

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**

¹Curso 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 66,67% e F1-score de 64,44%. 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. Contextualização

A Indústria 4.0 tem impulsionado a adoção de tecnologias como Internet das Coisas (IoT) e Inteligência Artificial (IA) para otimizar processos e melhorar a qualidade dos produtos [Cheng et al. 2018]. Nesse cenário, a detecção de anomalias em imagens industriais desempenha um papel crucial, permitindo a identificação precoce de defeitos em produtos e reduzindo custos operacionais [Bergmann et al. 2019]. No entanto, a implementação de modelos tradicionais de deep learning em dispositivos com recursos limitados, como microcontroladores ou GPUs de baixo poder, ainda é um desafio significativo [Wang et al. 2020].

Redes neurais binárias (BNNs) surgem como solução promissora para esse problema, oferecendo eficiência computacional e leveza ao representar pesos e ativações com valores binários (+1 ou -1) [Courbariaux et al. 2015]. Essa abordagem reduz drasticamente o uso de memória e o custo computacional, tornando-a ideal para aplicações em dispositivos de borda [Rastegari et al. 2016]. No entanto, a binarização pode levar à perda de precisão, especialmente em tarefas complexas como a detecção de anomalias [Qin et al. 2020].

Este trabalho segue a abordagem utilizada por McDonnell [McDonnell 2018] para treinar uma Wide ResNet-20-10 binarizada, extrair embeddings e, em seguida, utilizando a rede completa para classificação de anomalias em cabos industriais.

1.1. 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. 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.

2.1. 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].

2.2. 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].

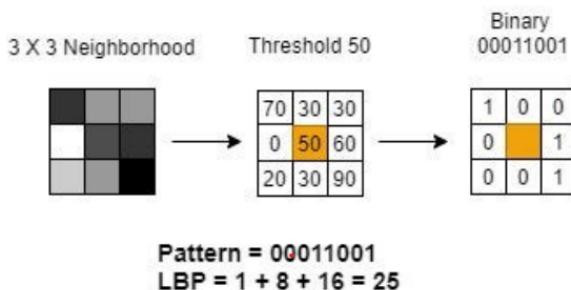


Figura 1. Representação de uma etapa para extração do valor do descritor LBP.

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].

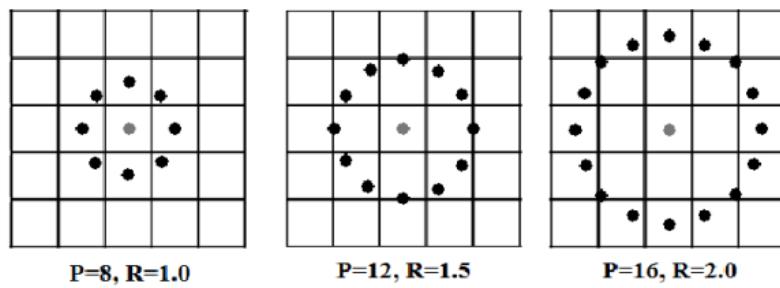


Figura 2. Três operadores LBP circulares simétricos com diferentes raios.

2.3. 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.

