

Classificação Automática de Defeitos em Superfícies

Comparativo: MLP, DNN, CNN e o Impacto de Autoencoders (SAE)

Gabriel Alvares de Sousa Guimarães

CPE727 - Aprendizado Profundo



1. Introdução e Contexto

Contexto Industrial

A Indústria Siderúrgica:

- A qualidade superficial do aço plano define seu valor de mercado.
- Defeitos como rachaduras e incrustações comprometem a segurança estrutural.

O Problema da Inspeção Manual:

- **Subjetividade:** A classificação varia entre inspetores humanos.
- **Fadiga:** Queda drástica de precisão em turnos longos.
- **Gargalo:** Velocidade incompatível com linhas modernas de laminação.



Objetivos do Trabalho

A proposta é substituir a inspeção manual por Visão Computacional.

Perguntas do Trabalho:

1. **Arquitetura:** O ganho de performance das CNNs justifica sua complexidade frente a redes densas (MLP/DNN)?
2. **Pré-treinamento (SAE):** A inicialização com *Sparse Autoencoders* melhora a estabilidade e reduz o overfitting?
3. **Confiabilidade:** O modelo foca no defeito real ou em ruídos de fundo? (Validado via Grad-CAM).

2. Fundamentação Teórica

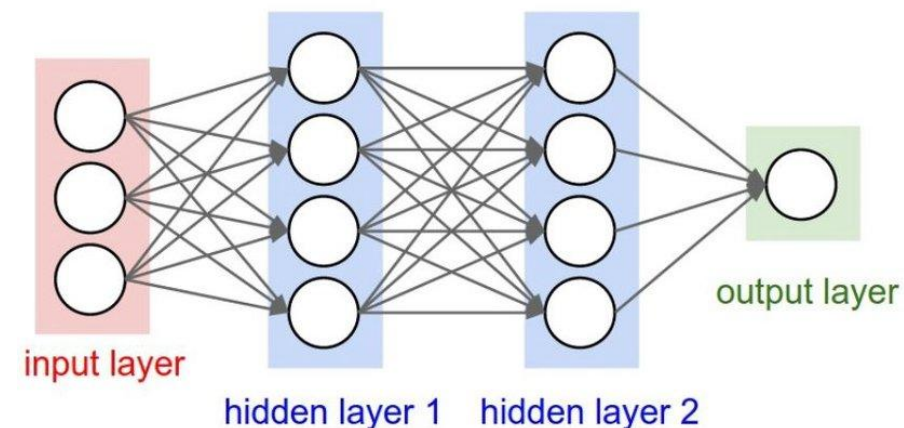
Redes Neurais Densas (MLP/DNN)

Conceito:

- Neurônios totalmente conectados (*Fully Connected*).
- A imagem é transformada em um vetor único ("Flatten").

Limitação Crítica:

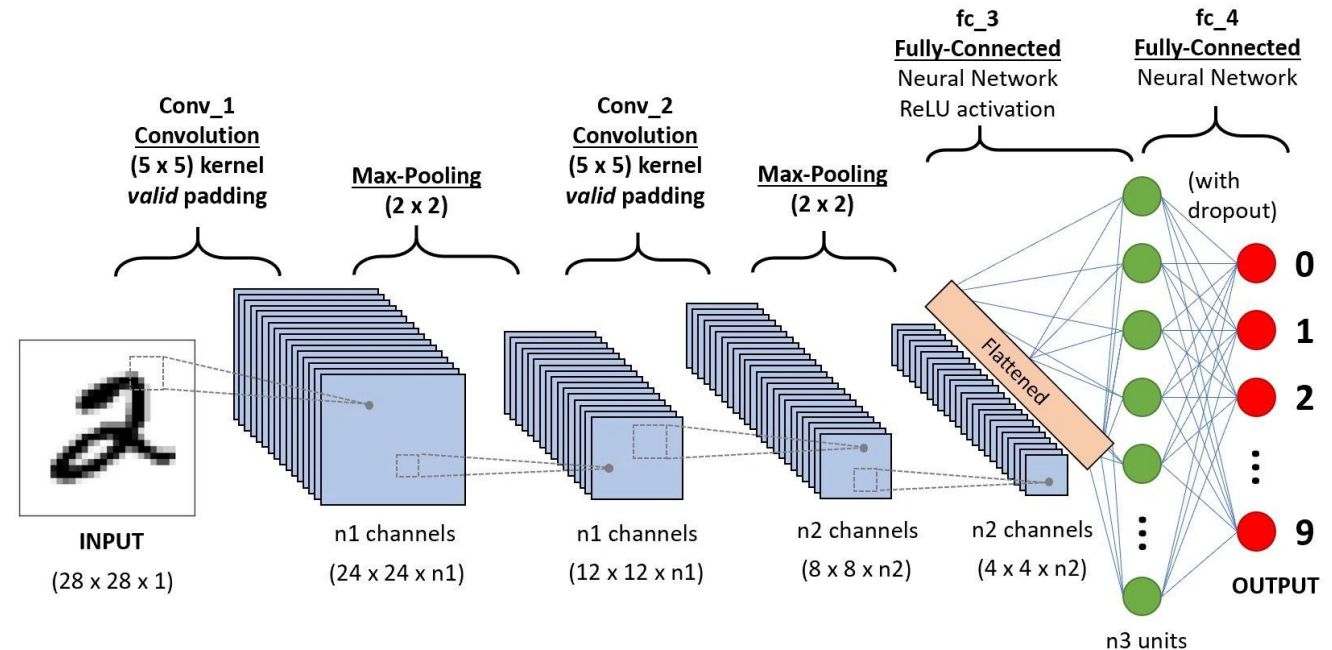
- **Perda da topologia espacial:** A rede não sabe que um pixel é vizinho do outro.
- Dificuldade em detectar padrões de textura invariantes.



Redes Neurais Convolucionais (CNN)

A Solução para Imagens:

- **Convolução:** Filtros deslizantes que aprendem características locais (bordas, texturas).
- **Pooling:** Redução de dimensionalidade mantendo as features mais fortes.
- **Hierarquia:** Camadas iniciais veem linhas; camadas finais veem formas complexas.



