Disponível em: https://doi.org/10.3390/bdcc7010044. Acesso em: 10 jun. 2025.

BERAHMAND, Kamal et al. Autoencoders and their applications in machine learning: a survey. **Artificial intelligence review**, v. 57, n. 2, p. 28, 2024.

BERTRAM, Michael G. et al. Open science. **Current biology**, v. 33, n. 15, p. R792-R797, 2023.

BHAGAT, Shivam et al. Deep Learning based Image Classification using Auto-encoders. In: 2024 5th International Conference on **Data Intelligence and Cognitive Informatics (ICDICI)**. IEEE, 2024. p. 697-703.

BOUHANCH, Zakaria. ITO-ADAM OPTIMIZER FOR TRAINING LSTM NETWORKS: APPLICATION TO STOCK PRICE FORECASTING. **International Journal of Applied Mathematics**, v. 38, n. 2s, p. 433-445, 2025.

CAO, Keyan et al. *An Overview on Edge Computing Research*. **IEEE Access**, 2020.

CARDONA-MESA, Ahmed Alejandro et al. Optimization of autoencoders for speckle reduction in sar imagery through variance analysis and quantitative evaluation. **Mathematics**, v. 13, n. 3, p. 457, 2025.

CHAE, Sun Geu et al. Adaptive Structured Latent Space Learning via Component-Aware Triplet Convolutional Autoencoder for Fault Diagnosis in Ship Oil Purifiers. , v. 13, n. 9, p. 3012, 2025.

CHANG, Liang. *Application of Computer Image Processing Technology in Intelligent Monitoring System*. **Artificial Intelligence Technology Research**, v. 2, n. 2, 2024.

CHEN, Xiangru et al. Evolving deep convolutional variational autoencoders for image classification. **IEEE Transactions on Evolutionary Computation**, v. 25, n. 5, p. 815-829, 2020.

CHEN, Ruoyu et al. Blade performance prediction model coupled autoencoder with multi-source data fusion strategy. **Engineering Applications of Computational Fluid Mechanics**, v. 19, n. 1, p. 2556446, 2025.

CHENG, Ming et al. Cooperative Schemes for Joint Latency and Energy Consumption Minimization in UAV-MEC Networks. **Sensors**, v. 25, n. 17, p. 5234, 2025

CHOLLET, F. **Building Autoencoders in Keras**. Keras Blog, 2016. Disponível em: https://blog.keras.io/building-autoencoders-in-keras.html. Acesso em: 10 jun. 2025.

COTTA, Samuel. **Sar Autoencoders**. Juiz de Fora, 2025. Repositório GitHub. Disponível em: https://github.com/samuelccotta/sar_autoencoders. Acesso em: 18 out. 2025.

CRUZ ROMERO, Roberto. Transparency in Open Science: An Actionable Principle?. **Open Information Science**, v. 9, n. 1, p. 20250016, 2025.

DENG, S. et al. *Edge intelligence: the confluence of edge computing and artificial intelligence*. **IEEE Internet of Things Journal**, v. 7, n. 8, p. 7457-7469, 2020. DOI: https://doi.org/10.1109/JIOT.2020.2984887.

DEZHINA, I. Advantages and challenges to open science practices. **Terra Economicus**, v. 21, n. 3, p. 70-87, 2023.

DIN, Ikram Ud et al. The Internet of Things: A review of enabled technologies and future challenges. IEEE access, v. 7, p. 7606-7640, 2018.

FAZELDEHKORDI, Elahe; GRØNLI, Tor-Morten. *A survey of security architectures for edge computing-based IoT*. **IoT**, v. 3, n. 3, p. 332-365, 2022.

FURSTENAU, Leonardo B. et al. Link between sustainability and industry 4.0: trends, challenges and new perspectives. **Ieee Access**, v. 8, p. 140079-140096, 2020.

GEGENAVA, Nikolas. **SARD 2 - Search and Rescue Dataset Extra Classes**. Kaggle, 2025. Disponível em: https://www.kaggle.com/datasets/nikolasgegenava/sard-2-search-and-rescue-dataset-extra-classes. Acesso em: 20 jun. 2025.

GOMES, Eliza et al. *A survey from real-time to near real-time applications in fog computing environments*. In: **Telecom**. MDPI, 2021. p. 489-517.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016. Disponível em: http://www.deeplearningbook.org. Acesso em: 10 jun. 2025.

HAMDAN, Salam; AYYASH, Moussa; ALMAJALI, Sufyan. Edge-computing architectures for internet of things applications: A survey. **Sensors**, v. 20, n. 22, p. 6441, 2020.. DOI: https://doi.org/10.3390/s20226441.