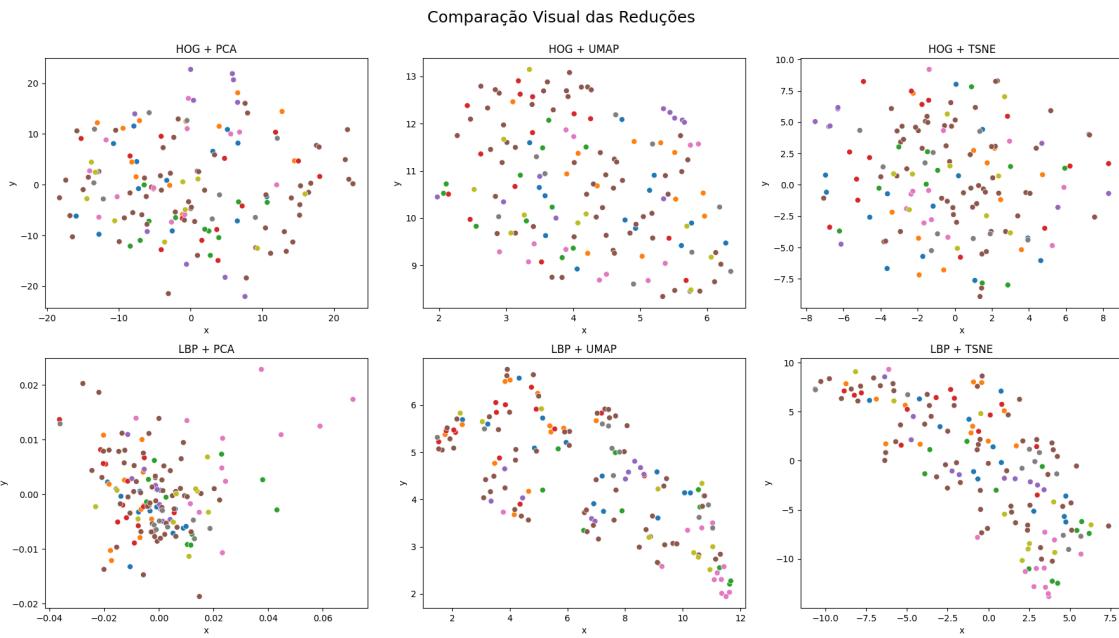


Figura 3. Projeção dos embeddings HOG e LBP em duas dimensões usando PCA, UMAP e t-SNE.

3. Metodologia

3.1. Etapa 1 – Descrição do Conjunto de Dados

As imagens utilizadas neste trabalho 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, são utilizadas 104 imagens, organizadas por classe.

3.2. Etapa 2 – Treinamento da Rede Neural Binarizada

Nesta etapa, investigamos a aplicação de uma arquitetura de rede neural binária (BNN) baseada na Wide Residual Network (WRN) para a tarefa de classificação multiclasse de anomalias visuais em cabos industriais. A escolha por uma versão binarizada da WRN-20-10 fundamenta-se na proposta de McDonnell [McDonnell 2018], que demonstra que redes profundas e largas, mesmo sob restrições de quantização extrema, podem preservar boa capacidade de generalização.

O problema em questão envolve a distinção entre nove classes visuais distintas, conforme definidas no subconjunto `cable` da base MVTec. Essas classes incluem variações sutis de falhas estruturais, como cortes parciais em isolamentos, fios ausentes ou

deformações mecânicas. Tal complexidade torna a tarefa desafiadora, especialmente ao se utilizar redes quantizadas, que reduzem significativamente a expressividade do modelo em troca de eficiência computacional.

Nos parágrafos a seguir, detalhamos a preparação dos dados, as técnicas de aumento de dados utilizadas, a arquitetura adotada, o processo de treinamento e as estratégias de avaliação de desempenho.

