

Aplicação de Aprendizagem Profunda na Detecção de Pneumonia

1nd Lucas Van-Lume Lima
Centro de Informática
Universidade Federal de
Pernambuco
Recife, Brasil
lvll@cin.ufpe.br

2nd Jorge Guilherme L. de V. Cabral
Centro de Informática
Universidade Federal de
Pernambuco
Recife, Brasil
jglvc@cin.ufpe.br

I. Objetivo(s)

O objetivo deste trabalho é implementar um sistema de inteligência artificial utilizando uma rede neural convolucional (CNN) com capacidade para identificar pneumonia, seja bacteriana ou viral, em radiografias torácicas. Este sistema visa aprimorar o diagnóstico precoce e preciso dessas enfermidades, acelerando o processo de identificação e tratamento. Com essa abordagem, espera-se obter uma detecção mais eficaz de pneumonia e, por conseguinte, contribuir para o tratamento oportuno e efetivo de pacientes afetados, melhorando suas perspectivas de recuperação e qualidade de vida.

II. Justificativa

A pneumonia é uma doença respiratória comum e potencialmente grave, responsável por uma alta taxa de mortalidade, especialmente em crianças e idosos. A distinção entre um pulmão saudável e um doente é crucial para a escolha do tratamento adequado, porém essa diferenciação pode ser desafiadora apenas com base em radiografias torácicas. O uso de IA para auxiliar na classificação dessas imagens pode aumentar a precisão e reduzir o tempo necessário para o diagnóstico, melhorando os resultados clínicos e a eficiência do sistema de saúde.

Além disso, o avanço nas técnicas de aprendizagem profunda, como o algoritmo *transfer learning*, tem mostrado resultados promissores na área de diagnóstico por imagem. Implementar essas técnicas no contexto da detecção do tipo de pneumonia pode aprimorar o processo diagnóstico e estabelecer uma base para aplicações futuras em outras áreas da radiologia e medicina.

III. Metodologia

Nesta seção, apresentamos a metodologia adotada para implementar o classificador de imagens de radiografias torácicas de pacientes com potencial diagnóstico de pneumonia bacteriana e viral. Sendo assim, o fluxo do projeto será dividido em etapas conforme descrito nos próximos tópicos:

A. Análise de dados

Inicialmente, é essencial obter um conjunto de dados representativo contendo imagens médicas de pulmões, que abranjam tanto casos de pneumonia quanto saudáveis. Esses dados foram obtidos a partir de [1], com dados de pacientes pediátricos de um a cinco anos de idade do Centro Médico Feminino e Infantil de Guangzhou, China.

Foram coletados e rotulados um total de 5856 imagens de raios-X de tórax de crianças, incluindo 4273 caracterizados com pneumonia, sendo eles 2780 casos de doença bacteriana e 1493 de doença viral. Além disso, o conjunto inclui 1583 amostras de imagens de raio-X saudáveis. A figura 1 apresenta uma amostra dessas imagens apresentadas no conjunto de dados.



Fig. 1. Exemplo de imagens do conjunto de dados com imagens de pulmões com e sem pneumonia.

TABELA I: Distribuição dos casos apresentados no dataset.

Tipo de Imagem	Quantidade de Casos
Pneumonia Bacteriana	2780
Pneumonia Viral	1493
Saudável	1583

B. Divisão do conjunto de Dados

Com 5856 dados coletados no total, eles foram divididos inicialmente em subconjuntos para serem usados nas etapas de treinamento e teste. Essa abordagem permite que o modelo seja treinado e testado em diferentes subconjuntos do dataset original para que sua eficiência e precisão sejam devidamente analisadas.

Dessa forma, 5218 imagens foram selecionadas para o subconjunto de treinamento do modelo, incluindo 3.883 caracterizadas como pneumonia (2.538 bacterianas e 1.345 virais) e 1.349 normais. Além disso, foi separado uma parte do conjunto de dados para teste com 234 imagens normais e 390 imagens de pneumonia (242 bacterianas e 148 virais) de 624 pacientes.

TABELA II: Distribuição de classes no conjunto de dados.

