

Aplicação de Redes Neurais Binárias para a Identificação de Anomalias em Cabos Industriais

**Pedro Henrique Rodrigues da Silva¹, Marco Túlio Gontijo de Sousa¹,
Pedro Henrique Azevedo de Medeiros¹, Rodrigo de Oliveira Santos¹**

¹¹Curso de Ciência de Dados, Pontifícia Universidade Católica de Minas Gerais
Av. Dom José Gaspar, 500, Belo Horizonte, MG, 30535-610

Abstract. A detecção de anomalias em imagens industriais é um problema crítico para a Indústria 4.0, com aplicações que vão desde a inspeção de qualidade até a manutenção preditiva. No entanto, a implementação de soluções baseadas em deep learning em dispositivos com recursos limitados, como microcontroladores ou GPUs de baixo poder, ainda é um desafio. Este trabalho propõe a aplicação de redes neurais binárias (BNNs) para a detecção de anomalias, visando desenvolver modelos leves e eficientes que possam ser implementados em dispositivos de borda. A abordagem utiliza uma arquitetura Wide ResNet-20-10 binarizada, alcançando acurácia de 91,11% e F1-score de 90,55%. A relevância do tema é justificada pela necessidade de soluções que combinem alta precisão com baixo custo computacional, além de sua aplicabilidade em cenários industriais reais.

1. Introdução

A evolução da Indústria 4.0 tem impulsionado a integração de tecnologias como Internet das Coisas (IoT) e Inteligência Artificial (IA) em processos industriais, com o objetivo de aumentar a eficiência, reduzir custos operacionais e melhorar a qualidade dos produtos [Cheng et al. 2018]. Nesse cenário, a inspeção automatizada por meio de visão computacional desempenha um papel fundamental, permitindo a identificação precoce de falhas em componentes, como cabos industriais, com precisão superior à de abordagens manuais.

Entretanto, apesar dos avanços em modelos de aprendizado profundo, sua implementação em ambientes com recursos computacionais restritos ainda representa um desafio relevante. A maioria das redes neurais convolucionais modernas exige grande capacidade de processamento e memória, o que inviabiliza sua execução em dispositivos embarcados, como microcontroladores ou GPUs de baixo consumo.

Neste contexto, redes neurais binárias (BNNs) surgem como uma alternativa promissora. Ao representar pesos e ativações com apenas dois valores possíveis (+1 ou -1), essas redes reduzem significativamente a complexidade computacional e o uso de memória, tornando-se adequadas para aplicações em dispositivos de borda [Courbariaux et al. 2015, Rastegari et al. 2016]. No entanto, a binarização tende a comprometer a capacidade de generalização do modelo, especialmente em tarefas com alta complexidade visual, como a detecção de anomalias.

Este trabalho propõe a aplicação de uma arquitetura binarizada baseada na Wide Residual Network (WRN-20-10), seguindo a abordagem apresentada por McDonnell [McDonnell 2018], para a classificação de anomalias em imagens de cabos industriais. O objetivo é avaliar se, mesmo sob restrições numéricas severas, é possível manter um desempenho competitivo em tarefas de inspeção automatizada.

Os resultados obtidos demonstram que a arquitetura binarizada é capaz de alcançar acurácia de 91,11% e F1-score macro de 90,55% em um cenário de classificação binária, mantendo baixo custo computacional e reduzido consumo de memória, o que reforça sua viabilidade para implantação em sistemas embarcados.

Este artigo está organizado da seguinte forma: a Seção 2 discute os fundamentos teóricos das redes neurais binárias e técnicas de extração de embeddings; a Seção 3 descreve o conjunto de dados e os procedimentos metodológicos utilizados; a Seção 4 apresenta os resultados experimentais e métricas de desempenho; a Seção 5 detalha o custo computacional da abordagem proposta; por fim, a Seção 6 discute os principais achados e possíveis desdobramentos futuros.