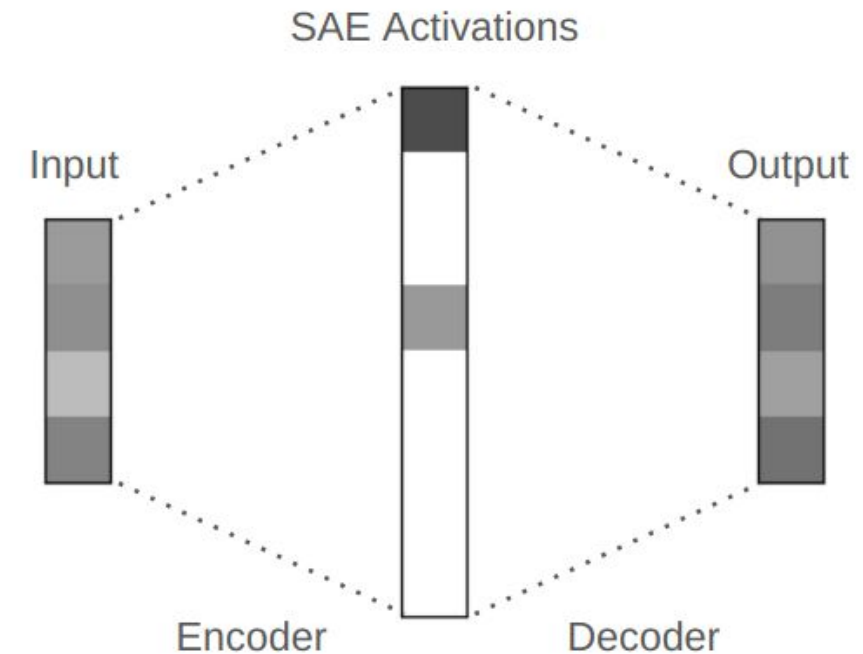
Autoencoders (SAE)

Aprendizado Não Supervisionado:

- O modelo não usa rótulos (labels).
- **Objetivo:** Reconstruir a entrada

Transfer Learning (A Hipótese):

- Treinamos o *Encoder* para comprimir a imagem.
- Usamos esses pesos para inicializar a rede classificadora.
- Espera-se que a rede comece o treino num ponto "mais inteligente".



3. Metodologia

O Dataset NEU-DET

Origem: Northeastern University Surface Defect Database.

Volume: 1.800 imagens.

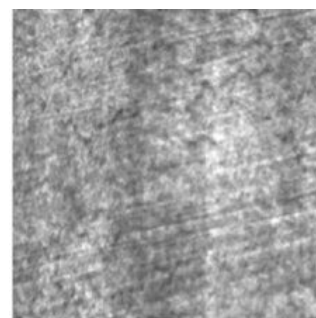
Balanceamento: 300 imagens por classe (Perfeitamente balanceado).

As Classes de Defeitos

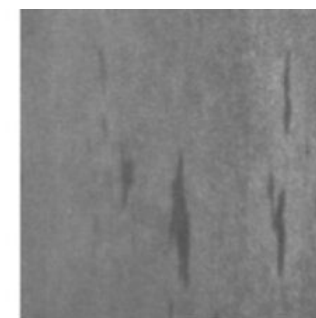
Desafio Visual: Alta similaridade de texturas

Classes:

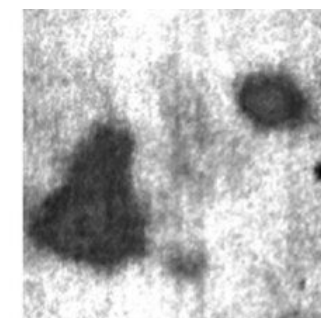
- *Crazing* (Rachaduras difusas).
- *Inclusion* (Partículas escuras).
- *Patches* (Manchas cinzas).
- *Pitted Surface* (Corrosão pontual).
- *Rolled-in Scale* (Óxido laminado).
- *Scratches* (Riscos lineares).



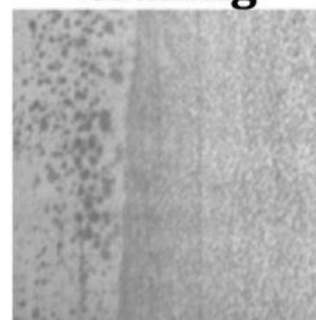
Crazing



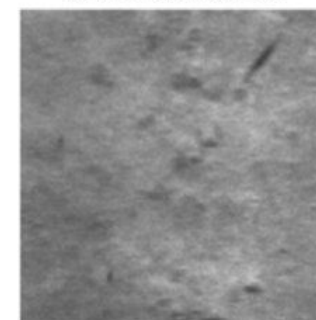
Inclusion



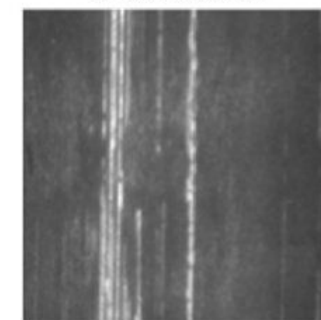
Patches



Pitted surface



Rolled-in scale



Scratches

Pipeline de Pré-processamento

1. Carregamento e Conversão (Grayscale):

- As imagens são carregadas via biblioteca PIL e convertidas para Escala de Cinza (Modo 'L').
- **Motivo:** Defeitos geométricos dependem de textura e contraste, não de cor. Isso reduz a complexidade do modelo (1 canal de cor em vez de 3).

2. Normalização (ToTensor):

- Converte os pixels de inteiros $[0, 255]$ para ponto flutuante **[0.0, 1.0]**.
- **Motivo:** Redes neurais convergem mais rápido e com mais estabilidade quando os dados de entrada estão em uma escala pequena e normalizada.

Estratégia de treinamento

1. Holdout Externo (15% - Teste):

- 270 Imagens Intocadas.
- Separadas *antes* de qualquer treino.
- Usadas apenas para a tabela final.

2. Stratified K-Fold (k=10):

- Aplicado aos 85% restantes, divididos entre treino e validação.
- Gera **Média e Desvio Padrão** das métricas ($\mu \pm \sigma$)

Figuras de Mérito

1. Acurácia:

- Mede a taxa de acertos totais.
- **Justificativa:** É uma métrica válida neste estudo porque o dataset é **perfeitamente balanceado** (300 imagens por classe). Em datasets desbalanceados, ela seria descartada.