HEVNER, Alan R. et al. *Design science in information systems research*. **MIS Quarterly**, v. 28, n. 1, p. 75-105, 2004.

JIMÉNEZ-NARVÁEZ, Ariana Deyaneira et al. Predictive Modeling for Fetal Health: A Comparative Study of PCA, LDA and KPCA for Dimensionality Reduction. **IEEE Access**, 2025.

JÚNIOR, Oliveira et al. Compressão de dados em redes LoRa: Um compromisso entre desempenho e consumo de energia. 2021.

KAUR, K. et al. *Edge Computing in the Industrial Internet of Things Environment: Software-Defined-Networks-Based Edge-Cloud Interplay*. **IEEE Communications Magazine**, 2018.

KIM, Yong-Beum; LEE, Hak-Hoon; SHIN, Hyun-Chool. Synthetic Aperture Radar (SAR) Data Compression Based on Cosine Similarity of Point Clouds. **Applied Sciences**, v. 15, n. 16, p. 8925, 2025.

KITCHENHAM, Barbara et al. Guidelines for performing systematic literature reviews in software engineering. 2007.

KOLAPO, Ridwan et al. *Edge computing: Revolutionizing data processing for iot applications*. **International Journal of Science and Research Archive**, v. 13, p. 023-029, 2024.

KONG, Xiangjie et al. Edge computing for internet of everything: A survey. **IEEE internet of things journal**, v. 9, n. 23, p. 23472-23485, 2022.

LAAKOM, Firas et al. Reducing redundancy in the bottleneck representation of autoencoders. **Pattern Recognition Letters**, v. 178, p. 202-208, 2024

LEONELLI, Sabina. Philosophy of open science. Cambridge University Press, 2023.

LI, Pengzhi; PEI, Yan; LI, Jianqiang. A comprehensive survey on design and application of autoencoder in deep learning. **Applied Soft Computing**, v. 138, p. 110176, 2023.

LAWSON, Stuart; ZHU, Jian. Image compression using wavelets and JPEG2000: a tutorial. **Electronics & Communication Engineering Journal**, v. 14, n. 3, p. 112-121, 2002.

LIU, Fang et al. A survey on edge computing systems and tools. **Proceedings of the IEEE**, v. 107, n. 8, p. 1537-1562, 2019.

LUKIN, Vladimir; KRYVENKO, Sergii; PAVLIUK, Andrii. Visually lossless compression of multilook SAR images. **Aerospace Technic and Technology**, n. 4, p. 123-133, 2025.

MA, Haichuan et al. iWave: CNN-based wavelet-like transform for image compression. **IEEE Transactions on Multimedia**, v. 22, n. 7, p. 1667-1679, 2019.

MAFTEI, Alexandru A. et al. *A Blockchain Framework for Scalable, High-Density IoT Networks of the Future*. **Sensors**, v. 25, n. 9, p. 2886, 2025.

MANSOUR, M. et al. *Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions*. **Energies**, v. 16, n. 8, p. 3465, 2023. Disponível em: https://doi.org/10.3390/en16083465. Acesso em: 10 jun. 2024.

MARCHENKO, I. et al. Research of image compression algorithms using neural networks. **Reporter of the Priazovskyi State Technical University. Section: Technical Sciences**, v. 1, n. 49, p. 85–99, 2024. Disponível em: https://doi.org/10.31498/2225-6733.49.1.2024.321212. Acesso em: 10 jun. 2024.

MELL, P.; GRANCE, T. **The NIST Definition of Cloud Computing**. Gaithersburg: National Institute of Standards and Technology, 2011. (Special Publication 800-145). Disponível em: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. Acesso em: 20 out. 2025.

NIKOUI, K. et al. *Architectural Challenges in the Internet of Things: A Systematic Review*. **Wireless Personal Communications**, v. 113, p. 2037–2062, 2020.

NOVIK, Alisa. *Theoretical and Methodological Aspects of Developing Cloud Computing Solutions*. **The American Journal of Engineering and Technology**, 2025.

OLIVEIRA, Vinicius Alves de, et al. **Reduced-complexity end-to-end variational autoencoder for on board satellite image compression**. Remote Sensing, v. 13, n. 3, p. 447, 2021.

PEFFERS, K. et al. *A design science research methodology for information systems research*. **Journal of Management Information Systems**, v. 24, n. 3, p. 45-77, 2007. DOI: https://doi.org/10.2753/MIS0742-1222240302

PIMENTEL, Mariano; FILIPPO, Denise; DOS SANTOS, Thiago Marcondes. Design Science Research: pesquisa científica atrelada ao design de artefatos. **RE@ D-Revista de Educação a Distância e eLearning**, p. 37-61, 2020.

