SUSTAINABLE BUSINESS INTERNATIONAL JOURNAL

Revista científica - ISSN 1807-5908
Avaliado pelo sistema Double Blind Review
Disponivel em https://periodicos.uff.br/sbijournal

PPGAd uff
Programa de Pós-Graduação em Administração - UFF

Estudo Comparativo de *Autoencoders* para Redução de Dados Visuais de Veículos Aéreos Não Tripulados (UAVs) em Missões de Busca e Resgate (SAR)

Samuel Cotta
samuel.cotta@estudante.ufjf.br
https://orcid.org/0000-0003-2140-1340

Mário Antônio Ribeiro Dantas
mario.dantas@ufjf.br
https://orcid.org/0000-0002-2312-7042

Marco Antônio Pereira Araújo
marco.araujo@ufjf.br
https://orcid.org/0000-0001-9504-6463

SUSTAINABLE BUSINESS INTERNATIONAL JOURNAL
Revista científica - ISSN 1807-5908
Avaliado pelo sistema Double Blind Review
Disponivel em https://periodicos.uff.br/sbijournal
PPGAd uff

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.

Palavras-chave: compressão de imagens; autoencoder; UAV; reconstrução visual; eficiência computacional; sustentabilidade computacional.

ABSTRACT

This study evaluates the effectiveness of autoencoders (conventional, variational, and redundancy-penalized) in compressing aerial UAV images, focusing on latency-constrained embedded applications. Using the SARD 2 dataset, the models were analyzed for reconstruction quality (PSNR, SSIM, and MS-SSIM) and processing time. The results refuted the central hypothesis, demonstrating that the optimized conventional autoencoder outperformed more complex models, achieving the best image quality while maintaining competitive latency. The work concludes that, in constrained environments, optimized structural simplicity can be more effective than theoretical complexity.

Keywords: image compression; autoencoders; UAV; visual reconstruction; computational efficiency; computational sustainability.

1 INTRODUÇÃO

A crescente utilização de Veículos Aéreos Não Tripulados (UAVs) – ou drones – em diversos contextos (agricultura, segurança pública e busca/salvamento) gera grandes volumes de dados visuais. Nesses cenários, a latência (atraso na transmissão) e a qualidade da imagem são fatores críticos, afetando a rapidez na tomada de decisão. (Ramos et al., 2023; Teng et al., 2025).

Embora JPEG e JPEG2000 sejam amplamente utilizados, em cenários de tempo real e alta compressão, eles podem introduzir artefatos e degradar regiões de interesse (ROI), devido a aplicação de transformações e quantização uniformes. Isso ocorre porque técnicas baseadas em transformadas matemáticas, como a Transformada Discreta de Cosseno (DCT) e a Transformada Wavelet Discreta (DWT), tendem a descartar informações consideradas menos relevantes, o que pode resultar em artefatos visuais e degradação perceptível da imagem (Potlapalli & Khetavath, 2025;Laakom et al., 2024; Huang & Wu, 2024). Além disso, em aplicações críticas, a necessidade de preservar regiões de interesse com alta qualidade torna os métodos clássicos ainda menos adequados.

Por essa razão, abordagens baseadas em aprendizado de máquina, como autoencoders e abordagens sensíveis à Região de Interesse (ROI), têm demonstrado melhor preservação de detalhes sob restrições de *bitrate. Autoencoders* com complexidade reduzida, por exemplo, já foram aplicados com sucesso em compressão de imagens de satélite, superando padrões tradicionais e mantendo desempenho competitivo mesmo sob restrições de tempo e memória. (Alves de Oliveira et al., 2021). De fato, abordagens mais recentes, baseadas em aprendizado de máquina (autoencoders, Redes Adversárias Generativas - GANs) e compressão por região de interesse (ROI), têm se destacado por oferecer melhor preservação de detalhes e maior eficiência, especialmente em baixas taxas de bits. No entanto, a aplicação em cenários de tempo real ainda pode ser limitada pelo custo computacional dessas técnicas (Marchenko et al., 2024; Ungureanu et al., 2024).

Nestes ambientes, a operação de redução de dados pode se basear em técnicas de inteligência artificial e aprendizado profundo, especificamente o uso dos *autoencoders.* (Laakom et al., 2024; Huang & Wu, 2024). É fundamental salientar que o foco deste trabalho é a comparação entre as arquiteturas de *autoencoders*. Isso se justifica pelas limitações do JPEG e JPEG2000 em cenários de *edge computing* com alta compressão crítica, que as tornam inerentemente subótimas.

O presente estudo se justifica pela crescente demanda por soluções que conciliam qualidade visual, desempenho computacional e aplicabilidade em contextos restritos, como sistemas embarcados em robôs aéreos. Conforme o levantamento realizado na literatura, percebem-se lacunas de análises abrangentes e integradas, que avaliem os principais indicadores de desempenho que são essenciais para aplicações de veículos aéreos autônomos. Nesse sentido, este artigo propõe ampliar o conhecimento científico sobre compressão de imagens, indicando caminhos práticos para melhorias na eficiência energética e na de redução de tráfego dos dados, apoiando-se no conhecimento geral acerca da eficiência em sistemas *edge* e Internet das Coisas (IoT). (Barbuto et al., 2023; Pioli et al., 2024).

Embora estudos recentes tenham avançado na proposição de arquiteturas específicas, como *autoencoders* variacionais (VAE), *denoising* e com penalização de redundância latente (Subburaj & Bhavana, 2024; Zhu, 2024), revisões sistemáticas e meta-análises, indicam a escassez de comparações que avaliem, conjuntamente, o impacto dessas arquiteturas nos principais aspectos de desempenho. Além disso, os trabalhos existentes costumam privilegiar cenários genéricos de IoT, com poucos focados especificamente nos ambientes operacionais críticos de drones, onde restrições energéticas, limitações no enlace de comunicação e variabilidade ambiental tornam ainda mais complexos esses desafios. (Pioli Junior, 2024; Teng et al., 2025).

