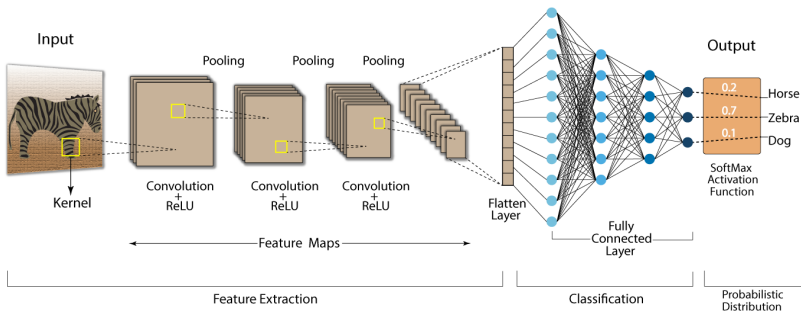


Redes Neurais Convolucionais (CNNs)

Cesar de Oliveira F. Silva

Setembro / 2026



Ideia central

CNNs aprendem automaticamente padrões espaciais em imagens por meio de **convolução, ativação não linear, subamostragem** (pooling) e, ao final, camadas de decisão para classificação ou segmentação.

O que é uma CNN?

- Uma **Convolutional Neural Network** é uma rede neural profunda voltada para dados com estrutura espacial, como imagens e rasters.
- Sua arquitetura combina:
 - **camadas convolucionais** para extração de padrões locais;
 - **funções de ativação** para introduzir não linearidade;
 - **pooling** ou **stride** para redução de resolução;
 - **camadas finais** para classificação, regressão ou segmentação.
- Em vez de conectar todos os pixels a todos os neurônios, a CNN explora:
 - **conectividade local**;
 - **compartilhamento de parâmetros**;
 - **aprendizado hierárquico de atributos**.

Arquiteturas profundas para dados ambientais espaciais

- Dados raster e espaciais: multiespectral, hiperespectral, SAR, DEM e séries temporais.
- Por que usar redes profundas?
 - capturam padrões espaciais em múltiplas escalas;
 - exploram contexto local e regional;
 - aprendem atributos discriminativos diretamente dos dados.
- Pipeline típico:

pré-processamento → patches / tiles → CNN → pós-processamento

Convolução: conceito e notação

Definição

Convolução combina a entrada com um filtro (kernel) para produzir um **mapa de atributos** (*feature map*).

$$Y[i, j] = \sum_{m, n} X[i + m, j + n] \cdot K[m, n] + b$$

- X : imagem ou tensor de entrada;
- K : kernel/filtro;
- b : viés;
- Y : saída da convolução.

Interpretação

A convolução funciona como um detector de padrões: bordas, texturas, transições espectrais e estruturas espaciais.

Filtros / kernels

- Filtros são matrizes de pesos aprendidos durante o treinamento.
- Cada filtro responde a um tipo de padrão presente na entrada.
- Vários filtros podem ser aplicados à mesma imagem, gerando múltiplos mapas de atributos.

Pontos fundamentais

- **Conectividade local:** cada neurônio olha apenas uma vizinhança da imagem.
- **Compartilhamento de parâmetros:** o mesmo filtro percorre toda a imagem.
- Isso reduz drasticamente o número de parâmetros em relação a redes totalmente conectadas.
- Em camadas iniciais, filtros costumam aprender bordas e texturas; em camadas profundas, estruturas mais complexas.

Camada convolucional

Estrutura básica

Imagem de entrada $\otimes$ filtro $\longrightarrow$ feature map

- É a principal unidade de extração de atributos da CNN.
- O kernel desliza sobre a imagem e realiza multiplicação elemento a elemento seguida de soma.
- O resultado preserva informação espacial e destaca padrões relevantes para a tarefa.

Na prática

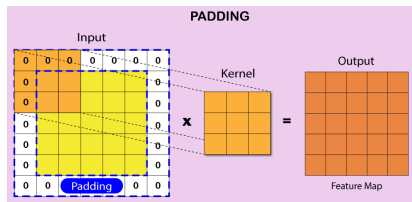
Em sensoriamento remoto, camadas convolucionais podem aprender:

- texturas urbanas ou agrícolas;
- padrões espectrais de vegetação, água e solo exposto.

Padding e stride

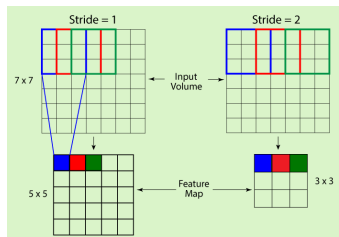
Padding

- Adiciona bordas, tipicamente com zeros, à entrada.
- Controla o tamanho da saída.
- Ajuda a preservar informação nas bordas da imagem.