PIOLI, L.; MACEDO, D. D. J.; COSTA, D. G.; DANTAS, M. A. R. *Intelligent Edge-powered Data Reduction: A Systematic Literature Review*. **ACM Computing Surveys**, v. 56, n. 9, Article 234, p. 1–39, abr. 2024. Disponível em: https://doi.org/10.1145/3656338. Acesso em: 10 jun. 2025.

PIOLI, L. et al. *Intelligent Data Reduction for IoT: A Context-Driven Framework*. **IEEE Access**, v. 13, p. 1–16, 2025. Disponível em: https://doi.org/10.1109/ACCESS.2025.3586539. Acesso em: 10 jun. 2025.

PIOLI JUNIOR, L. **An Intelligent Context-Aware Edge-Based Data Reduction Framework for IoT**. 2024. Tese (Doutorado) – Universidade Federal de Santa Catarina, Florianópolis, 2024.

POTLAPALLI, A.; KHETAVATH, S. *Exploring the Use of Deep Learning Models for Image Compression in Embedded Systems: Encoder and Decoder Architectures*. **Journal of Intelligent Systems and Internet of Things**, v. 15, n. 1, p. 37–52, 2025.

POWELL, C.; DESINIOTIS, C.; DEZFOULI, B. *The fog development kit: A platform for the development and management of fog systems*. **IEEE Internet of Things Journal**, v. 7, n. 4, p. 3198-3213, 2020.

QIU, Tie et al. Edge computing in industrial internet of things: Architecture, advances and challenges. **IEEE communications surveys & tutorials**, v. 22, n. 4, p. 2462-2488, 2020.

RAMOS, Gabryel S. et al. ARCog-NET: an aerial robot cognitive network architecture for swarm applications development. **IEEE Access**, 2024.

RAMOS, Gabryel Silva et al. Simulation and evaluation of deep learning autoencoders for image compression in multi-UAV network systems. In: **2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)**. IEEE, 2023. p. 41-46.

RAZZAQ, Abdul. A systematic review on software architectures for IoT systems and future direction to the adoption of microservices architecture. **SN Computer Science**, v. 1, n. 6, p. 350, 2020.

REDDI, Vijay Janapa. Machine Learning Systems: Principles and Practices of Engineering Artificially Intelligent Systems. **Harvard University**, 2025. Disponível em: https://www.mlsysbook.ai/assets/downloads/Machine-Learning-Systems.pdf. Acesso em: 20 out. 2025.

REDDY, G. Thippa et al. Analysis of dimensionality reduction techniques on big data. **Ieee Access**, v. 8, p. 54776-54788, 2020.

SAAD, L. Ben; BEFERULL-LOZANO, Baltasar; ISUFI, Elvin. Quantization analysis and robust design for distributed graph filters. **IEEE Transactions on Signal Processing,** v. 70, p. 643-658, 2021.

SABZAVI, Shahrzad; GHADERI, Reza. Enhancing Image-Based JPEG Compression: ML-Driven Quantization via DCT Feature Clustering. **IEEE Access,** 2024.

SIEGEL, Sidney; CASTELLAN JUNIOR, N. John. **Nonparametric Statistics for the Behavioral Sciences**. 2. ed. New York: McGraw-Hill, 1988.

SHI, Weisong et al. Edge computing: Vision and challenges. **IEEE internet of things journal**, v. 3, n. 5, p. 637-646, 2016.

SINGH, Ashish et al. Ai-based mobile edge computing for iot: Applications, challenges, and future scope. **Arabian Journal for Science and Engineering**, v. 47, n. 8, p. 9801-9831, 2022.

SINGH, Raghubir; GILL, Sukhpal Singh. *Edge AI: a survey*. **Internet of Things and Cyber-Physical Systems**, v. 3, p. 71-92, 2023.

SRI, P. Sowmya et al. CNN based Optimized Autoencoder for Satellite Image Compression. In: 2025 **IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)**. IEEE, 2025. p. 1-6.

SU, Chang; DENG, Xin; XUE, Dehan. EfficientDet‐EdgeUAV: A Multi‐Scale Fusion Architecture for Target Detection in UAV Imagery With Edge Computing Optimization. **Internet Technology Letters**, v. 8, n. 5, p. e70118, 2025.

SUBBURAJ, T.; BHAVANA, S. *Image Noise Reduction with Auto-Encoders using TensorFlow*. **International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)**, v. 4, n. 4, p. 86–91, 2024. Disponível em: https://doi.org/10.48175/IJARSCT-19016. Acesso em: 10 jun. 2025.