2. Fundamentação Teórica

2.1. Redes Neurais Binárias

A binarização quantiza pesos e ativações para $\{-1, +1\}$, tornando as operações de convolução extremamente eficientes. Em uma CNN tradicional, a operação básica é:

$$z = \sigma(w \otimes a),$$

onde w e a são tensores de pesos e ativações, \otimes representa convolução e σ é a função de ativação. A aplicação de binarização utiliza:

$$Q_w(\mathbf{w}) = \alpha \mathbf{b}_w, \quad Q_a(\mathbf{a}) = \beta \mathbf{b}_a,$$

com $\mathbf{b}_w, \mathbf{b}_a \in \{\pm 1\}$ e escalares α, β . A propagação direta torna-se:

$$z = \sigma(\alpha \beta (\mathbf{b}_w \odot \mathbf{b}_a)),$$

onde \odot é o produto XNOR–popcount. A retropropagação utiliza o estimador direto (STE) para contornar o gradiente nulo, definindo:

$$\text{STE}(x) = \begin{cases} -1 & \text{se } x < -1, \\ x & \text{se } -1 \leq x \leq 1, \\ +1 & \text{se } x > +1. \end{cases}$$

2.2. Embeddings

Embeddings de imagens são representações matemáticas que traduzem características visuais de uma imagem em vetores numéricos de alta dimensão. Esses vetores capturam informações como formas, cores, texturas e até mesmo contextos semânticos, permitindo que algoritmos de aprendizado de máquina processem e analisem imagens de maneira eficiente.

A necessidade de utilizar embeddings de imagens surge especialmente em cenários onde o volume de dados visuais é grande e a análise manual seria inviável. Eles permitem que sistemas automatizados processem imagens de forma escalável e eficaz. A

técnica começou a ser amplamente utilizada com o advento de arquiteturas como AlexNet e VGGNet, que demonstraram a eficácia das CNNs em extrair características visuais relevantes. Desde então, modelos mais avançados, como ResNet e EfficientNet, têm refinado a criação de embeddings, tornando-os mais precisos e compactos.

Um embedding de imagem é um vetor de números que representa uma imagem em um espaço multidimensional. Cada número no vetor corresponde a um atributo da imagem, por exemplo, cor, forma, textura, etc. O vetor captura a essência da imagem e nos permite comparar com outras imagens. Embeddings também são muito conhecidos como descritores de imagem.

Os embeddings consistem nas principais características da imagem. Eles são amplamente utilizados em cenários onde não existe um número fixo de classes, quando há muitas classes, ou quando se deseja comparar, agrupar ou ranquear imagens visualmente semelhantes.

2.3. HOG (Histogram of Oriented Gradients)

Desenvolvido por Dalal e Triggs, o Histograma de Gradientes Orientados (HOG) é um descritor cuja função se dá através da distribuição (histograma) de gradientes orientados de uma imagem. O HOG é capaz de extrair bordas de um ou mais objetos, para depois enviá-las a um algoritmo classificador responsável por definir, por exemplo, se na imagem analisada há ou não uma face [Jr and da Silva 2022].

Gradientes representam a mudança de tonalidade no decorrer da imagem em uma determinada direção. Normalmente, uma mudança abrupta de cor representa uma transição entre objeto e fundo.

Além do HOG, muitos outros descritores se baseiam em gradientes orientados. No entanto, HOG se destaca por usar o conceito de *Shape of Context*, ou seja, a divisão da imagem em blocos, extraíndo histogramas de orientação de gradiente em cada célula. Essa abordagem aumenta a robustez na detecção de contornos estruturais [Jr and da Silva 2022].

2.4. LBP (Local Binary Patterns)

Para explicar o LBP, deve-se ter em mente o que é textura em uma imagem. Textura é definida pelas relações locais entre os pixels de uma imagem e capturada por meio de padrões binários. Esses padrões descrevem a variação de intensidade ao redor de cada pixel central e formam a base para a identificação das características texturais da imagem.

Desta maneira, o cálculo do operador LBP pode ser visualizado na Figura 1, onde o pixel central comanda a binarização das células ao redor. Aqueles maiores ou iguais ao valor central são convertidos em 1; caso contrário, 0. O código binário LBP deste pixel central é composto dessas marcações de maneira anti-horária. Ao final, o padrão binário é obtido convertendo o código em decimal [Ojala et al. 2002, Cai and Ying 2009].