Tipo de Dado/ Divisão	Pneumonia	Normal	Total
Treinamento	3876	1342	5218
Teste	390	234	624

C. Tratamento dos dados

No contexto deste trabalho, o tratamento de dados desempenha um papel fundamental na preparação das imagens para o treinamento e avaliação dos modelos de aprendizado profundo. Utilizamos transformações de dados comuns disponíveis na biblioteca Keras para garantir que as imagens estejam em conformidade com os requisitos dos modelos pré-treinados. As etapas de pré-processamento incluíram:

- **Redimensionamento:** Todas as imagens foram redimensionadas para uma resolução de 256x256 pixels. Este passo

ajuda a padronizar o tamanho das imagens de entrada.

- **Normalização:** Normalização dos valores dos pixels de cada imagem, dividindo pelo desvio padrão dos pixels da imagem, resultando em uma distribuição de pixels com desvio padrão 1.
- **Zoom:** Aplicação de um zoom aleatório nas imagens dentro de um intervalo de 10%. Isso simula variações na distância focal e no tamanho dos objetos.
- **Rotacionamento:** Rotacionar as imagens em um intervalo de até 20 graus (para ambos os lados). Isso ajuda a tornar o modelo mais robusto a variações de orientação.

Esses procedimentos de pré-processamento são essenciais para garantir que as imagens de entrada estejam em conformidade com as expectativas dos modelos pré-treinados e, assim, facilitar o treinamento e a avaliação de nossos modelos de aprendizado profundo.

D. Transfer Learning

As redes convolucionais escolhidas foram implementadas utilizando a técnica de Transfer Learning, amplamente utilizada em técnicas de classificação de imagens do estado-da-arte. Em vez de treinar uma rede do zero, utiliza-se um modelo pré-treinado em um conjunto de dados de referência para um problema de classificação de imagens diferente, no nosso caso usamos ImageNet. Essa Abordagem reduz significativamente o tempo de treinamento e a necessidade de grandes quantidades de dados rotulados. Além disso, melhora a precisão do modelo ao aproveitar as características previamente aprendidas de um vasto e diverso conjunto de dados, resultando em uma maior eficiência e desempenho na nova tarefa de classificação.

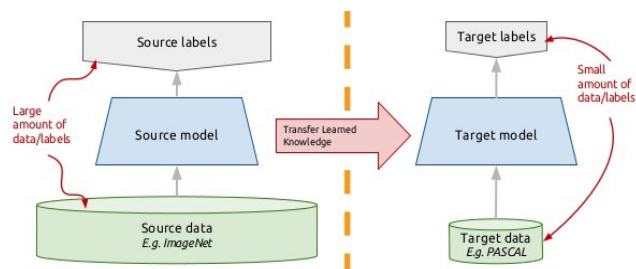


Fig. 2. Ilustração da aplicação da técnica Transfer Learning

IV. Arquitetura das Redes Neurais Convolutivas

Nesta etapa, apresentaremos as arquiteturas das redes neurais convolucionais (CNNs) que serão utilizadas no projeto: Inception V3, DenseNet-12 e Xception. Cada uma dessas redes possui suas particularidades e agem de forma distinta no nosso projeto.

A. Inception V3

Esta rede é conhecida por sua eficiência computacional e alta precisão em tarefas de classificação de imagens, utilizando uma combinação de convoluções de diferentes tamanhos e pooling para capturar características em múltiplas escalas. Graças à sua robustez, eficiência e profundidade, a Inception V3 se torna uma escolha popular para tarefas como detecção de doenças em imagens médicas.