2. Recall e Precision :

- **Recall:** A capacidade de encontrar todos os defeitos. (Foco: Evitar enviar peças ruins ao cliente).
- **Precision:** A confiabilidade de que um alerta de defeito é real. (Foco: Não parar a produção por alarmes falsos).
- **Justificativa:** Permitem analisar o *trade-off* entre segurança e eficiência na linha de produção.

Figuras de Mérito

3. F1-Score (Métrica Principal):

- Média harmônica entre Precision e Recall.
- **Justificativa:** Foi a métrica escolhida para o ranking final dos modelos. O F1 pune modelos que "chutam" muito (alto recall, baixa precisão) ou que são conservadores demais, exigindo equilíbrio.

4. Matriz de Confusão:

- **Justificativa:** Ferramenta diagnóstica. Permite visualizar exatamente quais classes são confundidas (ex: Se o modelo troca *Rolled-in Scale* por *Crazing* devido à textura similar).

4. Experimentos

Arquiteturas Implementadas

Arquitetura	Dimensão de Entrada	Configuração de Camadas (Layers)	Regularização e Ativação	Dimensão de Saída / Latente
DNN	Vetor 1D (Flatten) 1 x 40.000	3 Blocos Densos (Fully Connected) + Camada de Classificação	<ul style="list-style-type: none"> • Batch Normalization • Dropout • ReLU 	6 Neurônios (Classes de Defeito)
CNN	Matriz 2D (Grayscale) 200 x 200	3 Blocos Convolucionais + Camadas Densas Finais	<ul style="list-style-type: none"> • MaxPooling (2x2) • ReLU 	6 Neurônios (Classes de Defeito)
SAE	Matriz 2D (Grayscale) 200 x 200	Encoder (Compressão) Decoder (Reconstrução)	<ul style="list-style-type: none"> • Esparsidade (Sparsity) • MSE Loss (Reconstrução) 	Espaço Latente: 32 dim Saída: 200 x 200

Hiperparâmetros

Parâmetro	Configuração / Valor
Hardware	GPU NVIDIA (CUDA)
Otimizador	Adam (Adaptive Moment Estimation)
Learning Rate (LR)	<ul style="list-style-type: none">• SAE: $1e^{-3}$• Classificadores (MLP/DNN/CNN): $1e^{-4}$
Função de Perda (Loss)	<ul style="list-style-type: none">• Classificação: CrossEntropyLoss• Reconstrução (SAE): MSELoss
Critério de Parada	Early Stopping (Paciência = 10 épocas)

Limiares de Decisão

1. A Escolha do Argmax (Abordagem Atual):

- Utilizamos a função Argmax (classe de maior probabilidade) para calcular as métricas.
- **Justificativa:** Garante uma comparação justa e direta entre as arquiteturas (MLP vs. CNN) sem introduzir a variável de ajuste de limiar.

2. Problemas para uso Industrial:

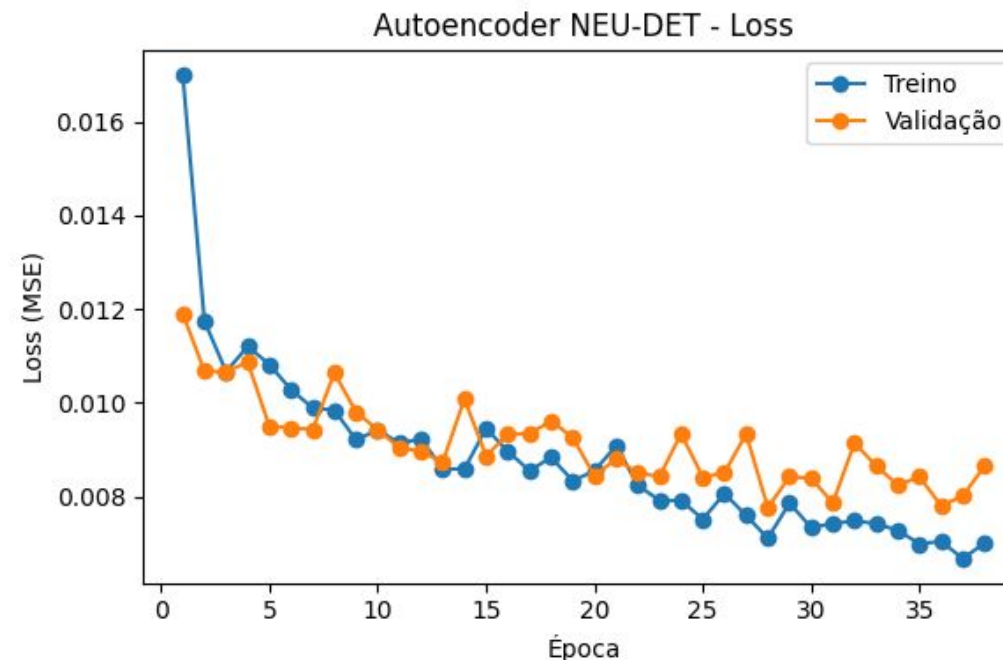
- Embora o Argmax seja ótimo para comparar modelos, ele é "forçado": decide mesmo quando a confiança é baixa (ex: 35%).
- Na indústria, custos são assimétricos: um **Falso Negativo** (enviar peça defeituosa) é muito mais grave que um **Falso Positivo**.

3. Possibilidade de Melhoria (Reject Option):

- Para implantação futura, propõe-se adicionar uma **Zona de Rejeição**.
- **Regra:** Se a confiança for $< 80\%$, o modelo se abstém e encaminha para inspeção humana, aumentando a segurança do sistema.