Stride

- Define o passo de deslocamento do filtro.
- *Stride* maior $\Rightarrow$ saída menor.
- Reduz custo computacional, mas pode perder detalhe



Batch Normalization e ativação ReLU

Batch Normalization

Normaliza as ativações intermediárias da rede, reduzindo o *internal covariate shift* e estabilizando o treinamento.

- Permite taxas de aprendizado maiores;
- acelera convergência; atua como regularização leve;
- em CNNs, costuma ser aplicada entre convolução e ativação.

ReLU

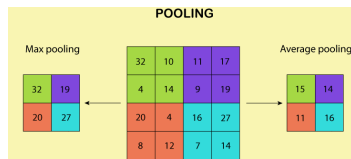
A função de ativação mais comum em CNNs:

$$y = \max(0, x)$$

- zera valores negativos; preserva positivos; introduz não linearidade;
- favorece treinamento eficiente em redes profundas.

Pooling / subamostragem

- Pooling reduz a resolução espacial dos mapas de atributos.
- Opera sobre cada *feature map* independentemente.
- Objetivos principais:
 - reduzir custo computacional; diminuir número de parâmetros;
 - controlar overfitting; aumentar invariância a pequenas variações espaciais.



Principais variantes

- **Max pooling:** mantém o valor máximo da janela;
- **Average pooling:** usa a média da janela;
- **Global pooling:** resume todo o mapa em um valor por canal.

Hierarquia de atributos em CNNs

- **Camadas iniciais:** bordas, cantos, texturas, gradientes.
- **Camadas intermediárias:** padrões compostos e arranjos locais.
- **Camadas profundas:** objetos, classes e contexto espacial mais abstrato.

Leitura intuitiva

Uma CNN constrói representações progressivamente mais ricas:

pixels → padrões locais → estruturas → semântica

- Em LULC, isso é crucial para combinar **textura local** e **contexto de vizinhança**.

Flatten, fully connected, dropout e softmax

- Após a última etapa convolucional, os mapas de atributos podem ser transformados em um vetor: **flatten**.
- Esse vetor alimenta camadas **fully connected**, que combinam os atributos aprendidos para decisão final.

Softmax

Na classificação multiclasse, a saída final é convertida em probabilidades:

$$p_k = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

- Para classificação binária, frequentemente usa-se função logística/sigmoide.

Saídas de CNNs: classificação vs segmentação

- Em **classificação**, a representação final pode ser resumida e enviada a camadas densas.
- Em **segmentação**, a rede preserva a estrutura espacial e produz um mapa de classes por pixel.

Classificação

features → flatten / global pooling → FC → softmax

Segmentação

encoder → decoder → mapa de classes por pixel

- Dropout pode ser usado como regularização, especialmente em partes densas ou no bottleneck.

Resumo do fluxo de uma CNN

Entrada $\rightarrow$ Convolução $\rightarrow$ BatchNorm $\rightarrow$ ReLU $\rightarrow$ Pooling $\rightarrow \dots \rightarrow$ Flare

- Extração automática de atributos;
- compressão progressiva da representação;
- decisão final baseada em padrões hierárquicos aprendidos.

Arquiteturas clássicas: panorama rápido

- **LeNet:** pioneira em reconhecimento de padrões visuais.
- **AlexNet:** popularizou deep learning em visão computacional.
- **VGG:** profundidade com filtros pequenos (3×3).
- **Inception:** múltiplas escalas processadas em paralelo.
- **ResNet:** conexões residuais para treinar redes muito profundas.

Mensagem prática

A evolução das arquiteturas buscou melhor equilíbrio entre **capacidade representacional**, **estabilidade de treinamento** e **eficiência computacional**.

Variantes modernas relevantes para raster

- **DenseNet:** reutilização intensiva de atributos.
- **MobileNet / EfficientNet:** eficiência computacional para inferência escalável.
- **Depthwise separable convolutions:** menos custo sem perder muita capacidade.
- **CNN + Transformers:** capturam padrões locais e dependências espaciais de longo alcance.

Para dados ambientais

Essas variantes são relevantes quando há restrições de memória, necessidade de inferência em tiles grandes ou integração multi-escala.

De classificação à segmentação semântica

- **Classificação:** um rótulo por imagem, patch ou tile.
- **Detecção:** localização de objetos de interesse.
- **Segmentação semântica:** um rótulo por pixel.