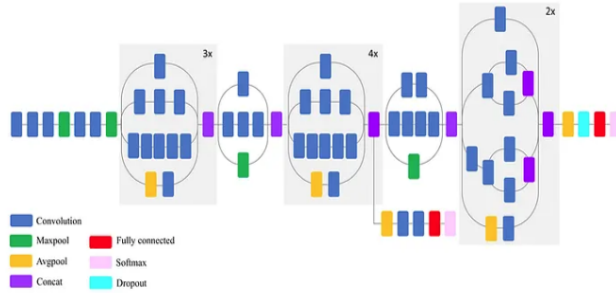


Fig. 3. Arquitetura da ImageNet V3

B. DenseNet-121

A rede DenseNet-121 foi proposta por Gao Huang, Zhuang Liu, Laurens van der Maaten e Kilian Q. Weinberger em 2016. As DenseNets possuem uma conexão densa entre camadas, onde cada uma recebe entradas de todas as camadas anteriores. Com isso, é esperado que ela tenha o melhor desempenho entre as redes selecionadas, pois possui a melhor acurácia com o dataset ImageNet.

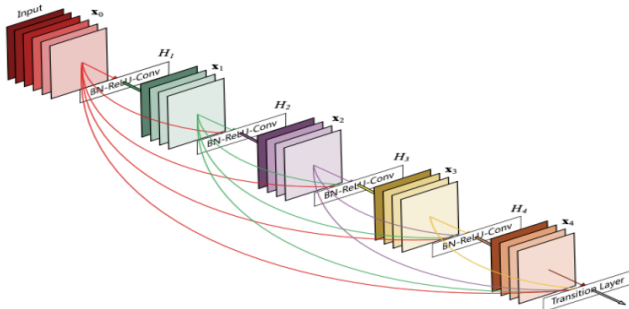


Fig. 4. Arquitetura da DenseNet-121

C. Xception

O modelo Xception é uma arquitetura de rede neural convolucional (CNN) que representa uma evolução em relação aos modelos tradicionais, sendo uma extensão do conceito de Inception. Proposto por François Chollet, criador do Keras, o Xception, cujo nome vem de “Extreme Inception”, é baseado inteiramente em convoluções separáveis em profundidade (depthwise separable convolutions), que reduzem significativamente o número de parâmetros sem comprometer o desempenho. Essa estrutura permite que o modelo aprenda de forma mais eficiente, com melhor aproveitamento dos dados e menor sobrecarga computacional. O Xception tem se destacado em diversas tarefas de classificação e detecção de imagens, superando modelos clássicos em benchmarks relevantes.

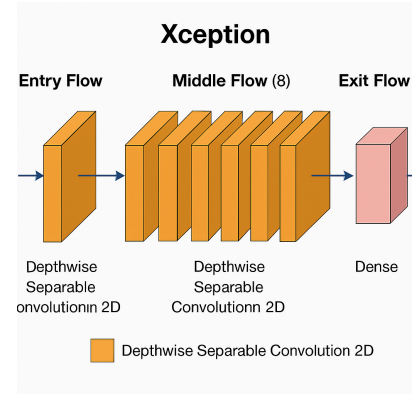


Fig. 5. Arquitetura da Xception

No entanto, assim como em qualquer projeto de aprendizado profundo, a escolha da arquitetura mais apropriada deve considerar as características específicas do problema e do conjunto de dados. Por essa razão, o presente artigo busca avaliar qual das arquiteturas aplicadas é mais eficaz na aprendizagem dos padrões presentes nos dados, considerando critérios como acurácia, perda e capacidade de generalização.

V. Experimentos

Nesta seção, descrevemos os procedimentos experimentais adotados para avaliar o desempenho de diferentes modelos de redes neurais convolucionais.

A. Estrutura Geral

Cada modelo foi composto por duas grandes partes:

- Camada de extração de características (feature extractor): responsável por extrair representações semânticas a partir das imagens de entrada.
- Camada de classificação personalizada (classifier head): projetada para interpretar essas representações e realizar a predição final, adaptada à tarefa binária (presença ou ausência de pneumonia).

A seguir, detalha-se a composição de cada uma dessas partes.

1. Camada de Extração de Características

As arquiteturas InceptionV3, Xception e DenseNet121 foram utilizadas sem suas camadas densas finais de classificação (`include_top=False`), preservando apenas a base convolucional. Essas redes, originalmente treinadas para classificar imagens em mil categorias, são capazes de aprender representações visuais robustas, o que se mostrou eficaz em domínios médicos mesmo com menor volume de dados.

Além disso, foi aplicado um método de pooling global (especificamente average pooling) ao final da base convolucional, de modo a reduzir a dimensionalidade dos vetores de características antes do repasse à cabeça classificadora. Essa etapa gera um vetor fixo por imagem, mantendo a eficiência computacional.

Importante destacar que todos os pesos da base foram congelados durante o treinamento inicial, ou seja, não foram atualizados. Essa decisão buscou evitar o sobreajuste devido ao tamanho relativamente pequeno do conjunto de dados e à complexidade dos modelos.

2. Cabeça de Classificação

Sobre a base extraída, foi adicionada uma estrutura densa personalizada, projetada com foco em robustez e regularização:

- Camada de normalização em lote (Batch Normalization): auxilia na estabilização do processo de aprendizado e acelera a convergência.

- Camadas de Dropout: aplicadas com taxas de 0,45 e 0,25 em diferentes pontos da arquitetura, com o objetivo de reduzir a dependência entre neurônios e mitigar o risco de sobreajuste.
- Camadas densas intermediárias: com 220 e 60 neurônios, utilizando a função de ativação ReLU, responsáveis por transformar o vetor de características em uma representação mais abstrata e discriminativa.
- Camada de saída: composta por um único neurônio com ativação sigmoide, retornando a probabilidade de a imagem pertencer à classe positiva (pneumonia).

Essa combinação de camadas busca o equilíbrio entre capacidade de aprendizado e controle de complexidade, permitindo que a rede aprenda padrões relevantes para a tarefa, mesmo com um número limitado de amostras.

3. Estratégia de Compilação

Os modelos foram compilados com a função de perda de entropia cruzada binária (binary crossentropy), adequada para problemas de classificação binária. O otimizador selecionado foi o Adamax, uma variação estável do algoritmo Adam, configurado com uma taxa de aprendizado inicial de 0,001. As métricas utilizadas para o monitoramento do treinamento foram:

- Acurácia: proporção de classificações corretas.
- Precisão (Precision): capacidade do modelo de evitar falsos positivos.
- Revocação (Recall): capacidade do modelo de identificar corretamente os casos positivos.

Essas métricas foram escolhidas devido à relevância clínica do problema abordado, em que tanto a identificação correta dos casos positivos quanto a minimização de falsos negativos são fundamentais para auxiliar o diagnóstico médico.

VI. Análise dos Resultados

Nesta seção serão abordados os resultados obtidos como resultados finais nas avaliações de cada arquitetura.

A. InceptionV3

O modelo InceptionV3 foi treinado por 20 épocas com aplicação de Dropout para reduzir o risco de overfitting, considerando o desbalanceamento do conjunto de dados, que contém mais exemplos de pulmões com pneumonia do que saudáveis. O modelo atingiu uma acurácia de 85% no conjunto de teste, com resultados equilibrados entre precisão e recall.

A tabela de métricas mostra que o modelo teve bom desempenho na classe PNEUMONIA, com recall de 0.93 e f1-score de 0.89, indicando que ele foi bastante eficaz em detectar corretamente os casos da doença. Por outro lado, o desempenho na classe NORMAL foi inferior, com recall de apenas 0.72, evidenciando que o modelo teve dificuldades em reconhecer pulmões saudáveis, o que pode gerar falsos positivos.

Ainda assim, o f1-score geral de 0.85 (weighted avg) indica um desempenho robusto para o número de épocas treinadas, principalmente em tarefas onde é preferível errar por excesso (falsos positivos) do que por omissão (falsos negativos).

	precision	recall	f1-score	support
NORMAL	0.86	0.72	0.78	234
PNEUMONIA	0.85	0.93	0.89	390
accuracy			0.85	624
macro avg	0.85	0.82	0.83	624
weighted avg	0.85	0.85	0.85	624

Fig. 6. Tabela das métricas da InceptionV3 em 20 épocas.

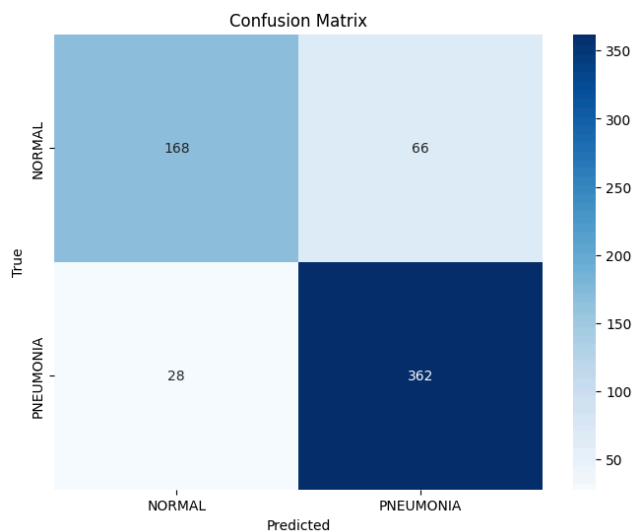


Fig. 7. Matriz de Confusão da InceptionV3 em 20 épocas.

B. Xception

O modelo atingiu uma acurácia de 83% no teste, o menor valor entre os três modelos avaliados. Apesar de apresentar valores razoáveis de loss durante o treinamento, o desempenho final sugere que o modelo não convergiu tão bem quanto os demais.

De acordo com a tabela de métricas, o modelo demonstrou desempenho mediano na classe NORMAL, com f1-score de 0.76 e recall de 0.74, o que indica certa dificuldade em classificar corretamente casos saudáveis. Já na detecção de pneumonia, os resultados foram mais positivos, com f1-score de 0.86 e recall de 0.88, mostrando que o modelo conseguiu capturar bem os padrões da doença.

O valor médio de f1-score (macro avg) foi de 0.81, e o weighted avg foi de 0.83, reforçando que, embora o modelo não tenha sido o mais eficiente, ainda apresentou resultados aceitáveis, especialmente considerando o baixo número de épocas e a complexidade da tarefa.

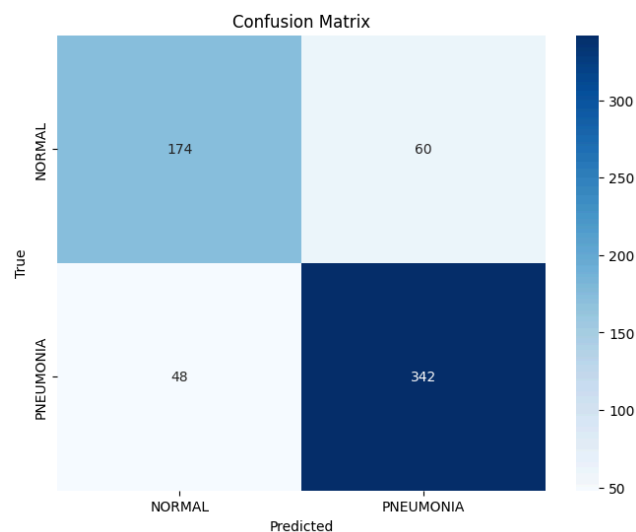


Fig. 8. Gráfico da loss da Xception em 20 épocas.

	precision	recall	f1-score	support
NORMAL	0.78	0.74	0.76	234
PNEUMONIA	0.85	0.88	0.86	390
accuracy			0.83	624
macro avg	0.82	0.81	0.81	624
weighted avg	0.83	0.83	0.83	624

Fig. 9. Tabela das métricas da Xception em 20 épocas.

C. DenseNet

O modelo DenseNet121 obteve o melhor desempenho entre todas as arquiteturas testadas, atingindo acurácia de 87% no conjunto de teste. Esse resultado reflete uma boa capacidade de generalização, mesmo considerando o risco de overfitting percebido pela diferença em relação à acurácia de treinamento.

As métricas mostram uma alta performance tanto na classe PNEUMONIA quanto na classe NORMAL. O modelo teve recall de 0.94 e f1-score de 0.90 para pneumonia, sendo extremamente eficaz na detecção da doença. Para casos normais, também apresentou precisão de 0.89 e f1-score de 0.81, indicando equilíbrio no desempenho entre as classes.

Com um macro f1-score de 0.86 e weighted avg f1-score de 0.87, o DenseNet121 se destacou pela sua precisão e sensibilidade em ambas as categorias, tornando-se a melhor escolha para o problema de detecção automática de pneumonia em radiografias de tórax neste projeto.

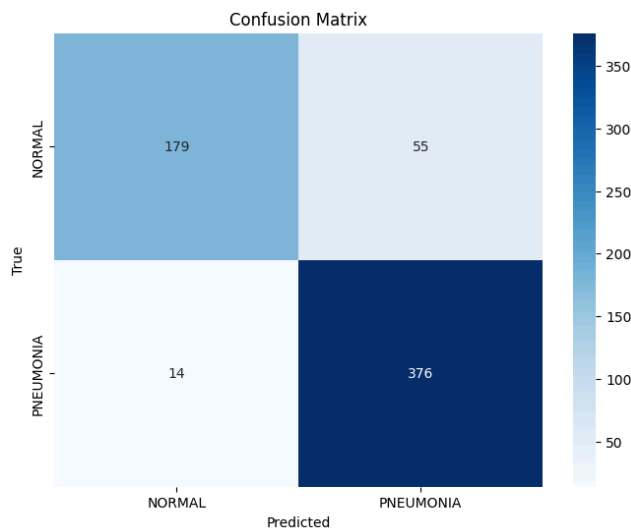


Fig. 12. Matriz de Confusão da DenseNet em 20 épocas.

VII. Conclusões

A partir dos resultados obtidos podemos verificar a eficácia e as limitações de cada arquitetura das redes convolucionais usadas no contexto de classificação de imagens de pulmões com pneumonia e saudáveis. Tendo a DenseNet se destacando como a mais precisa e robusta, seguida pela InceptionV3 que também mostrou

bons resultados, enquanto que a Xception teve desempenho inferior.

Durante o desenvolvimento e implementação do projeto, foram encontradas diversas dificuldades, que foram tratadas com estratégias específicas para garantir a eficiência e a precisão do modelo. Abaixo estão descritos os principais problemas enfrentados e as soluções aplicadas:

A. Limitação de GPU

As limitações de GPU foram um desafio significativo devido ao grande e complexo conjunto de dados, além do treinamento de três arquiteturas distintas. Esses fatores aumentaram consideravelmente a demanda por recursos computacionais, impactando a velocidade e eficiência do treinamento.

Sendo assim, para mitigar essa limitação, foram utilizadas múltiplas versões do Google Colab, aproveitando as diferentes GPUs disponíveis em cada sessão. Essa abordagem permitiu distribuir a carga de trabalho e potencialmente acelerar o processo de treinamento, otimizando os recursos disponíveis.

B. Overfitting

Possivelmente o maior problema enfrentado tenha sido lidar com os casos de overfitting. Devido a perda da capacidade do modelo de generalizar bem para novos dados e sendo particularmente problemático em modelos complexos como alguns usados nesse projeto, onde o modelo aprende ruídos e detalhes específicos do conjunto de treinamento, em vez de padrões generalizáveis.

Com isso, foram necessários aplicar técnicas como dropout e data augmentation. O dropout desativa aleatoriamente neurônios durante o treinamento, forçando o modelo a aprender representações mais robustas. Já a data augmentation, foi necessária para aumentar a variedade dos dados de treinamento através de transformações como rotação e zoom, melhorando a capacidade do modelo de generalizar para novos dados. Essas técnicas combinadas ajudaram a tornar os modelos mais robustos e eficazes.

VII. Referências

- [1] Kermay, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2.
- [2] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
- [3] Torrey, L., & Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI global.
- [4] Fagbemi, David. "Guide to Transfer Learning in Deep Learning". Disponível: <https://medium.com/@davidfagb/guide-to-transfer-learning-in-deep-learning-1f685db1fc94#:~:text=Transfer%20learning%20is%20an%20approach,data%20to%20the%20second%20model>