Treinamento do Autoencoder (SAE)

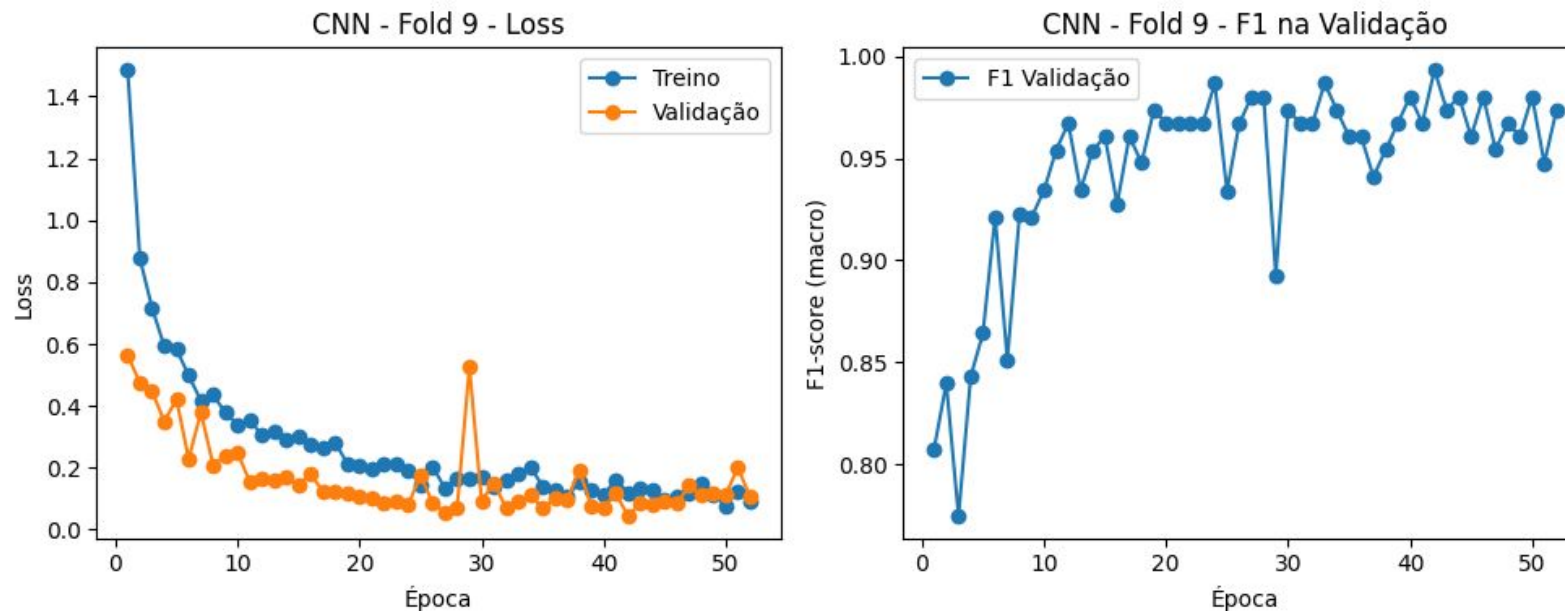
O gráfico de Loss (MSE) demonstra queda acentuada nas primeiras épocas. Convergência rápida e estável.



Interpretação: O Encoder aprendeu a reconstruir a chapa de aço e seus defeitos com sucesso, criando filtros úteis antes mesmo de saber as classes.

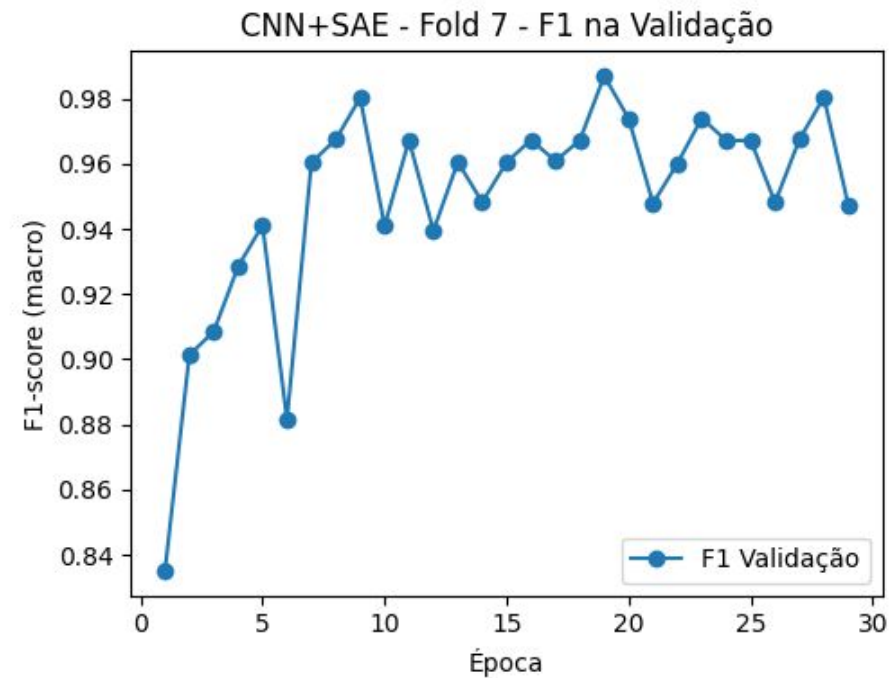
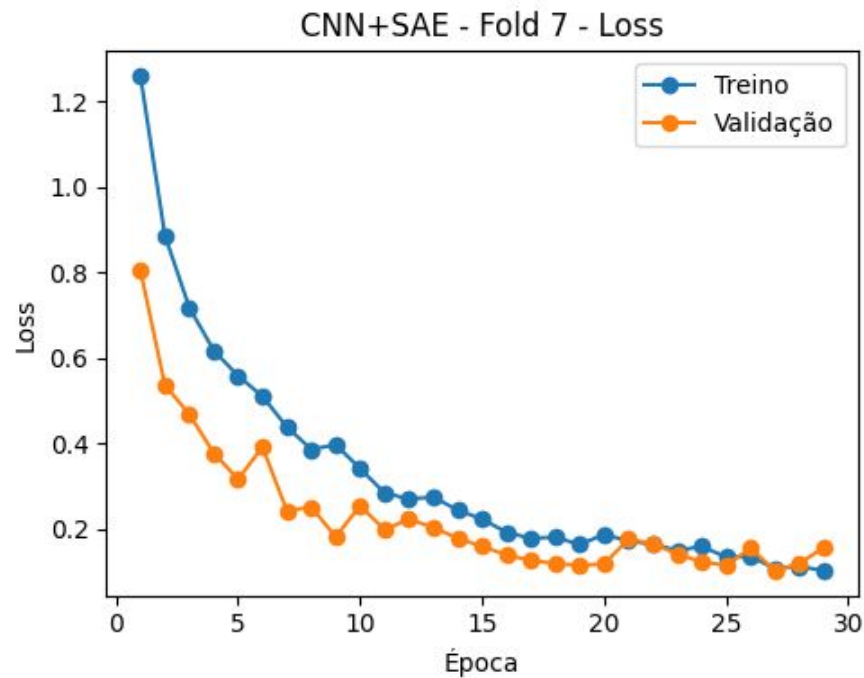
Treinamento

Os gráficos apresentam a Loss e o F1 no melhor fold do treinamento de cada arquitetura.