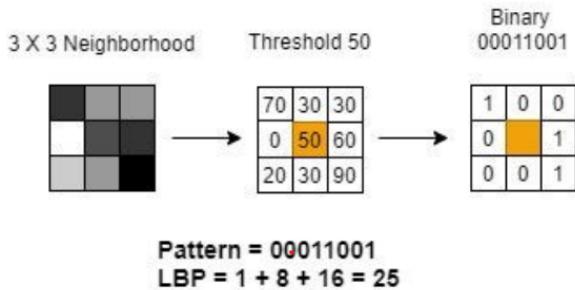


Figura 1. Representação de uma etapa para extração do valor do descriptor LBP.

Como pode ser visto na Figura 1, a grade 3×3 é o padrão para a ideia inicial da abordagem. Esse operador não é tão robusto em relação às variações de textura, causadas, por exemplo, por ponto de vista ou direção de iluminação. As variáveis calculadas em um raio 3×3 não capturam estruturas em larga escala. Dessa maneira, para contrapor essa deficiência, é introduzido o operador multi-resolução LBP.

O operador LBP multi-resolução [Ojala et al. 2002] recebe como parâmetro para cálculo o raio R de um círculo e os P pixels uniformemente distribuídos sobre ele, como ilustrado na Figura 2. Dessa forma, a influência de P e R deve ser considerada no desempenho do operador. Operadores com mais labels produzem histogramas mais longos e matrizes de distância mais pesadas computacionalmente; por outro lado, poucos labels podem resultar em perda de informação [Ojala et al. 2002, Cai and Ying 2009].

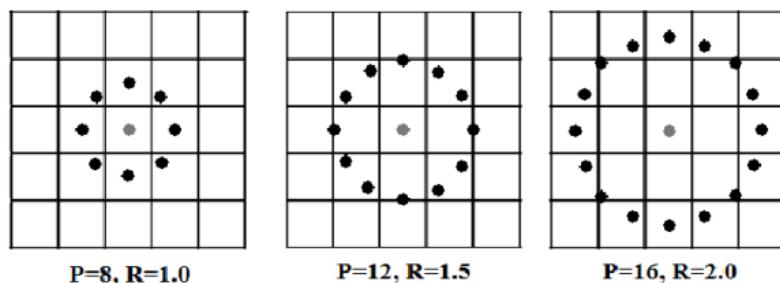


Figura 2. Três operadores LBP circulares simétricos com diferentes raios.

2.5. Análise Visual com UMAP, PCA e t-SNE

Com o objetivo de analisar a complexidade da base de dados e avaliar a capacidade dos descritores manuais em separar as classes visuais, aplicamos técnicas de redução de dimensionalidade sobre os embeddings gerados por HOG e LBP.

As técnicas utilizadas foram PCA (Análise de Componentes Principais), UMAP (Uniform Manifold Approximation and Projection) e t-SNE (t-distributed Stochastic Neighbor Embedding), conforme ilustrado na Figura 3. Cada ponto representa uma imagem, e as cores indicam as diferentes classes de cabos presentes na base de dados.

Observa-se que os descritores clássicos (HOG e LBP) apresentam baixa separabilidade entre as classes, especialmente em projeções lineares como a PCA. Isso indica que, apesar das imagens pertencerem a diferentes classes, elas compartilham características visuais bastante semelhantes, dificultando a distinção entre os grupos com base apenas em informações locais como bordas e texturas.