Diante desse panorama, o presente estudo propõe investigar a seguinte questão de pesquisa: dentre os modelos de *autoencoders*, especificamente, os convencionais, VAE e com penalização de redundância, existe algum que resulte em um desempenho superior, sob a perspectiva das principais métricas operacionais na transmissão de imagens de UAVs em missões de busca e resgate (SAR)?

Como hipótese central, supõe-se que autoencoders com penalização de redundância latente apresentem melhor equilíbrio entre compressão, fidelidade e tempo de processamento em ambientes SAR.

A fim de responder a essa questão e testar a hipótese formulada, este trabalho pretende analisar comparativamente o desempenho de diferentes arquiteturas de *autoencoders* na transmissão de imagens de robôs aéreos, considerando-se a latência de ponta a ponta, a taxa de compressão e a qualidade de reconstrução.

Para operacionalizar esse propósito, definiram-se os seguintes objetivos específicos:

— OE1: investigar os fundamentos teóricos e as abordagens existentes de compressão de imagens baseadas em *autoencoders* aplicadas a sistemas de drones.

— OE2: implementar versões funcionais de *autoencoders* convencionais, VAE e com penalização de redundância em um ambiente controlado de simulação.

— OE3: interpretar os resultados obtidos quanto ao impacto das diferentes arquiteturas sobre a eficiência da compressão e a qualidade visual das imagens reconstruídas.

As principais contribuições deste trabalho são:

— Avaliação experimental comparativa de três arquiteturas de autoencoders aplicadas à compressão de imagens de UAVs;

— Análise multifatorial baseada em latência, taxa de compressão e qualidade de reconstrução;

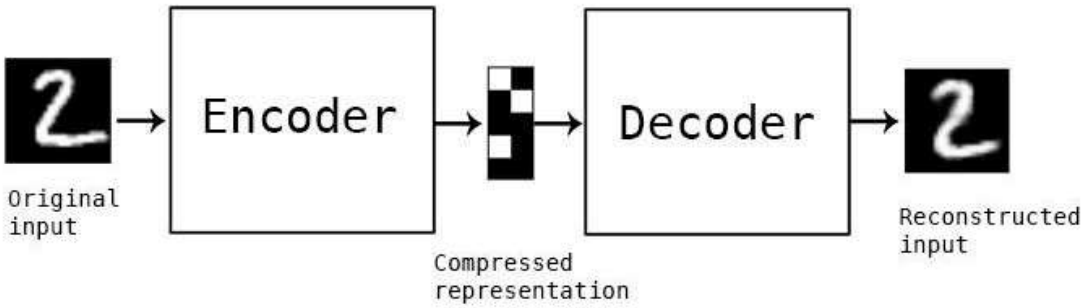
— Proposição de um pipeline metodológico replicável para pesquisas futuras na área.

2 REVISÃO DA LITERATURA

2.1. Fundamentos de *autoencoders*

Autoencoders são redes neurais não supervisionadas projetadas para aprender representações compactas dos dados por meio de uma arquitetura composta por duas partes principais: o encoder, responsável por reduzir a dimensionalidade da entrada e gerar um vetor latente, e o decoder, que reconstrói a entrada original a partir desse vetor. O objetivo do treinamento é minimizar a diferença entre a entrada e sua reconstrução, promovendo compressão e retenção das características essenciais do conjunto de dados (Goodfellow, Bengio, & Courville, 2016). Variações modernas, como *autoencoders* esparsos, *denoising* e variacionais (VAE), expandem esse princípio ao introduzir penalidades adicionais ou ruídos durante o treinamento, criando representações latentes mais robustas e apropriadas para diferentes finalidades (Goodfellow et al., 2016), onforme apresentado na Figura 1 abaixo.

Figura 1: Estrutura básica de um *autoencoder*



Fonte: Keras Blog

2.2. Compressão de Imagens com *Autoencoders*

No contexto da compressão de imagens, os *autoencoders* vêm ganhando destaque pela flexibilidade de suas arquiteturas e pela capacidade de adaptação a diferentes requisitos de aplicação. Estudos recentes detalham o desempenho de variantes como as convencionais, VAE e penalizadas, em cenários com alta dinâmica de conteúdo visual. (Zhu, 2024; Armand et. al, 2024; Duan et al., 2023; Tyagi et al., 2023)

Aplicações recentes demonstram que a escolha da arquitetura do *autoencoder* influencia diretamente o equilíbrio entre taxa de compressão e fidelidade visual das imagens reconstruídas. Enquanto modelos convencionais priorizam a minimização do erro de

reconstrução, variantes como o VAE buscam maior generalização, já os modelos penalizados por redundância focam na capacidade informacional, como evidenciado em experimentos recentes. (Armand et. al, 2024; Xiao et al., 2024; Duan et al., 2023)

2.3. Latência, Qualidade de Reconstrução e Métricas de Avaliação

No contexto de aplicações com UAVs, a eficiência de transmissão de imagens não pode ser avaliada exclusivamente pela taxa de compressão. A latência ponta a ponta, definida como o tempo total entre a aquisição, compressão, transmissão, descompressão e entrega da imagem reconstruída, é um fator crítico em aplicações em tempo real, como resgates e monitoramento ambiental (Ramos et al., 2023; Teng et al., 2025).