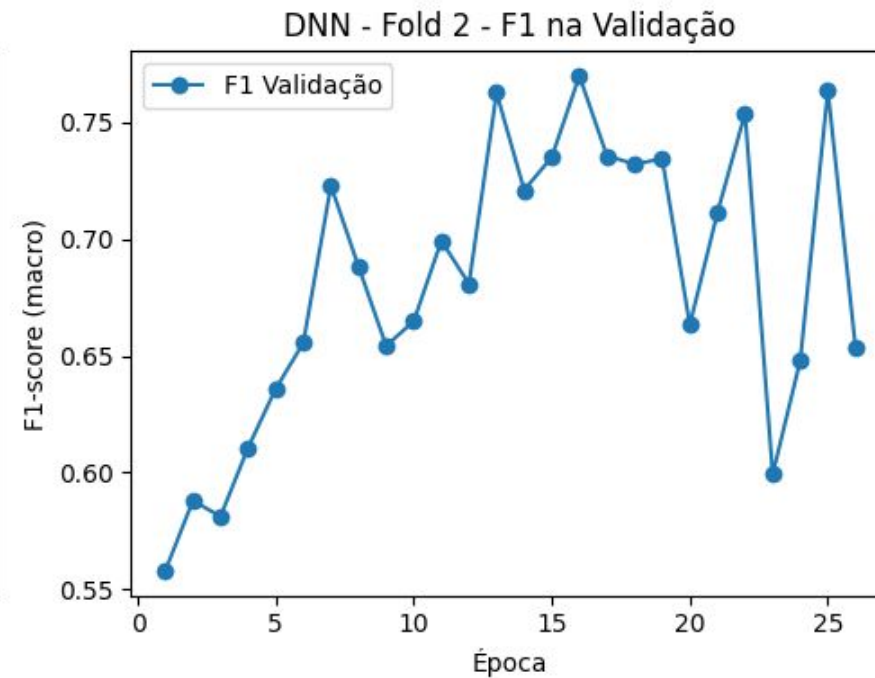
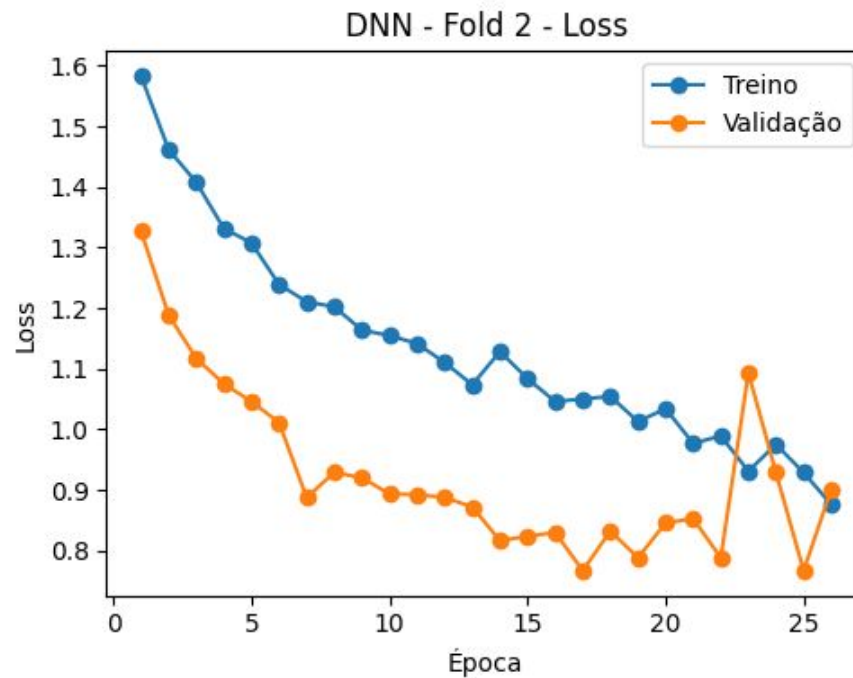
CNN