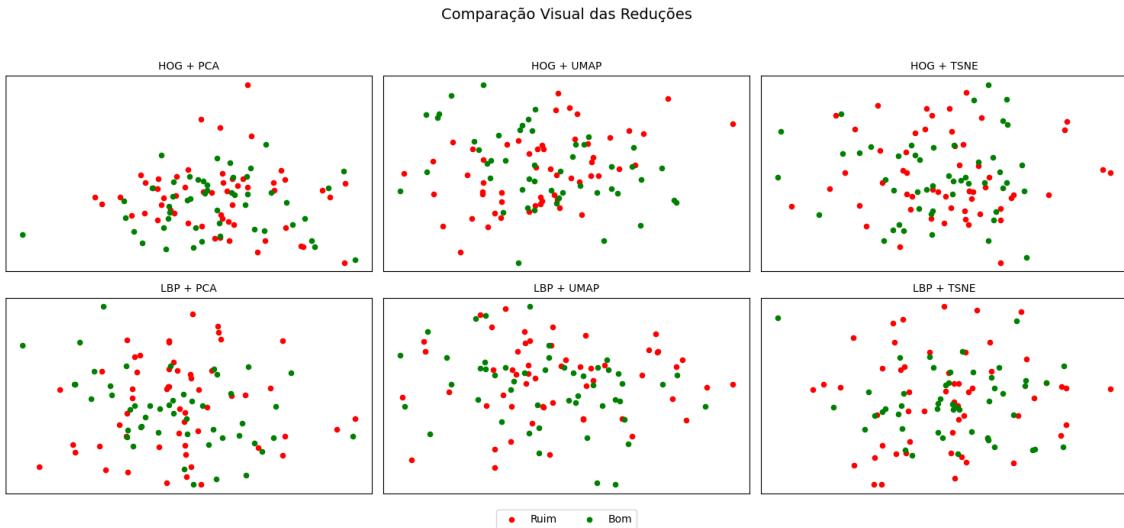


Figura 3. Projeção dos embeddings HOG e LBP em duas dimensões usando PCA, UMAP e t-SNE.

3. Metodologia

A metodologia proposta neste trabalho está estruturada em três grandes etapas: preparação e organização do conjunto de dados, treinamento da rede neural binarizada e avaliação do desempenho do modelo treinado. O objetivo é investigar a viabilidade da utilização de uma rede WRN-20-10 binarizada na tarefa de detecção de anomalias visuais em cabos industriais.

3.1. Etapa 1 – Descrição do Conjunto de Dados

As imagens utilizadas foram extraídas do MVTec Anomaly Detection Dataset [Bergmann et al. 2019], amplamente empregado em pesquisas sobre inspeção visual automatizada. Dentre as categorias disponíveis, foi selecionado o subconjunto `cable`, que contém imagens de cabos industriais com e sem defeitos.

Esse subconjunto inclui nove classes visuais: `bent_wire`, `cable_swap_combined`, `cut_inner_insulation`, `cut_outer_insulation`, `good`, `missing_cable`, `missing_wire`, `poke_insulation` e `cable_defect_combined`. Essas classes representam diferentes padrões estruturais, abrangendo tanto situações normais quanto diversas formas de falhas mecânicas e elétricas. No total, foram utilizadas 104 imagens organizadas por classe.

Para fins de classificação binária, as nove classes foram agrupadas em duas categorias: cabos bons (classe `good`) e cabos com defeito (todas as demais). Essa simplificação

visa alinhar o problema à realidade industrial, onde o interesse reside, principalmente, em identificar a ocorrência de qualquer tipo de anomalia.

3.2. Etapa 2 – Treinamento da Rede Neural Binarizada

Nesta etapa, foi investigada a aplicação de uma arquitetura de rede neural binária baseada na Wide Residual Network (WRN) para a tarefa de classificação multiclasse de anomalias visuais em cabos industriais. A escolha pela versão binarizada da WRN-20-10 segue a proposta de McDonnell [McDonnell 2018], que demonstra que redes profundas e largas, mesmo sob quantização extrema, podem manter desempenho competitivo.

A seguir, detalha-se a preparação dos dados, as estratégias de aumento de dados, a arquitetura da rede utilizada e o processo de treinamento utilizado.

Preparação dos Dados

Todas as imagens foram redimensionadas para 224×224 pixels e convertidas para o formato RGB. A divisão dos dados seguiu a proporção de 80% para treinamento e 20% para teste, com estratificação das nove classes. Do conjunto de treinamento, 10% foi reservado para validação.

Durante o treinamento, foram aplicadas transformações de aumento de dados utilizando o framework Albumentations: rotações de até 15° , inversão horizontal, desfoque gaussiano, variações de brilho e contraste. Para validação e teste, foram utilizadas apenas redimensionamento e normalização dos valores de pixel.