Para avaliar o desempenho das arquiteturas de compressão, utilizam-se métricas objetivas como o PSNR (*Peak Signal-to-Noise Ratio*), SSIM *(Structural Similarity Index*) e o MS-SSIM *(Multi-Scale Structural Similarity),* 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, 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.

2.4. Redução de Dados em Drones: Sustentabilidade Computacional

As operações com robôs aéreos impõem restrições significativas de energia, capacidade computacional embarcada e largura de banda. Como apontam Pioli et al. (2024), a aplicação de técnicas de redução de dados no ambiente de borda pode contribuir para a diminuição da carga de transmissão, resultando em ganhos operacionais e energéticos importantes. A compressão inteligente em robôs aéreos pode, assim, atuar como fator de sustentabilidade computacional, ampliando a autonomia de voo e a cobertura operacional.

Estudos recentes, como o de Barbuto et al. (2023), apontam que, apesar dos avanços consideráveis na área de *edge computing*, persistem lacunas significativas quanto à análise integrada de latência, compactação e qualidade/reconstrução visual, especialmente em ambientes autônomos críticos. Já Pioli Junior (2024) destaca que propostas de compressão sensível ao contexto, embora promissoras, ainda carecem de validação prática em cenários como, por exemplo, em redes de drones. Essas constatações reforçam a necessidade de estudos comparativos focados na avaliação experimental de arquiteturas otimizadas para tais ambientes. Estudos complementares também discutem estratégias de aceleração de inferência em dispositivos de borda, c*omo edge–cloud collaborative inference e network*

pruning, que podem ser integradas a futuros modelos (Li et al., 2023; Liu et al., 2023; Shuvo et al., 2022)

Dessa forma, embora avanços significativos tenham sido observados, ainda faltam estudos comparativos experimentais voltados a UAVs com restrições computacionais reais, justificando o presente trabalho.

3 METODOLOGIA

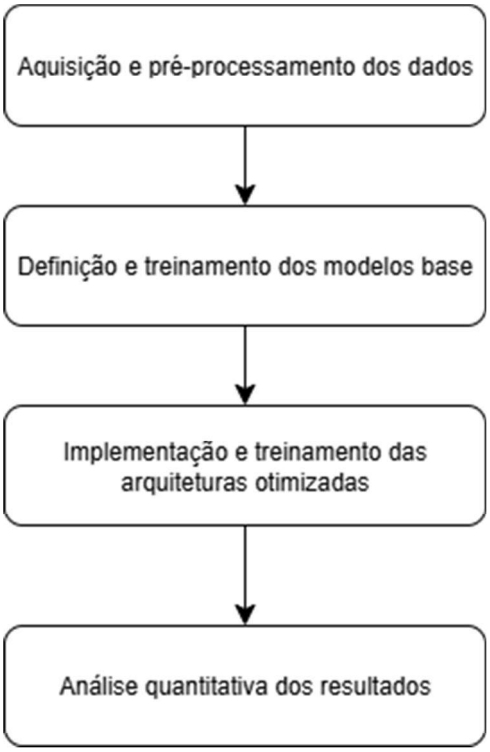
Este estudo caracteriza-se como uma pesquisa experimental aplicada, voltada à comparação do desempenho de diferentes arquiteturas de *autoencoders* em tarefas de compressão de imagens oriundas de drones. A abordagem experimental foi selecionada por permitir a implementação prática dos modelos, o controle das variáveis e a mensuração objetiva dos resultados, possibilitando a geração de evidências empíricas com potencial aplicabilidade em cenários reais de monitoramento aéreo.

As etapas metodológicas seguem uma sequência estruturada que contempla desde o pré-processamento dos dados até a análise crítica dos resultados. Inicialmente, as imagens são submetidas ao pré-processamento, incluindo redimensionamento e normalização. Em seguida, são implementadas três arquiteturas distintas de *autoencoder*: convencional, variacional (VAE) e com penalização de redundância. Cada modelo é treinado até a convergência da função de perda, e as imagens de teste são então comprimidas e reconstruídas por cada arquitetura. São registradas métricas de taxa de compressão, PSNR, SSIM, MS-SSIM e latência média de processamento para comparação quantitativa. Foram realizados ajustes estruturais e de regularização nos modelos para garantir estabilidade

econvergência. Por fim, análise experimental dos resultados, considerando aspectos estatísticos e operacionais.

O fluxo metodológico completo pode ser visualizado na Figura 2, que ilustra as principais etapas do experimento, desde o recebimento dos dados de entrada até a análise final dos resultados.

Figura 2: *Pipeline* metodológico para avaliação comparativa das arquiteturas de *autoencoders*



Fonte: elaborado pelo autor

3.1. Tipo de Pesquisa

A pesquisa é classificada como experimental, com delineamento quantitativo comparativo, fundamentado na reprodução de experimentos e na análise quantitativa de métricas de desempenho. Para a contextualização conceitual e a demonstração da relevância do tema, foi realizada uma pesquisa bibliográfica seletiva e crítica, baseada em revisões sistemáticas sobre inteligência de borda e redução de dados em sistemas distribuídos (Barbuto et al., 2023; Pioli et al., 2024), que destacam a necessidade de soluções eficientes e inteligentes de processamento de dados em ambientes de *edge computing*. Por outro lado, referências específicas de estudos experimentais sobre compressão e arquiteturas de *autoencoders*

foram utilizadas para embasar o desenho experimental, tais como Ramos et al. (2023), Laakom et al. (2024) e Zhu (2024).

3.2. Aquisição de 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. Este conjunto de dados contém imagens de alta resolução capturadas por drones 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, serão selecionadas amostras balanceadas entre as classes e cada imagem será redimensionada e normalizada conforme os requisitos dos modelos.