Treinamento



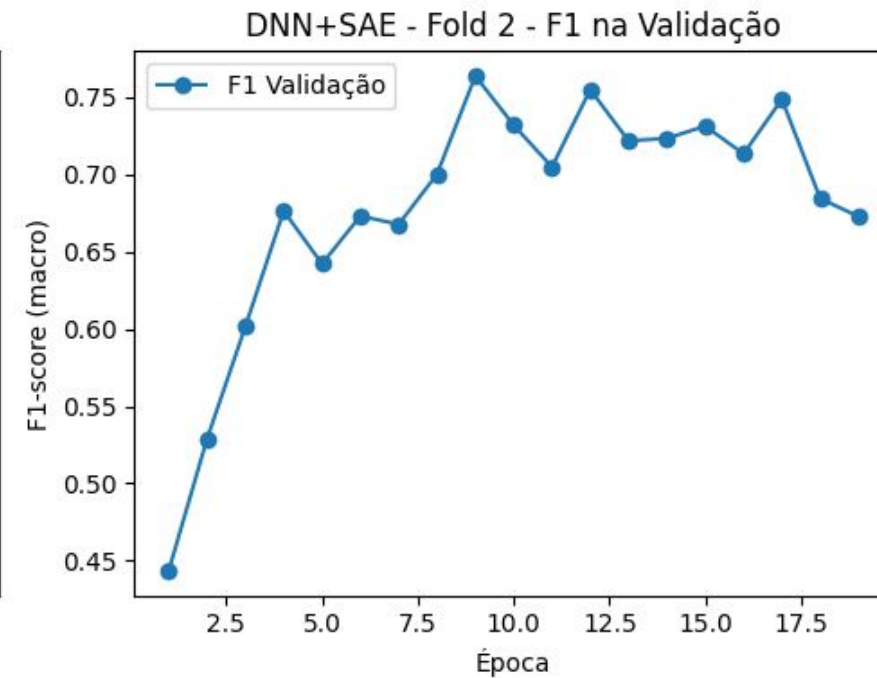
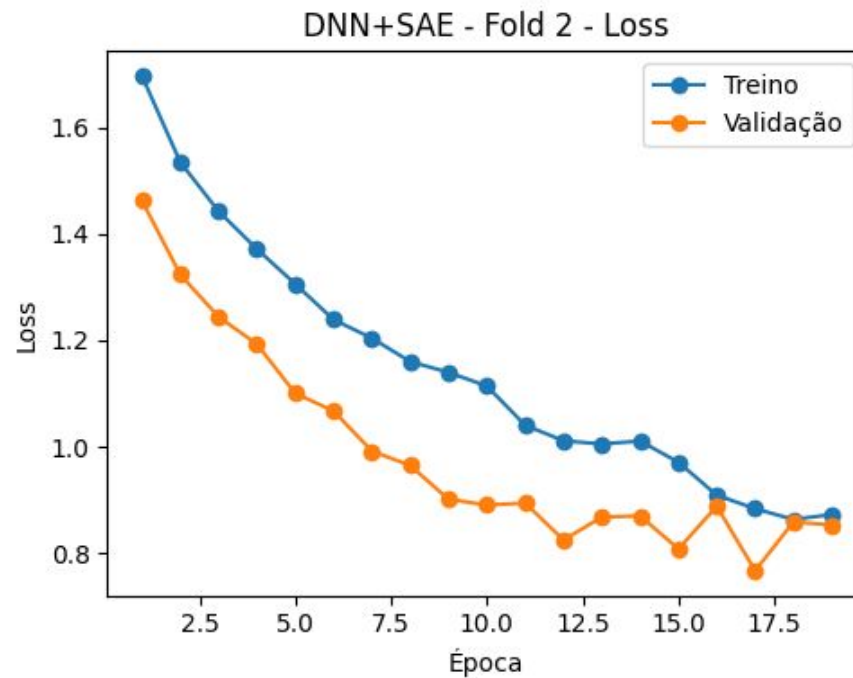
CNN +SAE

Treinamento



DNN

Treinamento



DNN + SAE

5. Resultados e Discussão

Baseline (DNN)

1. O Modelo Base:

- Utilizamos a **Deep Neural Network (DNN)** pura como baseline para avaliar o potencial de redes profundas *sem* convolução.
- **Resultado (K-Fold):** F1-Score médio de **70.61% \pm 2.25**.

2. Análise de Desempenho:

- O modelo utiliza técnicas modernas (Batch Normalization, Dropout) para evitar a instabilidade de redes rasas.
- **Limitação:** Ao usar a camada *Flatten* na entrada, a DNN "destrói" a estrutura 2D da imagem, tratando pixels vizinhos como dados independentes.

3. Conclusão do Baseline:

- A DNN consegue detectar defeitos óbvios (padrões globais de pixel), mas falha em diferenciar texturas complexas.
- Este resultado (~70%) é o piso que as arquiteturas especializadas (CNN) e as técnicas de pré-treinamento (SAE) devem superar.

Impacto do SAE nas Redes Densas (DNN)

1. Comparativo de Desempenho (K-Fold):

- **DNN Pura:** F1-Score médio de **70.61%** \pm 2.25.
- **DNN + SAE:** F1-Score médio de **72.71%** \pm 2.01.

2. Análise do Ganho:

- Houve uma melhoria média de **~2.1%** na Acurácia e no F1-Score.
- Isso confirma que, para redes densas (que ignoram a estrutura espacial), a inicialização via Autoencoder ajuda a extrair características mais robustas desde o início.

3. Conclusão Parcial:

- O pré-treinamento não supervisionado foi benéfico, atuando como um auxílio para a convergência em arquiteturas limitadas (não-convolucionais).

CNN vs. CNN+SAE

1. Resultados Elevados:

- Ambas as redes atingiram o patamar de **97%**, confirmando a superioridade da abordagem convolucional sobre a densa (~72%).

2. Comparação Direta:

- **CNN Pura:** F1-Score de **97.39%** ± 1.34 .
- **CNN + SAE:** F1-Score de **97.07%** ± 1.39 .