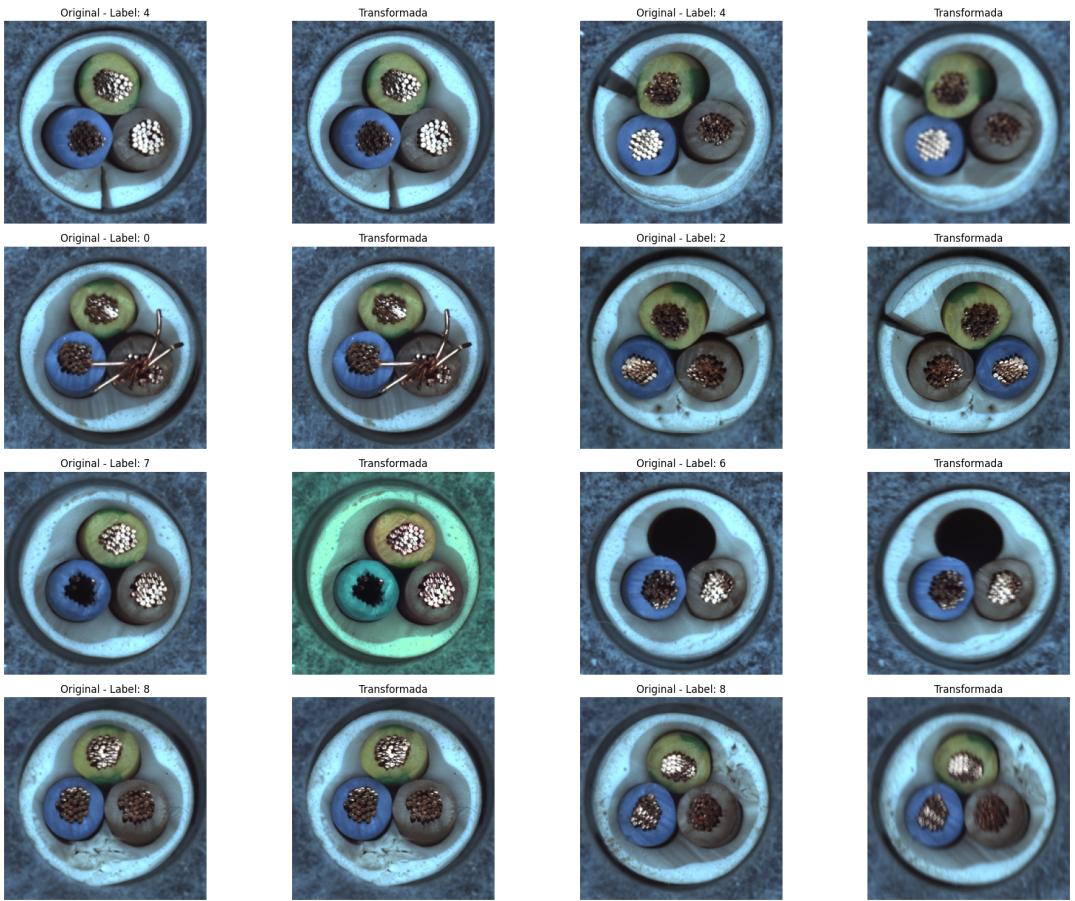
Preparação dos Dados

Antes do treinamento da rede neural binarizada, foi necessário padronizar e estruturar os dados de forma adequada ao pipeline de aprendizado profundo. Todas as imagens foram redimensionadas para 224×224 pixels e convertidas para o formato RGB, garantindo compatibilidade com as dimensões esperadas pela arquitetura WRN-20-10.

A divisão dos dados foi realizada com 80% das amostras destinadas ao treinamento e 20% ao teste, mantendo-se a proporção entre as nove classes por meio de estratificação. Do conjunto de treinamento, 10% foi separado aleatoriamente para compor um subconjunto de validação, utilizado para monitorar o desempenho do modelo ao longo das épocas.

Com o objetivo de aumentar a robustez do modelo e mitigar o sobreajuste, foi aplicado um conjunto de transformações durante o treinamento. Utilizou-se o framework Albumentations para aplicar, de forma aleatória, rotações de até 15° , inversão horizontal, desfoque gaussiano, variações de brilho e contraste. Para os conjuntos de validação e teste, foram aplicadas apenas redimensionamento e normalização dos valores de pixel, assegurando consistência durante a inferência.

A Figura 4 ilustra o efeito dessas transformações sobre uma imagem de cabo industrial, comparando a versão original com diferentes amostras geradas via *data augmentation*. Essa visualização evidencia a variabilidade introduzida no conjunto de treinamento, fundamental para melhorar a generalização do modelo.



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

Arquitetura da Rede

A arquitetura adotada neste trabalho é a Wide ResNet-20-10 binarizada, proposta por McDonnell [McDonnell 2018], uma variação da tradicional ResNet com profundidade moderada (20 camadas) e largura expandida por um fator $k = 10$. Essa arquitetura equilibra profundidade e expressividade, sendo adequada para cenários com restrições de memória e processamento.

A estrutura da rede é composta por um bloco inicial de convolução, seguido por três grupos de blocos residuais idênticos, responsáveis por processar representações intermediárias com resoluções decrescentes. As ativações intermediárias são processadas por operações binarizadas, utilizando pesos e ativações com valores em $\{-1, +1\}$, com exceção da primeira e da última camada, que permanecem em ponto flutuante.

Ao final da rede, aplica-se uma operação de *global average pooling*, seguida por uma camada totalmente conectada responsável pela previsão da classe.

A Tabela 1 resume a composição da arquitetura utilizada:

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 treinamento da rede foi conduzido utilizando o otimizador Stochastic Gradient Descent (SGD), com *momentum* de 0,9 e *weight decay* de 1×10^{-4} , combinando simplicidade com estabilidade na atualização dos pesos. A função de perda utilizada foi a *cross-entropy*, adequada para classificação multiclasse com nove categorias.

A taxa de aprendizado foi mantida constante em 1×10^{-2} ao longo das 200 épocas de treinamento, sem uso de agendador de ajuste dinâmico. Essa decisão visa facilitar a análise do comportamento do modelo sob um regime de otimização estável e previsível.

O treinamento foi realizado com tamanho de lote igual a 32. Em cada época, o modelo era avaliado no conjunto de validação, e o estado com menor perda validada era salvo automaticamente para posterior avaliação no conjunto de teste.

A Figura 5 apresenta um exemplo visual de um batch de imagens do conjunto de treino. Observa-se a diversidade de padrões visuais presentes nas amostras, o que reforça o desafio da tarefa de classificação.



Figura 5. Exemplo de batch de imagens do conjunto de treino. As classes representam diferentes padrões de anomalias ou condição normal de cabos industriais.

Avaliação de Desempenho

O desempenho do modelo foi avaliado no conjunto de teste utilizando o modelo salvo na época 99, correspondente ao ponto de menor perda de validação durante o treinamento. Esse procedimento de *early checkpointing* permite selecionar a configuração de pesos mais generalizável, evitando o risco de sobreajuste.

A Figura 6 apresenta a evolução das métricas de perda, acurácia e F1-score ao longo das 200 épocas. Observa-se uma melhora progressiva no conjunto de treino, en-

quanto o conjunto de validação apresenta oscilações significativas, reflexo da complexidade da tarefa e da variabilidade entre as classes.