3.3. Pré-processamento dos dados

Para adequar o dataset SARD 2 ao cenário de *edge computing* com restrições de *hardware* em UAVs, o pré-processamento das imagens originais (1920x1080 px) envolveu três etapas. Primeiramente, as imagens foram redimensionadas (*downscaling*) para a dimensão de entrada de 128×128×3 px. Esta escolha é fundamental para simular as restrições de *hardware* e memória de sistemas embarcados, sendo compatível com a faixa de dimensões adotadas em estudos restritivos (Alves de Oliveira et al., 2021). Testes exploratórios confirmaram a escolha de 128×128 px, pois resoluções maiores não justificavam o acréscimo de complexidade e latência. Em seguida, o ordenamento dos canais de cor foi ajustado de BGR para RGB. Por fim, os dados foram submetidos à Normalização Min-Max (divisão por 255), um procedimento crucial para garantir a estabilidade e a velocidade de convergência do treinamento.

3.4. Ambiente e procedimentos de implementação

A implementação dos experimentos iniciais, contemplando o treinamento dos modelos e avaliação dos resultados, foi realizada em ambiente virtual de nuvem, que possui maior capacidade computacional. Essa escolha se deve à indisponibilidade 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, em trabalhos futuros, a inferência será direcionada para ambientes de borda, visando eficiência computacional e redução de tráfego de dados, conforme abordado na literatura relacionada (Azizian & Bajić, 2024; Bao et al., 2023; Yamazaki et al., 2022). 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 ponta a ponta, prática adotada em estudos similares (Zhang et al., 2024; Ramos et al., 2023).

A tabela abaixo apresenta um resumo com as principais configurações do ambiente para facilitar a reprodutibilidade dos experimentos, as demais bibliotecas utilizadas (Tensorflow, Keras, Numpy, etc) podem ser consultadas diretamente no *notebook* disponível no Github.

Tabela 1: Resumo das Configurações do servidor de aplicação

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: do autor

Foram implementadas as arquiteturas de *autoencoder* (convencional, VAE e com penalização de redundância latente), treinamento até convergência e avaliação experimental. Os hiperparâmetros iniciais estão detalhados na Tabela 2. Os resultados são analisados com base nas principais 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:{removido para blind review}.

SUSTAINABLE BUSINESS INTERNATIONAL JOURNAL
Revista científica - ISSN 1807-5908
Avaliado pelo sistema Double Blind Review
Disponivel em https://periodicos.uff.br/sbijournal
PPGAd uff

Tabela 2: Hiperparâmetros usados nos modelos iniciais de *autoencoders*

Hiperparâmetro	Convencional	Variacional (VAE)	Penalização de Redundância
Input shape	(128×128×3)	(128×128×3)	(128×128×3)
Arquitetura Encoder	Conv2D 32 → MaxPool → Conv2D 16 → MaxPool → Conv2D 16	Conv2D 32 → MaxPool → Conv2D 16 → MaxPool → Flatten	Conv2D 32 → MaxPool → Conv2D 16 → MaxPool
Camadas extras	Upsampling → Conv2D 16 → Upsampling	Flatten → Dense(μ, log_var) → Lambda (sampling)	Flatten → Dense(256, L1=1e-5)
Latent space	Bottleneck espacial (implícito)	Dense(64) + Sampling estocástico	Dense(256) com L1
Decoder / Rebuild	Upsampling → Conv2D 16 → Upsampling → Conv2D 3	Dense → Reshape → Upsampling → Conv2D 3	Dense → Reshape → Upsampling → Conv2D 3
Função de ativação	ReLU (interm.) + Sigmoid (saída)	ReLU (interm.) + Sigmoid (saída)	ReLU (interm.) + Sigmoid (saída)
Regularização	—	—	L1 (1e-5) na Dense
Latent dim / penal dim	implícito via pooling	64	256
Optimizer	Adam (lr=1e-3)	Adam (lr=1e-3)	Adam (lr=1e-3)
Loss	MSE	MSE	MSE

Fonte: do autor

A Tabela 2 detalha os principais hiperparâmetros e arquiteturas iniciais dos modelos avaliados. Estruturalmente, o encoder utiliza duas camadas convolucionais seguidas de *MaxPooling*, e o decoder emprega *Upsampling* e *Conv2D* para a reconstrução. O espaço latente foi modelado implicitamente no autoencoder convencional, com 64 neurônios no VAE (acompanhado de amostragem estocástica) e 256 neurônios no modelo penalizado por redundância, que inclui regularização L1 (λ=1e−5). As funções de ativação foram ReLU (intermediárias) e Sigmoid (saída). A otimização dos modelos foi realizada utilizando o algoritmo Adam com taxa de aprendizado de 1e−3 e função de perda MSE, seguindo as práticas comuns da literatura (Goodfellow et al., 2016; Laakom et al., 2024)

4 RESULTADOS EXPERIMENTAIS E DISCUSSÃO

4.1. Descrição dos resultados

Os experimentos realizados permitiram avaliar o desempenho das arquiteturas de *autoencoders* na tarefa de compressão e reconstrução de imagens provenientes de drones em cenários de busca e salvamento. Os resultados quantitativos das métricas analisadas, PSNR médio, SSIM médio, MS-SSIM médio e tempo médio de processamento, são apresentados na Tabela 3, sendo ilustrados comparativamente na Figura 3 a seguir.

Esses indicadores foram escolhidos não apenas pelo valor técnico, mas pela relevância direta para missões reais de busca e salvamento, em que cada milissegundo de latência e cada ganho de fidelidade visual podem significar uma resposta mais rápida e eficaz em campo.