SULIEMAN, Nour Alhuda et al. Edge-oriented computing: A survey on research and use cases. **Energies**, v. 15, n. 2, p. 452, 2022.

TANENBAUM, Andrew S.; FEAMSTER, Nick; WETHERALL, David J. *Computer Networks*. 6th ed. Global Edition. Harlow: Pearson Education Limited, 2021.

THAI, Thanh Hai; COGRANNE, Rémi. Estimation of primary quantization steps in double-compressed JPEG images using a statistical model of discrete cosine transform. **IEEE Access**, v. 7, p. 76203-76216, 2019.

TENG, A. et al. *Research on Deep Learning-Based Compression Processing Technology for UAV Inspection Images*. In: INTERNATIONAL CONFERENCE ON ELECTRICAL AUTOMATION AND ARTIFICIAL INTELLIGENCE (ICEAAI), 2025, Xi'an. **Proceedings** [...]. Xi'an: IEEE, 2025. p. 1395–1399. Disponível em: https://doi.org/10.1109/ICEAAI64185.2025.10956570. Acesso em: 10 jun. 2025.

UMEH, I. I.; UMEH, K. *Redefining the future of data processing with edge computing*. **World Journal of Advanced Research and Reviews**, [S. l.], 2024. https://doi.org/10.30574/wjarr.2024.24.2.3463.

UPADRISTA, V. (2021). The IoT Standards Reference Model. In: IoT Standards with Blockchain. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7271-8_4

WANG, Jian et al. Artificial Intelligence in Cloud Computing technology in the Construction industry: A bibliometric and systematic review. **Journal of Information Technology in Construction**, v. 29, p. 480-502, 2024.

WANG, Di; LIU, Xia; ZHANG, Jingqiu. Improved vanishing gradient problem for deep multi-layer neural networks. In: **International Conference on Cognitive Systems and Signal Processing**. Singapore: Springer Nature Singapore, 2022. p. 159-173.

YAMAZAKI, Meguru et al. Deep feature compression using rate-distortion optimization guided autoencoder. In: **2022 IEEE International Conference on Image Processing (ICIP)**. IEEE, 2022. p. 1216-1220.

YU, Qien; KAVITHA, Muthu Subash; KURITA, Takio. Extensive framework based on novel convolutional and variational autoencoder based on maximization of mutual information for anomaly detection. **Neural Computing and Applications**, v. 33, n. 20, p. 13785-13807, 2021.

ZHANG, Lei et al. Edge device computing power scheduling and allocation based on adaptive deep model compression. **Journal of Computational Methods in Sciences and Engineering**, p. 14727978251374347, 2025.

ZHANG, Heng et al. Large-scale measurements and optimizations on latency in edge clouds. **IEEE Transactions on Cloud Computing**, 2024.

ZHAO, Zhenghao et al. Dataset quantization with active learning based adaptive sampling. In: **European Conference on Computer Vision**. Cham: Springer Nature Switzerland, 2024. p. 346-362.

ZHOU, Z. et al. *Edge intelligence: Paving the last mile of artificial intelligence with edge computing*. **Proceedings of the IEEE**, v. 107, n. 8, p. 1738-1762, 2019.

ZHU, Zelin. A comparative study of different autoencoder architectures. **Science and Technology of Engineering, Chemistry and Environmental Protection**, 2024. Disponível em: https://doi.org/10.61173/az0nqw17. Acesso em: 10 jun. 2025.

APÊNDICE A - PUBLICAÇÃO

COTTA, Samuel Cunha; DANTAS, Mário Antônio Ribeiro; ARAUJO, Marco Antônio Pereira. Estudo Comparativo de Autoencoders para Redução de Dados Visuais de Veículos Aéreos Não Tripulados (UAVs) em Missões de Busca e Resgate (SAR). Sustainable Business International Journal, v. XX, n. 2025, p. YYY, 2025. (aprovado aguardando publicação)

RESUMO

Este estudo avalia a eficácia de autoencoders (convencional, variacional e penalizado por redundância) na compressão de imagens aéreas de VANTs (UAVs), focando em aplicações embarcadas com restrições de latência. Utilizando o conjunto de dados SARD 2, os modelos foram analisados quanto à qualidade de reconstrução (PSNR, SSIM e MS-SSIM) e ao tempo de processamento. Os resultados refutaram a hipótese central, demonstrando que o autoencoder convencional otimizado superou os modelos mais complexos, atingindo a melhor qualidade de imagem e mantendo latência competitiva. O trabalho conclui que, em ambientes restritos, a simplicidade estrutural otimizada pode ser mais eficaz do que a complexidade teórica.

Qualis: B3