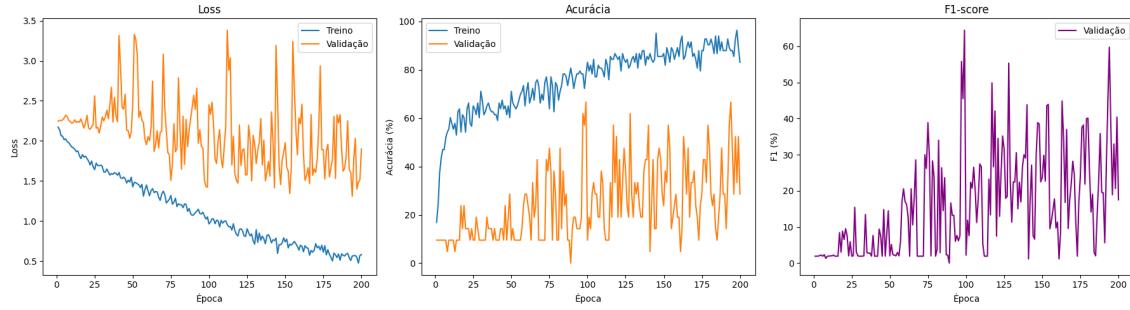


Figura 6. Evolução das métricas durante o treinamento: (esquerda) perda, (centro) acurácia e (direita) F1-score no conjunto de validação.

No conjunto de teste, o modelo alcançou uma acurácia global de **66,67%** e um F1-score macro de **64,44%**. O relatório de classificação completo está apresentado na Tabela 2.

Tabela 2. Relatório de classificação para o conjunto de teste (época 99)

Classe	Precisão	Revocação	F1-score	Suporte
0	0.4286	1.0000	0.6000	3
1	0.6667	0.6667	0.6667	3
2	0.3333	0.5000	0.4000	2
3	1.0000	0.6667	0.8000	3
4	1.0000	0.5000	0.6667	2
5	0.0000	0.0000	0.0000	2
6	1.0000	1.0000	1.0000	2
7	1.0000	0.5000	0.6667	2
8	1.0000	1.0000	1.0000	2
Acurácia total	66,67%			
F1-score (macro)	64,44%			

Embora o modelo tenha apresentado desempenho elevado em algumas classes, como 6, 8 e 3, outras, como a classe 5, não foram corretamente identificadas. Esse resultado sugere a existência de classes mais difíceis de generalizar, possivelmente devido à semelhança visual com outras categorias. Estratégias de regularização poderão ser exploradas nas próximas etapas para mitigar essas limitações.

3.3. Discussão e Próximos Passos

Os resultados obtidos indicam que a arquitetura binarizada WRN-20-10 foi capaz de aprender padrões relevantes mesmo sob forte restrição de representação numérica. Algumas classes com poucos exemplos apresentaram desempenho elevado, com F1-score chegando a 1.0, o que sugere que a rede consegue capturar certas variações estruturais

específicas. Por outro lado, a incapacidade do modelo em classificar corretamente todas as categorias, como evidenciado pelo desempenho nulo na classe 5, revela limitações importantes.

As oscilações observadas nas métricas de validação ao longo das épocas sugerem instabilidade decorrente tanto da variabilidade entre as amostras quanto da baixa cardinalidade de algumas classes. Além disso, por se tratar de uma tarefa de classificação multiclasse sobre imagens visualmente semelhantes, o desafio se intensifica ainda mais quando se adota uma arquitetura binarizada, cujo poder de representação é naturalmente mais limitado em relação a redes tradicionais em ponto flutuante.

Diante desse cenário, considera-se essencial investigar abordagens que ampliem a robustez do modelo. Entre elas, destacam-se o balanceamento das classes por meio de técnicas de aumento de dados direcionadas às categorias minoritárias, a aplicação de regularizações mais agressivas, como dropout e suavização de rótulos, e o uso de pesos pré-treinados em redes híbridas parcialmente binarizadas. Também se planeja realizar uma análise mais aprofundada da eficiência computacional da rede, medindo-se o tempo de inferência, o consumo de memória e o tamanho final do modelo em dispositivos de borda. Essas ações buscam não apenas aprimorar a acurácia do sistema, mas também torná-lo mais viável para aplicações industriais reais.

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.
- Qin, H., Gong, R., Liu, X., Bai, X., Song, J., and Sebe, N. (2020). Binary neural networks: A survey. *Pattern Recognition*, 105:107281.
- 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.
- Wang, Z., Wu, Z., Lu, J., and Zhou, J. (2020). Bidet: An efficient binarized object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2049–2058.