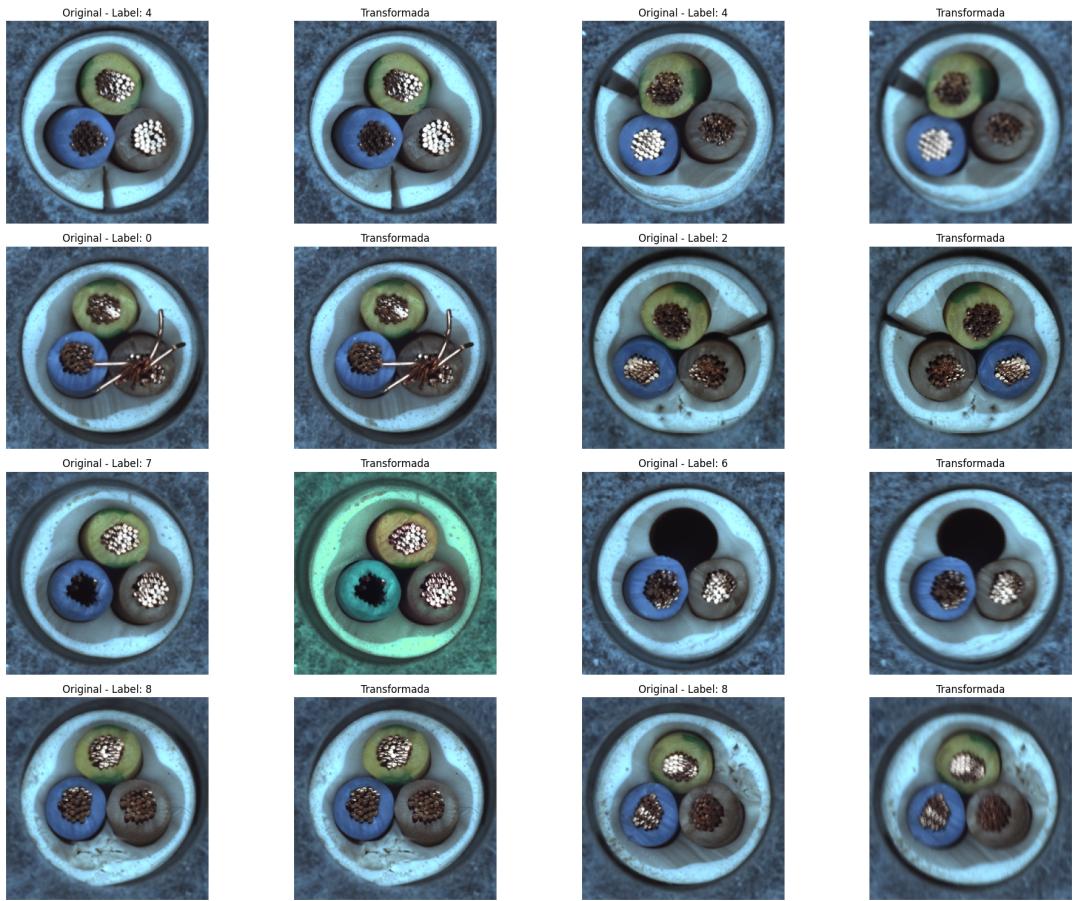


Figura 4. Imagem original e variações geradas por *data augmentation* com Albu-
mentations.

Arquitetura da Rede

Adotou-se a arquitetura Wide ResNet-20-10 binarizada, composta por 20 camadas com largura multiplicada por um fator $k = 10$. Essa arquitetura utiliza pesos e ativações binárias ($\{-1, +1\}$) nas camadas intermediárias, exceto na primeira e última camadas, que permanecem em ponto flutuante.