Tabela 3: Resultados Comparativos para as arquiteturas avaliadas

Modelo	PSNR	SSIM	MS-SSIM	Tempo (s)	Principais parâmetros de Otimização
Convencional (Base)	17.668	0.5466	0.8675	0.2507	---
VAE (Base)	15.0193	0.1373	0.4428	0.2656	---
Redundância (Base)	12.2808	0.0609	0.2114	0.27	---
Convencional Otimizado (FINAL)	20.7116	0.801	0.9359	0.2654	filter_max: 64 Batch Normalization kernel_regularizer=l2(1e-4) Conv2DTranspose Arquitetura assimétrica
VAE Otimizado (FINAL)	13.4047	0.0743	0.2615	0.28	beta=0.01, L=128
Redundância Otimizado (FINAL)	14.2196	0.0983	0.3329	0.2868	l1_reg=1e-6, L=512

Fonte: do autor

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(1e-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=1e−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.

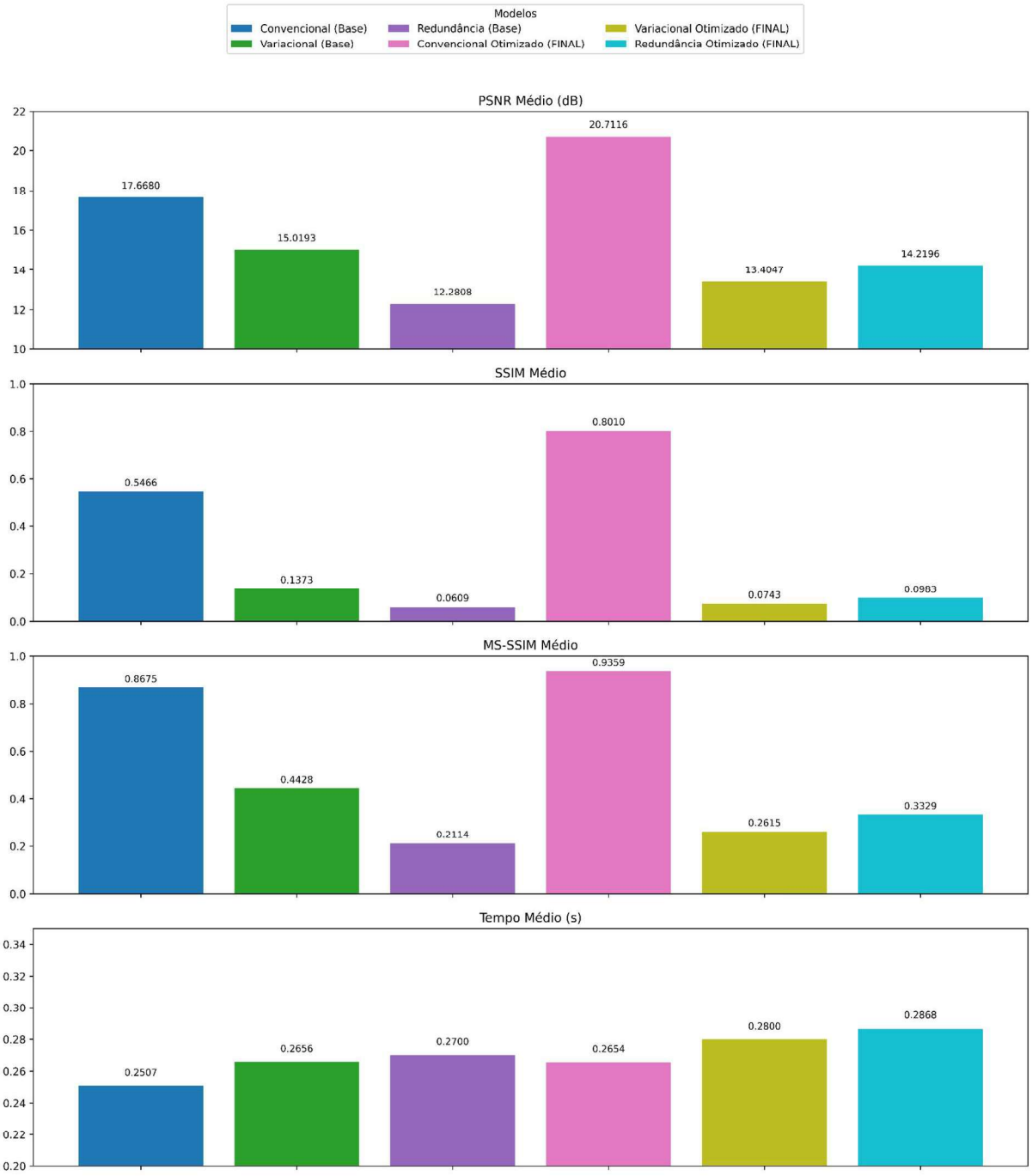
No treinamento o aumento do número de épocas para 800 foi o mais assertivo para os modelos, foram utilizados *callbacks* (*Learning Rate Scheduler e EarlyStopping*) para ajustar dinamicamente a taxa de aprendizado e interromper o treinamento quando não houvesse mais melhoria na validação, prevenindo 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 observado na Tabela 3.