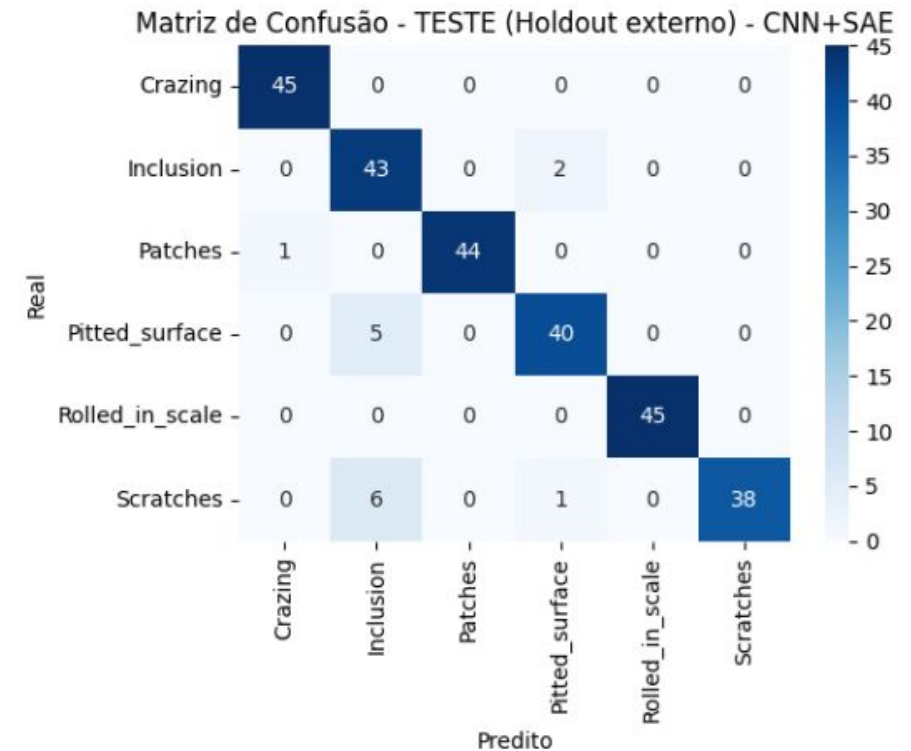
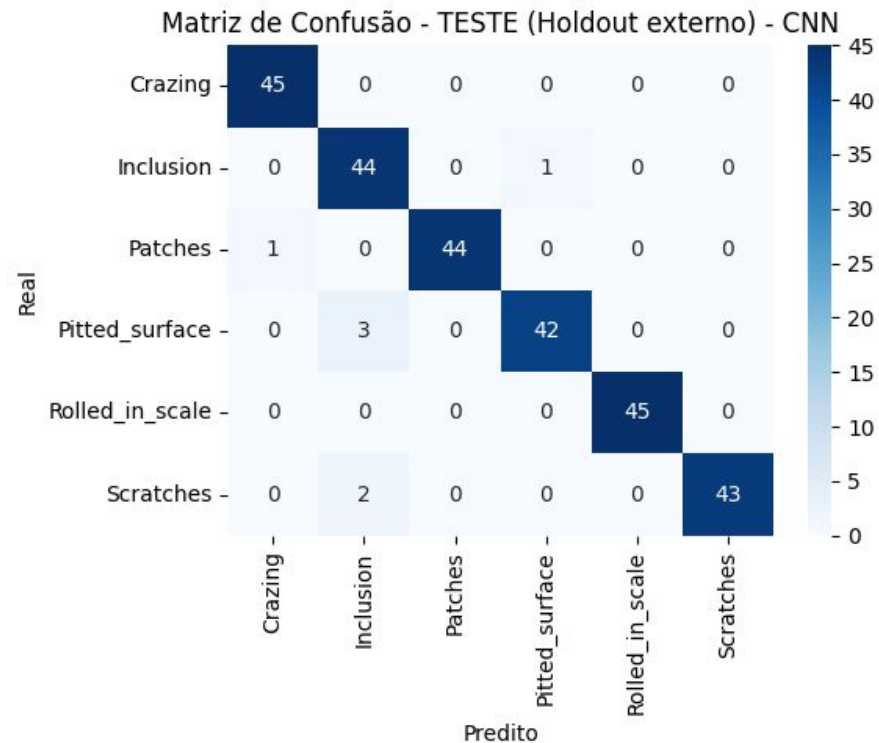
3. Conclusão:

- Estatisticamente, os modelos estão **empatados** (os intervalos de confiança se sobrepõem).
- **Interpretação:** A CNN pura é capaz de aprender filtros ótimos "do zero" para este dataset. O custo computacional extra do SAE não se justificou neste cenário específico.

Análise da Performance

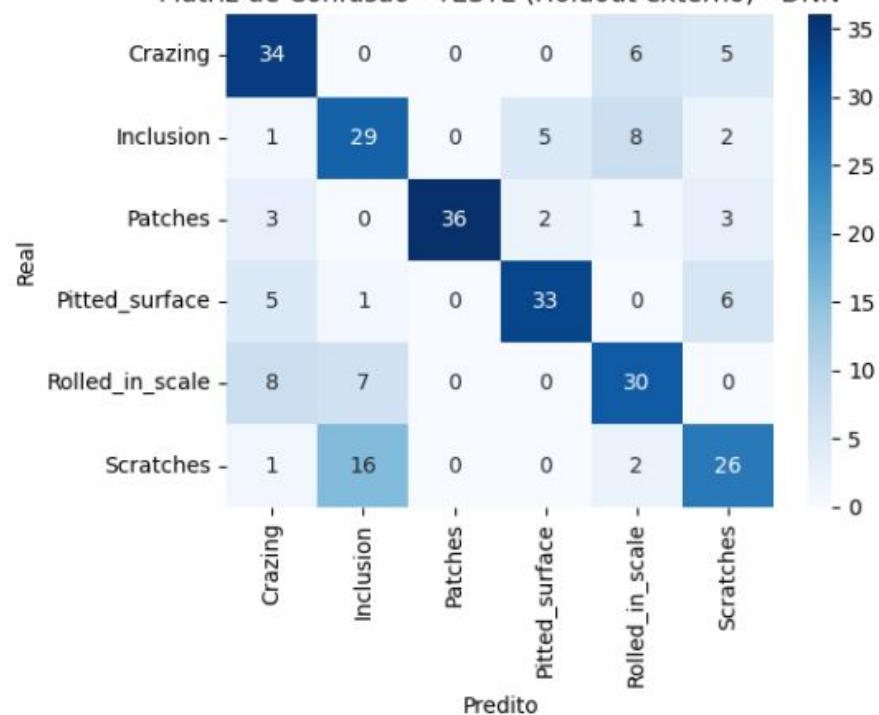
Modelo	Acurácia (%)	Precision	Recall	F1-Score
DNN	70.46 ± 2.44	0.735 ± 0.03	0.704 ± 0.02	0.706 ± 0.02
DNN+SAE	72.88 ± 2.96	0.750 ± 0.03	0.729 ± 0.03	0.727 ± 0.03
CNN	97.39 ± 1.34	0.975 ± 0.01	0.974 ± 0.01	0.974 ± 0.01
CNN+SAE	97.06 ± 1.39	0.973 ± 0.01	0.971 ± 0.01	0.971 ± 0.01