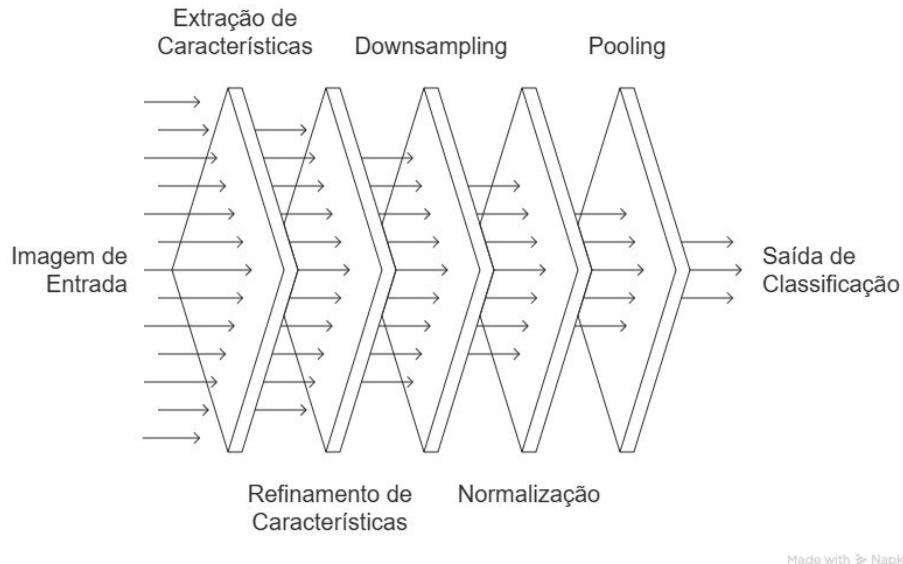


Figura 5. Diagrama esquemático da arquitetura da rede WRN-20-10 binarizada.

Tabela 1. Arquitetura da rede WRN-20-10 binarizada utilizada no experimento

Camada	Tipo	Filtro / Stride	Saída	Binária?
Conv1	Conv2D	3×3 , stride 1	$32 \times 32 \times 16$	Não
Grupo 1	6 blocos residuais	3×3 , stride 1	$32 \times 32 \times 160$	Sim
Grupo 2	6 blocos residuais	3×3 , stride 2	$16 \times 16 \times 320$	Sim
Grupo 3	6 blocos residuais	3×3 , stride 2	$8 \times 8 \times 640$	Sim
GAP	Global Avg Pool	-	$1 \times 1 \times 640$	-
FC	Fully Connected	-	$1 \times 1 \times 9$	Não

Treinamento do Modelo

O modelo foi treinado utilizando o otimizador SGD com momentum 0,9 e weight decay de 1×10^{-4} . A taxa de aprendizado foi fixada em 1×10^{-2} , e a função de perda utilizada foi a cross-entropy.

O treinamento durou 200 épocas, com lote de 32 amostras por iteração. O modelo com menor perda de validação foi salvo e posteriormente avaliado no conjunto de teste.



Figura 6. Exemplo de batch de imagens do conjunto de treino.

4. Avaliação de Desempenho

A Figura 7 apresenta as curvas de perda, acurácia e F1-score durante o treinamento. A acurácia no conjunto de treino superou 90% rapidamente, enquanto a validação apresentou flutuações, indicando o desafio de generalização.

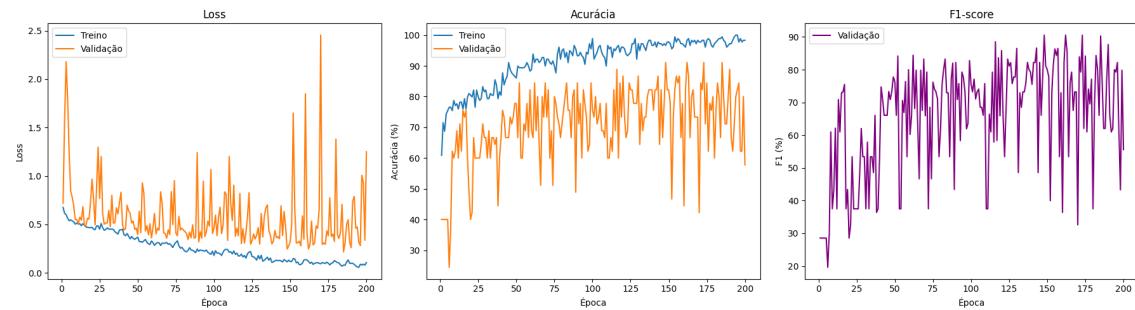


Figura 7. Evolução das métricas durante o treinamento.

Após o agrupamento binário das classes, o modelo atingiu 91,11% de acurácia e F1-score macro de 90,55%, conforme Tabela 2.