SUSTAINABLE BUSINESS
INTERNATIONAL JOURNAL
Revista científica - ISSN 1807-5908
Avaliado pelo sistema Double Blind Review
Disponivel em https://periodicos.uff.br/sbijournal
PPGAd-uff

Figura 3: Comparação gráfica das métricas entre os modelos base e otimizados



Fonte: do autor

A avaliação comparativa dos modelos de autoencoders 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 da Tabela 3, foram testados, os seguintes parâmetros (beta=0.1,L=128; beta=0.01,L=256) para o modelo VAE e (l1_reg=1e−6,L=512; l1_reg=5e−7,L=512; l1_reg=1e−6,L=1024) com resultados menores do indicados na Tabela 3 que foram desconsiderados.Além disso testes com a função de perda combinada (MSE + SSIM) e o uso de (Keras Tuner/Optuna) foram explorados na busca de hiperparâmetros, mas não forami considerados na análise final, devido a dificuldades de salvamento e carregamento dos modelos VAE. Todos os resultados estão no repositório do experimento dentro da pasta results.

4.2. Análise de desempenho em qualidade de reconstrução (PSNR, SSIM e MS-SSIM)

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. 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 pior desempenho, com um PSNR de apenas 12,2808 dB, indicando que, embora a otimização penalize a redundância do manifold, ela comprometeu severamente a fidelidade da imagem. A otimização implementada na arquitetura Convencional (que subiu de 17,6680 dB para 20,7116 dB) foi a que teve mais ganho de qualidade expressivo. Apesar de ser o melhor entre os modelos testados, o valor ainda é um resultado preliminar para este contexto, podendo refletir, entre outros, o *downscaling*, tornando a qualidade final apenas aceitável para inspeção humana, podendo ser evoluída em estudos complementares.

Trabalhos conduzidos em condições distintas de dataset, resolução e taxa de bits reportam PSNR absolutos mais elevados (por exemplo, ≈40 dB em cenários específicos). No presente estudo, o objetivo não é maximizar PSNR absoluto, mas avaliar o equilíbrio entre qualidade e custo computacional sob restrições típicas de edge (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 UAV/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, teste

estatísticos adicionais não foram necessários, uma vez que as variações entre as arquiteturas foram claramente observáveis e expressivas.

4.3. Análise de desempenho em eficiência computacional (Tempo de Processamento)

A latência média para o processo de compressão e reconstrução "tempo (s)" mostrou pouca variação entre todas as arquiteturas. Os tempos de processamento variaram em uma faixa estreita, entre 0,2507 s (Convencional Base) e 0,2868 s (Redundância Otimizado FINAL).

Apesar das diferentes complexidades e funções de perda (VAE e Redundância), não houve uma diferença de latência significativa que justificasse a escolha de um modelo em detrimento da qualidade. O modelo de melhor qualidade, Convencional Otimizado, manteve o tempo de processamento em 0,2654 s, comparável ao modelo mais rápido (0,2507 s).

5 CONCLUSÃO E TRABALHOS FUTURO

O objetivo central deste estudo foi avaliar e comparar, em condições restritivas de *edge computing*, a eficácia de diferentes arquiteturas de *autoencoders* (Convencional, Variacional e Penalizado por Redundância) na redução de dados visuais para operações de Veículos Aéreos Não Tripulados (UAV) em missões de Busca e Resgate (SAR). A análise multifatorial foi realizada sob a perspectiva das principais métricas operacionais: qualidade de reconstrução (PSNR, SSIM, MS-SSIM) e eficiência computacional (Tempo de Processamento).

5.1. Síntese dos Resultados Chave

Os resultados quantitativos refutaram categoricamente a hipótese inicial, que postulava a superioridade das arquiteturas que empregam mecanismos explícitos de penalização de redundância no espaço latente. A análise demonstrou que o *autoencoder* convencional otimizado foi a solução mais eficaz, superando os modelos mais complexos em qualidade visual sem comprometer a latência.

O modelo vencedor alcançou o maior desempenho em todas as métricas de qualidade, registrando um PSNR de 20,7116 dB e um MS-SSIM de 0,9359. Este resultado representa um aumento de mais de 6 dB de qualidade de reconstrução em comparação com o *autoencoder* Variacional Otimizado (13,4047 dB), e de aproximadamente 6,5 dB em relação ao Redundância Otimizado (14,2196 dB).

Essa diferença é operacionalmente crítica. A queda abrupta na qualidade dos modelos VAE e de Redundância Otimizado sugere que a penalização e a regularização do espaço latente, embora busquem representações esparsas, falharam em reter os detalhes visuais essenciais para a reconstrução de alta fidelidade do cenário SAR. Consequentemente, as imagens reconstruídas tornaram-se potencialmente inadequadas para a identificação confiável de alvos e vítimas. Tais resultados demonstram desempenho competitivo com abordagens

mais complexas encontradas na literatura recente, como as de Laakom et al. (2024) e Marchenko et al. (2024), alcançando qualidade de reconstrução semelhante com menor complexidade estrutural e custo computacional reduzido. Essa relação entre simplicidade e eficiência evidencia o potencial do modelo proposto para aplicações embarcadas em UAVs.

Adicionalmente, a latência de processamento (utilizada como *proxy* da latência ponta a ponta) mostrou pouca variação entre todos os modelos otimizados (entre 0,2654 s e 0,2868 s). Tal estabilidade confirma que, no cenário de compressão extrema adotado, o fator decisivo para a escolha da arquitetura em sistemas embarcados é, inequivocamente, a qualidade da reconstrução.