Matrizes de Confusão

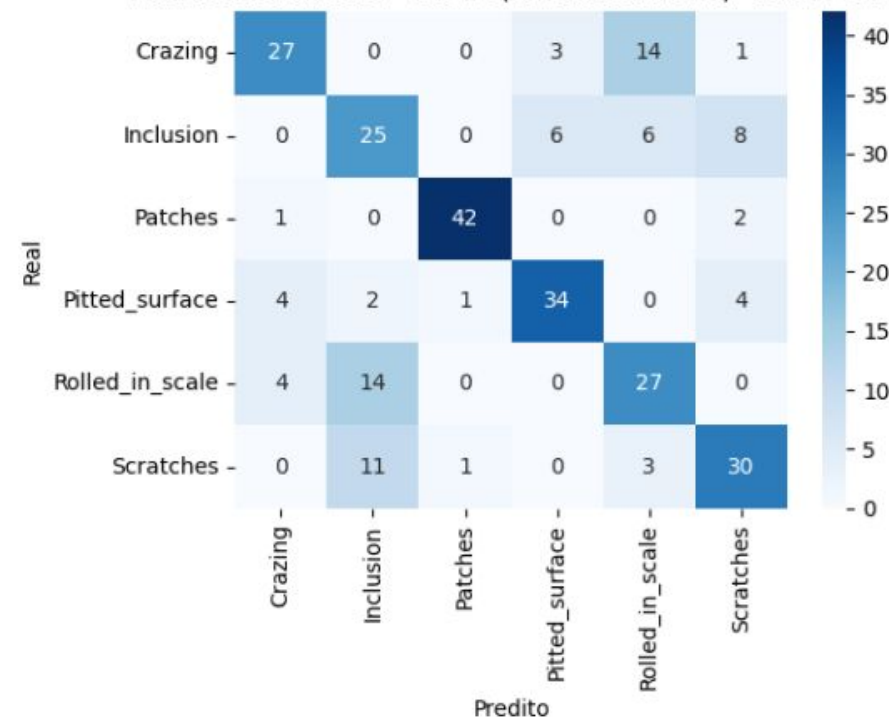


Matrizes de Confusão

Matriz de Confusão - TESTE (Holdout externo) - DNN



Matriz de Confusão - TESTE (Holdout externo) - DNN+SAE



Análise da Performance

1. Vantagem da convolução:

- A arquitetura CNN superou a DNN em mais de **25 pontos percentuais** (97% vs 72%). Isso prova definitivamente que a topologia espacial é o fator mais crítico para classificar defeitos no aço.

2. O Papel do SAE:

- **Positivo na DNN:** Melhorou a acurácia em +2.4%, validando seu uso em arquiteturas com dificuldade de extração de features.
- **Neutro na CNN:** Não superou o treinamento supervisionado puro. Isso indica que, com dados suficientes e balanceados, a CNN aprende filtros eficientes sem necessidade de pré-treino.

3. Recomendação Final:

- Para implantação industrial no NEU-DET, a **CNN Pura** é a escolha ideal: oferece a máxima performance (97.4%) com a menor complexidade de treinamento.

Explicabilidade com Grad-CAM

1. O que é (Grad-CAM):

- *Gradient-weighted Class Activation Mapping*: Técnica que usa os gradientes da última camada convolucional para gerar um mapa de calor das regiões que ativaram a decisão.

2. Objetivo Crítico:

- Validar se o modelo aprendeu o padrão do defeito real ou se está "trapaceando" usando vieses (ex: bordas escuras, marcas d'água ou iluminação de fundo).

3. Estratégia de Análise:

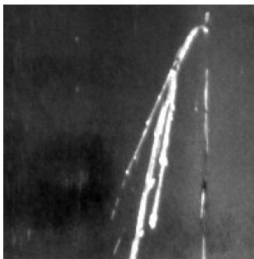
- A validação visual foi focada na **CNN Pura** (Melhor Modelo).

Análise Visual e Validação dos Resultados

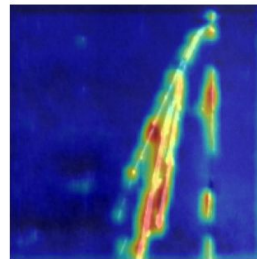
1. O Teste Visual:

O objetivo é ver se as áreas "quentes" (vermelhas) coincidem com a textura do defeito.

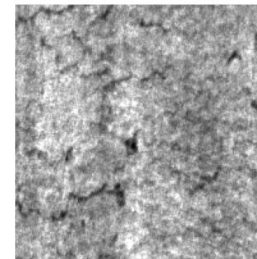
Original (CORRETA)
True: Scratches
Pred: Scratches



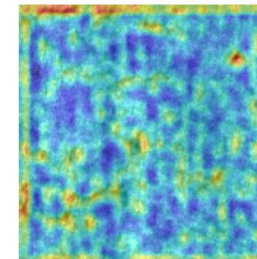
Class Activation Map



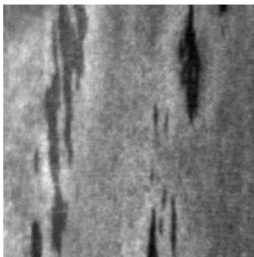
Original (CORRETA)
True: Crazing
Pred: Crazing



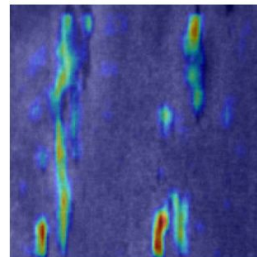
Class Activation Map



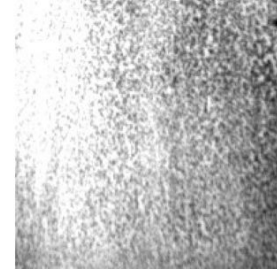
Original (CORRETA)
True: Inclusion
Pred: Inclusion



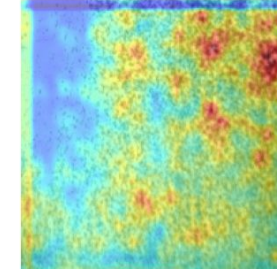
Class Activation Map



Original (CORRETA)
True: Pitted_surface
Pred: Pitted_surface



Class Activation Map



Trabalhos Futuros

Deep Transfer Learning: Testar ResNet50 e EfficientNet.

Detecção de Objetos: Evoluir para YOLO (localizar onde está o defeito, não apenas classificar a imagem inteira).

Edge Computing: Validar inferência em hardware embarcado para chão de fábrica.

Perguntas