Por que isso importa em raster?

Muitas tarefas ambientais exigem previsão espacialmente densa:

- uso e cobertura da terra;
- delineamento de corpos d'água;
- mapeamento de queimadas, nuvens, gelo, áreas urbanas.

Arquiteturas de segmentação: FCN, U-Net e DeepLab

- **FCN:** substitui camadas densas por operações totalmente convolucionais.
- **U-Net:** arquitetura encoder-decoder com *skip connections*.
- **DeepLab:** usa convoluções dilatadas e ASPP para contexto multi-escala.

Leitura comparativa

- FCN: base conceitual;
- U-Net: excelente preservação de detalhes;
- DeepLab: forte modelagem de contexto espacial amplo.

Classificação de uso e cobertura do solo (LULC)

- Entradas possíveis:
 - bandas espectrais;
 - índices como NDVI e NDWI;
 - elevação, declividade e derivados;
 - dados SAR e séries temporais.
- Estratégias comuns:
 - janelas deslizantes;
 - tiling com sobreposição;
 - fusão multi-resolução.
- Métricas:
 - IoU;
 - F1 por classe;
 - matriz de confusão espacial.

Processamento de imagens de satélite e sensoriamento remoto

- Correção radiométrica e atmosférica.
- Ortorectificação e reprojeção.
- Máscaras de nuvens, sombras, água e pixels inválidos.
- Integração multi-sensor: óptico + SAR + topografia.
- Alinhamento espacial e temporal entre fontes heterogêneas.

Cuidado

Decisões de resampling, reprojeção e normalização podem alterar significativamente a qualidade do sinal aprendido pela rede.

Técnicas práticas para dados raster

- Normalização por banda e tratamento de valores faltantes.
- Data augmentation:
 - rotações e flips;
 - jitter espectral;
 - ruído sintético;
 - simulação de nuvens e oclusões.
- Tratamento de desbalanceamento:
 - oversampling;
 - losses ponderadas;
 - focal loss;
 - hard example mining.

Treinamento e funções de perda

- **Cross-entropy:** base para classificação por pixel.
- **Weighted cross-entropy:** útil em classes raras.
- **Dice loss:** favorece sobreposição espacial.
- **Focal loss:** enfatiza exemplos difíceis.

Boa prática

Combinar perdas, por exemplo:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{Dice}$$

pode melhorar desempenho em bordas e classes pequenas.

- Validação espacial é preferível a splits aleatórios quando há autocorrelação geográfica.

Interpretação e explicabilidade

- Técnicas comuns:
 - saliency maps;
 - Grad-CAM;
 - occlusion sensitivity;
 - Guided Backprop;
 - LRP.
- Objetivos:
 - verificar se o modelo usa regiões espacialmente plausíveis;
 - detectar artefatos e atalhos espúrios;
 - apoiar análise qualitativa por especialistas do domínio.

Limitação

Explicabilidade visual é aproximada: boa visualização não garante causalidade real.

Implementação e deploy

- Inferência em rasters grandes via *sliding window* com sobreposição.
- Suavização de bordas entre tiles.
- Quantização e pruning para cenários com restrição de hardware.
- Exportação em formatos geoespaciais:
 - GeoTIFF;
 - máscaras raster;
 - metadados CRS e transform preservados.

Boas práticas e armadilhas

- Evitar **leakage espacial**: separar regiões inteiras entre treino, validação e teste.
- Avaliar por classe e por área geográfica, não apenas por média global.
- Documentar:
 - pré-processamento;
 - escalas e resoluções;
 - máscaras aplicadas;
 - intervalos de valores;
 - critérios de rotulagem.
- Monitorar robustez temporal e transferência entre regiões.

Resumo final

- CNNs aprendem atributos espaciais por convolução, ativação e subamostragem.
- Padding e stride controlam a geometria da transformação.
- Pooling reduz dimensionalidade e melhora robustez.
- BatchNorm, dropout e escolhas adequadas de loss ajudam no treinamento.
- Em dados ambientais, CNNs são especialmente úteis para classificação e segmentação de rasters.

Pixels → atributos → contexto → decisão espacial

Recursos recomendados

- **Bibliotecas:** PyTorch, TensorFlow, Rasterio, GDAL.
- **Datasets:** Sentinel-2, Landsat, BigEarthNet, SpaceNet.
- **Arquiteturas para estudar:** U-Net, DeepLabV3+, ResNet, EfficientNet.
- **Extensões modernas:** self-supervised learning, masked autoencoders, CNN+Transformer.