O diferencial desta abordagem reside no ajuste fino estrutural; a incorporação de *Batch Normalization,* um *kernel regularizer* e uma arquitetura assimétrica transformou o modelo convencional em uma solução de compressão de alto impacto. Isso prova que a otimização de arquiteturas leves é a chave para sistemas embarcados de drones. Esses resultados são particularmente relevantes para sistemas embarcados em UAVs, onde limitações de energia, largura de banda e custo de *hardware* podem inviabilizar modelos complexos. Assim, o autoencoder otimizado proposto oferece uma alternativa realista para implementação em missões SAR, com redução de latência e manutenção de qualidade visual. Em termos relativos, o modelo otimizado apresentou ganhos de mais de 50% nas métricas estruturais em relação às versões penalizadas, sem aumento perceptível no tempo de processamento. Dessa forma, a arquitetura proposta equilibra qualidade e desempenho, configurando-se como uma base promissora para aplicações de compressão embarcada em drones e dispositivos IoT.

5.2. Conclusão e Contribuições

Este estudo conclui que, nas condições de *edge computing* simuladas para missões SAR, o *autoencoder* convencional otimizado é o ponto de equilíbrio ideal entre qualidade perceptual e eficiência computacional. A superioridade deste modelo reforça o achado de que, em ambientes restritos, a simplicidade estrutural otimizada pode superar a complexidade teórica inerente aos modelos VAE e de Redundância.

As principais contribuições do trabalho são:

— Estabelecimento de um Novo *Baseline*: O *autoencoder* convencional otimizado, com seus resultados de 20,7116 dB e latência competitiva, deve ser considerado o novo *baseline* de referência para futuras investigações em compressão de imagens de UAV/SAR.

— Análise Operacional Integrada: O estudo preenche uma lacuna na literatura ao comparar as arquiteturas com foco em métricas operacionais (qualidade versus latência), oferecendo *insights* práticos para o desenvolvimento de soluções embarcadas.

— Metodologia Replicável: O trabalho fornece um *pipeline* experimental totalmente detalhado e replicável, crucial para o avanço de pesquisas em compressão aprendida para

visão computacional aérea.

A comparação de PSNR entre estudos exige cautela, pois diferenças de resolução, *datasets*, pipeline de pré-processamento e faixa de *bitrate* impactam diretamente os valores absolutos. O desenho experimental aqui adotado prioriza restrição de recursos e latência, razão pela qual os resultados devem ser interpretados pelo ganho relativo entre arquiteturas sob a mesma configuração, e não por recordes absolutos de PSNR.

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 UAV civis ou de defesa.

5.3. Desafios encontrados e Trabalhos Futuros

O principal desafio metodológico do estudo foi a indisponibilidade de uma infraestrutura de *edge computing* real, que impediu a medição da latência de transmissão de ponta a ponta. O tempo de processamento foi utilizado como proxy da latência, sendo necessário avançar para testes em modos reais de trabalho.

Com base nas conclusões e desafios, os trabalhos futuros sugeridos incluem:

— Realizar uma comparação direta dos modelos otimizados com *baselines* clássicas de compressão (JPEG e JPEG2000), a fim de quantificar o real ganho da compressão aprendida.

— Utilizar de soluções como o Keras Tuner, Optuna e Keras Quartenion adaptados para os modelos mais complexos como o VAE e redundância, bem como uso funções de perdas combinadas ajustadas aos modelos.

— Explorar outros modelos de *autoencoders*, como *Autoencoder* Esparso (SAE), *Autoencoder* Contrativo (CAE).

— Conduzir testes em ambientes operacionais reais de borda, para medir o impacto da latência de transmissão (largura de banda e jitter) no desempenho de ponta a ponta.

— Avaliar o impacto da escalabilidade da resolução na qualidade de reconstrução (ex. 256×256 px), caso o poder de processamento do *hardware* embarcado seja aprimorado.

— Verificar os impactos da redução de energia e explorar abordagens mistas ou modelos autoadaptativos para melhorar a robustez das soluções em campo.

— Traçar curvas taxa-distorção (RD) e BD-Rate em diferentes bitrates e resoluções (incluindo 256×256 e 512×512), comparando com JPEG/JPEG2000 e baselines neurais reportados na literatura, de modo a posicionar o modelo vencedor em métricas absolutas sem perder a ótica de latência.

— Avaliar qualidade perceptual (ex.: Mean Opinion Score (MOS) com avaliadores humanos ou métricas perceptuais adicionais) para complementar PSNR/SSIM em

condições extremas de compressão.

Em síntese, este trabalho reforça que a otimização estrutural de modelos leves constitui uma estratégia promissora para aplicações embarcadas de compressão visual em UAVs.

REFERÊNCIAS

Alves de Oliveira, V., Chabert, M., Oberlin, T., Poulliat, C., Bruno, M., Latry, C., & Camarero, R. (2021). Reduced-complexity end-to-end variational autoencoder for on board satellite image compression. Remote Sensing, 13(3), 447. https://doi.org/10.3390/rs13030447

Armand, A. K., Diédié, G. H. F., & Tchimou, N. T. E. (2024). Super-tokens Auto-encoders for image compression and reconstruction in IoT applications. IJASRE, 1, 1-4.

Azizian, B., & Bajić, I. V. (2024). Privacy-preserving autoencoder for collaborative object detection. IEEE Transactions on Image Processing.

Bao, X., Ye, C., Han, L., & Xu, X. (2023). Image Compression for Wireless Sensor Network: A Model Segmentation-Based Compressive Autoencoder. Wireless Communications and Mobile Computing, 2023(1), 8466088.