Tabela 2. Relatório de classificação para o conjunto de teste - Classificação Binária

Classe	Precisão	Revocação	F1-score	Suporte
0 (defeituoso)	0.9375	0.8333	0.8824	18
1 (bom)	0.8966	0.9630	0.9286	27
Acurácia total			91,11%	
F1-score (macro)			90,55%	

4.1. Discussão dos Resultados

Os resultados obtidos com a rede WRN-20-10 binarizada indicam que é possível atingir desempenho satisfatório mesmo sob restrições extremas de quantização. A acurácia global de 91,11% e o F1-score macro de 90,55% obtidos na classificação binária do conjunto de teste evidenciam que a arquitetura utilizada conseguiu generalizar adequadamente para novos dados, mesmo diante da diversidade visual presente nas amostras.

A partir do relatório de classificação, observou-se que a classe representando cabos com defeito apresentou maior variação na taxa de revocação (83,33%) em comparação

à classe de cabos bons (96,30%), sugerindo uma leve tendência do modelo a privilegiar predições da classe majoritária. Esse comportamento pode ser explicado, em parte, pelo desequilíbrio leve nas quantidades de amostras entre as categorias e pela similaridade estrutural entre alguns padrões de defeito.

A oscilação das métricas de validação ao longo das épocas — especialmente a partir da época 50 — indica que o modelo apresenta certa instabilidade no aprendizado, provavelmente associada à complexidade intrínseca do conjunto de dados. Ainda assim, o uso de técnicas de aumento de dados contribuiu para reduzir o risco de sobreajuste, possibilitando uma convergência mais robusta.

Ao comparar com os descritores clássicos HOG e LBP, visualizados via técnicas de redução de dimensionalidade (PCA, UMAP e t-SNE), observa-se que os embeddings gerados por tais abordagens apresentam baixa capacidade de separação entre as classes, reforçando a limitação de descritores manuais para o problema em questão. Por outro lado, a utilização de uma rede convolucional binarizada foi capaz de extrair representações mais discriminativas, mesmo com a expressividade reduzida pela binarização.

Por fim, o desempenho alcançado sugere que a WRN-20-10 binarizada representa uma alternativa viável para aplicações industriais em dispositivos com recursos computacionais limitados, conciliando boa capacidade preditiva com baixos custos de processamento e memória. No entanto, trabalhos futuros podem investigar abordagens híbridas, técnicas de regularização mais agressivas ou mesmo ajustes finos na arquitetura para melhorar ainda mais a robustez do modelo frente à variabilidade das anomalias visuais.

5. Custo Computacional e Eficiência da Rede

A avaliação do custo computacional da arquitetura WRN-20-10 binarizada é fundamental para validar sua aplicabilidade em cenários com restrições de hardware, como dispositivos de borda e sistemas embarcados. Nesta seção, são apresentados os resultados obtidos durante o processo de treinamento em relação ao tempo de execução e ao consumo de memória, considerando um ambiente típico de experimentação acessível.

5.1. Ambiente de Execução

O treinamento foi realizado na plataforma Google Colab, utilizando uma instância com GPU NVIDIA Tesla T4, 15 GB de memória RAM e sistema operacional Ubuntu 18.04. O backend emprega CUDA 11.2 e cuDNN 8, permitindo a execução eficiente de redes convolucionais otimizadas para aceleração em GPU. O treinamento foi conduzido em ambiente Python 3.10 com suporte a PyTorch 2.0.1.

5.2. Tempo de Treinamento

O tempo de treinamento foi monitorado ao longo das 200 épocas. A duração média por época foi de 8,30 segundos, com desvio padrão de 0,58 segundos, totalizando aproximadamente 1660 segundos (ou 27 minutos e 40 segundos) para o treinamento completo do modelo. Este resultado demonstra a viabilidade da arquitetura em ambientes com recursos computacionais modestos, sem comprometer significativamente o tempo de convergência.

5.3. Consumo de Memória

O consumo de memória durante o treinamento apresentou estabilidade. A memória média utilizada pela GPU foi de 2652,80 MB, enquanto a memória RAM manteve-se em torno

de 1954,36 MB. Estes valores refletem a leveza computacional característica das redes binarizadas, que operam com pesos e ativações representados por um único bit, exceto nas camadas de entrada e saída, que permanecem em ponto flutuante.