Barbuto, V., Savaglio, C., Chen, M., & Fortino, G. (2023). Disclosing edge intelligence: A systematic meta-survey. *Big Data and Cognitive Computing*, 7(1), 1–24. https://doi.org/10.3390/bdcc7010044

Bhagat, S., Pandey, M., Baheti, V., Goyal, M., & Sinha, M. (2024). Deep learning-based image classification using auto-encoders. In *Proceedings of the 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*. IEEE. https://doi.org/10.1109/ICDICI62993.2024.10810902

Chollet, F. (2016). Building autoencoders in Keras. *Keras Blog*. https://blog.keras.io/building-autoencoders-in-keras.html

Chen, J., Cai, H., Chen, J., Xie, E., Yang, S., Tang, H., Guo, Q., Huang, Y., Wang, X., & Han, S. (2024). Deep compression autoencoder for efficient high-resolution diffusion models. arXiv preprint arXiv:2410.10733. https://arxiv.org/abs/2410.10733

Duan, Z., Lu, M., Ma, J., Huang, Y., Ma, Z., & Zhu, F. (2023). Qarv: Quantization-aware resnet vae for lossy image compression. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(1), 436-450.

Gegenava, N. (2025). SARD 2 – search and rescue dataset extra classes [Data set]. Kaggle. https://www.kaggle.com/datasets/nikolasgegenava/sard-2-search-and-rescue-dataset-extra-classes

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. *MIT Press*. http://www.deeplearningbook.org

Huang, C.-H., & Wu, J.-L. (2024). Unveiling the Future of Human and Machine Coding: A Survey of End-to-End Learned Image Compression. Entropy, 26(5), 357. https://doi.org/10.3390/e26050357

Laakom, F., Raitoharju, J., Iosifidis, A., & Gabbouj, M. (2024). Reducing redundancy in the bottleneck representation of autoencoders. Pattern Recognition Letters, 178, 202-208.

Li, M., Zhang, X., Guo, J., & Li, F. (2023). Cloud–edge collaborative inference with network pruning. Electronics, 12(17), 3598.

Liu, G., Dai, F., Xu, X., Fu, X., Dou, W., Kumar, N., & Bilal, M. (2023). An adaptive DNN inference acceleration framework with end–edge–cloud collaborative computing. Future Generation Computer Systems, 140, 422-435.

Marchenko, I., Balalaieva, O., Korotenko, H., & Tarazanov, M. (2024). Research of image compression algorithms using neural networks. Reporter of the Priazovskyi State Technical University. Section: Technical Sciences, 1(49), 85–99. https://doi.org/10.31498/2225-6733.49.1.2024.321212

Pioli, L., Macedo, D. D. J., Costa, D. G., & Dantas, M. A. R. (2024). Intelligent edge-powered data reduction: A systematic literature review. *ACM Computing Surveys*, 56(9), Article 234, 1–39. https://doi.org/10.1145/3656338

Pioli Junior, L. (2024). *An intelligent context-aware edge-based data reduction framework for IoT* (Doctoral dissertation, Universidade Federal de Santa Catarina). https://repositorio.ufsc.br/handle/123456789/264055

Pioli, L., Macedo, D. D. J., Costa, D. G., & Dantas, M. A. R. (2025). Intelligent data reduction for IoT: A context-driven framework. *IEEE Access*, 13. https://doi.org/10.1109/ACCESS.2025.3586539

Potlapalli, A., & Khetavath, S. (2025). 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*, 15(1), 37–52.

Ramos, G. S., Lima, A. A., Almeida, L. F., Lima, J., & Pinto, M. F. (2023). Simulation and evaluation of deep learning autoencoders for image compression in multi-UAV network systems. In *LARS, SBR, WRE. IEEE*. https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332986

Reddi, V. J., Bansal, A., Ravichandran, D., et al. (2022). Machine learning systems: Designs that scale (1st ed.). O'Reilly Media.

Subburaj, T., & Bhavana, S. (2024). Image noise reduction with auto-encoders using TensorFlow. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, 4(4), 86–91. https://doi.org/10.48175/IJARSCT-19016

Shuvo, M. M. H., Islam, S. K., Cheng, J., & Morshed, B. I. (2022). Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. Proceedings of the IEEE, 111(1), 42-91.

Teng, A., Shan, X., Lu, J., Zha, J., & Zhu, J. (2025). Research on deep learning-based compression processing technology for UAV inspection images. In *2025 International Conference on Electrical Automation and Artificial Intelligence (ICEAAI)* (pp. 1395–1399). IEEE. https://doi.org/10.1109/ICEAAI64185.2025.10956570

Tyagi, A., Aditya, H., Khandelwal, R., Singh, J., Pal, Y., & Jaiswal, G. (2023, December). Autoencoder image compression for high compression and acceptable quality. In 2023 OITS International Conference on Information Technology (OCIT) (pp. 184-189). IEEE.

Ungureanu, V. I., Negirla, P., & Korodi, A. (2024). Image-compression techniques: Classical and "region-of-interest-based" approaches presented in recent papers. Sensors, 24(3), 791.

Xiao, P., Qiu, P., Ha, S. M., Bani, A., Zhou, S., & Sotiras, A. (2025). SC-VAE: Sparse coding-based variational autoencoder with learned ISTA. Pattern Recognition, 161, 111187.

Yamazaki, M., Kora, Y., Nakao, T., Lei, X., & Yokoo, K. (2022, October). Deep feature compression using rate-distortion optimization guided autoencoder. In 2022 IEEE International Conference on Image Processing (ICIP) (pp. 1216-1220). IEEE.

Zhang, H., Huang, S., Xu, M., Guo, D., Wang, X., Wang, X., ... & Wang, W. (2024). Large-scale measurements and optimizations on latency in edge clouds. IEEE Transactions on Cloud Computing

Zhu, Z. (2024). *A comparative study of different autoencoder architectures*. Dean & Francis. https://doi.org/10.61173/az0nqw17

Artigo submetido em: 14/07/2025

Avaliação (cega) 1ª rodada concluída em: 02/08/2025

Avaliação (cega) 2ª rodada concluída em: 05/11/2025