5.4. Resumo das Métricas Computacionais

Tabela 3. Resumo do custo computacional durante o treinamento da WRN-20-10 binarizada

Métrica	Valor	Descrição
Tempo médio por época	8,30 s	Tempo médio por ciclo de treino
Desvio padrão do tempo	0,58 s	Variabilidade entre épocas
Tempo total de treinamento	1660 s (27 min 40 s)	Duração total em 200 épocas
Memória média (GPU)	2652,80 MB	Ocupação estável da GPU (NVIDIA T4)
Memória média (RAM)	1954,36 MB	Uso da memória principal da instância

Os dados apresentados indicam que a WRN-20-10 binarizada é adequada para aplicações em ambientes restritos, combinando boa capacidade de generalização com eficiência em termos de tempo de treinamento e consumo de memória. Esses fatores tornam o modelo promissor para aplicações industriais que exigem soluções compactas e de resposta rápida.

5.5. Discussão e Próximos Passos

Os resultados obtidos com a rede binarizada WRN-20-10 demonstram que, mesmo sob fortes restrições de representação numérica, é possível alcançar desempenho competitivo na detecção de anomalias visuais em imagens industriais. A acurácia de 91,11% e o F1-score macro de 90,55% indicam que o modelo foi capaz de capturar padrões discriminativos relevantes entre cabos em condição normal e cabos com defeito. Esse resultado é particularmente relevante quando se considera a leveza computacional da arquitetura, o que reforça seu potencial de aplicação em dispositivos de borda, como microcontroladores e sistemas embarcados com recursos limitados.

Apesar dos resultados satisfatórios, algumas limitações importantes foram observadas. O modelo apresentou dificuldades em identificar corretamente todos os tipos de falhas, como evidenciado pela revocação mais baixa na classe defeituosa. Esse comportamento pode estar relacionado à alta similaridade visual entre algumas subclasses de anomalias, bem como ao desequilíbrio na distribuição das amostras entre as classes. Além disso, as oscilações nas métricas de validação ao longo das épocas indicam certa instabilidade no processo de aprendizado, possivelmente associada à variabilidade intrínseca das imagens e ao número reduzido de exemplos para algumas categorias.

Diante desses desafios, futuras investigações devem considerar estratégias que promovam maior robustez e capacidade de generalização do modelo. A ampliação e diversificação do conjunto de dados por meio de técnicas de aumento específicas pode contribuir para melhorar o desempenho em classes minoritárias. Da mesma forma, o uso de métodos de regularização, como a suavização de rótulos ou técnicas baseadas em mistura de dados, pode ajudar a estabilizar o treinamento e reduzir o risco de sobreajuste.

Outra possibilidade está na adoção de arquiteturas híbridas, que combinem camadas binarizadas e camadas em ponto flutuante, mantendo a eficiência computacional sem sacrificar completamente a expressividade do modelo.

Por fim, destaca-se a importância de avaliar o modelo em ambientes reais de aplicação, medindo seu tempo de inferência, consumo de memória e desempenho energético em dispositivos de borda. Essa etapa é essencial para validar não apenas a eficácia da abordagem proposta, mas também sua viabilidade operacional no contexto da Indústria 4.0. Assim, os resultados obtidos neste trabalho servem como ponto de partida promissor para o desenvolvimento de soluções leves, eficientes e aplicáveis a sistemas de inspeção visual automatizada em ambientes industriais.

Referências

- Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. (2019). Mvttec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9592–9600.
- Cai, L. and Ying, Z. (2009). Face recognition with locally linear embedding on local binary patterns. In *International Conference on Information Science and Engineering (ICISE)*, pages 273–276.
- Cheng, J., Chen, W., Tao, F., and Lin, C.-L. (2018). Industrial iot in 5g environment towards smart manufacturing. *Journal of Industrial Information Integration*, 10:10–19.
- Courbariaux, M., Bengio, Y., and David, J.-P. (2015). Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131.
- Jr, D. P. and da Silva, E. J. (2022). Análise de algoritmos de reconhecimento facial: Hog e yolo. In *19th International Conference on Information Systems and Technology Management*.
- McDonnell, M. D. (2018). Training wide residual networks for deployment using a single bit for each weight. In *International Conference on Learning Representations (ICLR)*.
- Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.
- Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer.