Universidade de Brasília – UnB
Campus Gama – FGA
Programa de Pós-Graduação em Engenharia Biomédica

Geração Automática de Laudos em Exames de Raios-X de Tórax com Explicabilidade Baseada em Atenção Aplicada a uma Rede Neural Recorrente

Josué Nascimento da Silva

Orientador: Cristiano Jacques Miosso

UnB – Universidade de Brasília

FGA – Faculdade Gama

Geração Automática de Laudos em Exames de Raios-X de Tórax com Explicabilidade Baseada em Atenção Aplicada a uma Rede Neural Recorrente

Josué Nascimento da Silva

Orientador: Cristiano Jacques Miosso

Dissertação de Mestrado em
Engenharia Biomédica

Publicação: 177A/2023

Brasília/DF, Setembro de 2023

UnB – Universidade de Brasília

FGA – Faculdade Gama

Programa de Pós-Graduação

Geração Automática de Laudos em Exames de Raios-X de Tórax com Explicabilidade Baseada em Atenção Aplicada a uma Rede Neural Recorrente

Josué Nascimento da Silva

Dissertação de mestrado submetida ao programa de pós-graduação em Engenharia Biomédica da Universidade de Brasília, como parte dos requisitos necessários para a obtenção do grau de mestre em Engenharia Biomédica

Aprovada por:

Cristiano Jacques Miosso

(Orientador)

Prof. Dr. Nilton Correia da Silva

(Examinador interno)

Dr. André Castilla

(Examinador externo)

Brasília/DF, Setembro de 2023

Ficha Catalográfica

Silva, Josué
Geração Automática de Laudos em Exames de Raios-X de Tórax com
Explicabilidade Baseada em
Atenção Aplicada a uma Rede Neural Recorrente
[Distrito Federal], 2023.
61p., 210 × 297 mm (FGA/UnB Gama, Mestrado em Engenharia Biomédica, 2023).
Dissertação de Mestrado em Engenharia Biomédica, Faculdade UnB Gama, Programa de Pós-Graduação em Engenharia Biomédica.

1. Inteligência Artificial
2. Geração Texutal
3. Explicabilidade
4. Controle de qualidade

I. FGA UnB/UnB.
II. Título (série)

Referência

Silva, Josué (2023). Geração Automática de Laudos em Exames de Raios-X de Tórax com
Explicabilidade Baseada em Atenção Aplicada a uma Rede Neural Recorrente. Dissertação de mestrado em engenharia biomédica, Publicação 177A/2023, Programa de Pós-Graduação, Faculdade UnB Gama, Universidade de Brasília, Brasília, DF, 61p.

Cessão de Direitos

Autor: Josué Nascimento da Silva

Título: Geração Automática de Laudos em Exames de Raios-X de Tórax com Explicabilidade Baseada em Atenção Aplicada a uma Rede Neural Recorrente

Grau: Mestre

Ano: 2023

---

josuetk63@gmail.com

Brasília, DF – Brasil

Resumo

Problemas específicos relacionados à análise de exames radiológicos têm sido amplamente documentados por pelo menos 50 anos. Entre as principais circunstâncias que levam a erros de diagnóstico, destacam-se avaliações realizadas por médicos em estágios iniciais de carreira, comunicação inadequada entre membros da equipe, jornadas noturnas, mudanças de turno e raciocínio falho.

Neste sentido, o uso de inteligência artificial como ferramenta para tomada de decisão e diagnóstico tem o potencial de auxiliar os profissionais de saúde a obter maior precisão e sensibilidade em suas análises, melhorando o tratamento dos pacientes. Nesse contexto, o objetivo deste trabalho é desenvolver uma arquitetura de modelo de inteligência artificial do tipo *encoder-decoder* capaz de gerar automaticamente laudos médicos com informações específicas extraídas das imagens dos exames. A ideia é que essas informações nas imagens reflitam os aspectos que orientam as decisões e análises indicadas no texto do laudo, representando uma contribuição em relação às abordagens predominantes na literatura, que geralmente se limitam apenas ao texto em si.

Com esse objetivo, foram utilizadas imagens de raio-X juntamente com seus respectivos laudos. Foi desenvolvida uma rede *encoder* baseada na arquitetura Densenet121 para extrair características dos exames, que são posteriormente traduzidas por um *decoder* baseado em *transformers*, permitindo aprender as relações semânticas entre as palavras, juntamente com a técnica de *long short term memory Long Short Term Memory* (LSTM) para a geração dos laudos.

Para relacionar as regiões das imagens com as palavras geradas, foi aplicada a técnica de *spatial attention*, que captura as regiões mais relevantes para a produção de palavras específicas pelo modelo. Esse processo foi aplicado em cinco condições: *lung hypoinflation*, *lung hyperdistention*, *cardiomegaly*, *aorta tortuous* e *spine degenerative*, resultando em cinco redes *encoder-decoder*. Durante o treinamento, foram obtidos valores de F1-score de 76% e área sob a curva (AUC) de 80% para o *encoder*. Os *encoders* foram avaliados utilizando validação cruzada para verificar sua capacidade de generalização em relação aos dados utilizados. Quanto ao *decoder*, na produção dos laudos, foram avaliados utilizando a métrica *recall-oriented understudy for gisting evaluation* (ROUGE), obtendo valores médios de 0.32.

Conclui-se que a arquitetura proposta é capaz de gerar laudos e marcações nas imagens dos exames, podendo servir como suporte para tomadas de decisões médicas. No entanto, é importante ressaltar uma limitação deste trabalho, que está relacionada ao escopo das patologias abordadas. A rede desenvolvida demonstrou eficácia nas condições específicas para as quais foi treinada, ou seja, *lung hypoinflation*, *lung hyperdistention*, *cardiomegaly*, *aorta tortuous*, e *spine degenerative*. No entanto, é fundamental reconhecer que a aplicabilidade do modelo permanece restrita a essas condições e não se estende a uma variedade mais ampla de patologias médicas. Esta limitação deve ser cuidadosamente ponderada ao avaliar

os resultados e ao considerar a aplicação do modelo em ambientes clínicos. Portanto, um dos principais objetivos em trabalhos futuros é a expansão do escopo das patologias abordadas, visando tornar o modelo mais abrangente e versátil em sua capacidade de auxiliar os profissionais de saúde em diagnósticos médicos diversos.

Palavras-chave: Radiologia, Geração textual, Encoder-Decoder, ExplainedAI.

Abstract

Specific issues related to the analysis of radiological exams have been extensively documented for at least 50 years. Among the primary circumstances contributing to diagnostic errors are evaluations conducted by early-career physicians, inadequate communication among team members, night shifts, shift changes, and flawed reasoning.

In this context, the use of artificial intelligence as a tool for decision-making and diagnosis has the potential to assist healthcare professionals in achieving greater accuracy and sensitivity in their analyses, thereby improving patient treatment. In this regard, the aim of this work is to develop an artificial intelligence model architecture of the encoder-decoder type capable of automatically generating medical reports with specific information extracted from exam images. The idea is for the information in the images to reflect the aspects guiding the decisions and analyses indicated in the report text, representing a contribution compared to prevailing approaches in the literature, which typically focus solely on the text itself.

To achieve this goal, X-ray images were used alongside their respective reports. An encoder network based on the Densenet121 architecture was developed to extract features from the exams, which are subsequently translated by a decoder based on transformers, allowing the model to learn the semantic relationships between words, along with the long short-term memory (LSTM) technique for report generation.

To link image regions with generated words, the spatial attention technique was applied, capturing the most relevant regions for producing specific words by the model. This process was applied under five conditions: lung hypoinflation, lung hyperdistention, cardiomegaly, aorta tortuous, and spine degenerative, resulting in five encoder-decoder networks. During training, F1-score values of 76% and an area under the curve (AUC) of 80% were achieved for the encoder. The encoders were evaluated using cross-validation to assess their generalization capacity with respect to the data used. Regarding the decoder, for report generation, evaluations were performed using the recall-oriented understudy for gisting evaluation (ROUGE) metric, obtaining average values of 0.32.

In conclusion, the proposed architecture is capable of generating reports and annotations on exam images, potentially serving as a support tool for medical decision-making. However, it is important to highlight a limitation of this work, which is related to the scope of the addressed pathologies. The developed network has demonstrated effectiveness in specific conditions for which it was trained, namely lung hypoinflation, lung hyperdistention, cardiomegaly, aorta tortuous, and spine degenerative. However, it is essential to recognize that the model's applicability remains restricted to these conditions and does not extend to a broader range of medical pathologies. This limitation should

be carefully considered when evaluating the results and contemplating the model's application in clinical settings. Therefore, one of the primary objectives in future work is to expand the scope of the addressed pathologies, aiming to make the model more comprehensive and versatile in its ability to assist healthcare professionals in various medical diagnoses.

Keywords: Radiology, Text Generation, Encoder-Decoder, ExplainedAI.

Sumário

Lista de Tabelas

Lista de Quadros

Lista de Figuras

Lista de Nomenclaturas e Abreviações

CNN *Convolutional Neural Network*

LSTM *Long Short Term Memory*

BN *batchnormalization*

GAP *Global Average Pooling*

RNA Rede neural artificial

1 Introdução

A análise e interpretação de exames é uma atividade comum entre os médicos, especialmente na especialidade da radiologia, que se concentra na análise de imagens médicas para fins diagnósticos e planejamento de tratamentos. Por meio de um treinamento longo e especializado, os radiologistas desenvolvem a capacidade de reconhecer características e padrões relevantes, traduzindo suas impressões em laudos clínicos. No entanto, durante essa análise clínica, podem ocorrer erros que afetam tanto a segurança dos pacientes quanto os aspectos econômicos da saúde [39]. Estima-se que, nos Estados Unidos, os erros médicos resultem em gastos anuais de 17 a 50 bilhões de dólares [40, 27], além de causarem a perda de aproximadamente 100 mil vidas anualmente [27]. No ambiente hospitalar, os erros são a terceira principal causa de morte, ficando atrás apenas das doenças cardíacas e do câncer [37].

Neste contexto, os erros na radiologia são um tema que envolve tanto aspectos médicos quanto qualitativos. Para melhorar os resultados das análises nessa especialidade, é necessário compreender a essência e a origem desses erros [20]. Dentro do campo radiológico, aproximadamente 15% dos diagnósticos correspondem a erros relacionados à identificação de patologias nos exames ou atrasos no reconhecimento das mesmas. Quando comparados com exames de autópsia, essas discrepâncias aumentam para cerca de 20% [9].

De fato, problemas na interpretação de exames ocorrem mesmo entre radiologistas experientes. Em cerca de 30% dos casos com possível achado diagnóstico, os médicos não conseguem identificar ou relatar adequadamente a anormalidade, apresentando uma taxa de 2% de falsos positivos para casos negativos, além de uma taxa diária de erro que varia entre 3% e 4% [40]. Os desafios na interpretação de exames radiológicos persistem há pelo menos 50 anos [5].

Outro aspecto que influencia a eficiência dos diagnósticos desses profissionais é a jornada de trabalho. Durante o turno noturno, os radiologistas tendem a apresentar menor confiabilidade à medida que a fadiga aumenta [16]. A fadiga pode resultar na não detecção de anormalidades sutis, que não são devidamente relatadas nos laudos. Além disso, médicos mais fatigados podem levar mais tempo para determinar se as descobertas presentes nos exames são suficientemente significativas para serem identificadas como alterações [16].

Com o objetivo de auxiliar os radiologistas, este trabalho propõe a implementação de um modelo de inteligência artificial capaz de gerar laudos de forma automática. Embora existam estudos na literatura sobre a geração de laudos com base em imagens de exames radiológicos, esses modelos não possuem a capacidade de identificar as regiões das imagens que são relevantes para a geração das palavras. Nesse contexto, esta pesquisa busca desenvolver um modelo que possa gerar laudos descritivos dos exames de imagem, ao mesmo tempo em que identifica, de forma explicável, as regiões específicas da imagem que estão associadas a cada palavra presente no laudo.

1.1 Definição do problema

Considerando os aspectos observados, nota-se que os erros de interpretação de exames de imagem radiológicos são influenciados por diversos fatores humanos, como experiência, horário de trabalho e trocas de turnos. Apesar dos avanços tecnológicos, os problemas relacionados a erros na análise e transcrição de laudos persistem há pelo menos 50 anos [5]. No entanto, mesmo quando os médicos têm acesso a informações adicionais, como o histórico clínico do paciente, a utilização desses dados não resulta em uma melhoria estatisticamente significativa nos diagnósticos em comparação com os casos em que essas informações externas não são consideradas [15].

Portanto, este estudo propõe desenvolver um modelo de inteligência artificial capaz de gerar laudos de exames de raio-X e identificar as regiões da imagem que correspondem a cada palavra gerada para a produção do laudo.

A geração dos laudos, juntamente com a identificação das áreas correspondentes às palavras geradas, oferece aos médicos um auxílio na tomada de decisão, possibilitando a identificação de achados que poderiam passar despercebidos ou não serem visíveis devido às circunstâncias previamente mencionadas.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo desta pesquisa consiste no desenvolvimento e avaliação de um modelo de aprendizado profundo, fundamentado na arquitetura *encoder-decoder*, com o propósito de gerar laudos radiológicos a partir de exames de raio-X e identificar as regiões na imagem associadas a cada palavra gerada.

1.2.2 Objetivos Específicos

Os objetivos específicos para que o objetivo geral seja alcançado são:

- Implementação e avaliação de estratégias de pré-processamento para pré-processamento para as imagens de raio-X.
- Implementação e avaliação de estratégias de pré-processamento para pré-processamento para o texto dos laudos.
- Implementação de programas para treinamento seguida da geração de modelo treinado de *encoder* para extração de características dos exames de imagem.
- Implementação de programas para treinamento seguida da geração de modelo treinado de *decoder* para interpretar as *features* da imagem transcrevê-las em texto.
- Avaliação do desempenho do sistema completo proposto para geração dos laudos com anotações nas imagens radiológicas.

2 Fundamentação Teórica

Neste capítulo, serão abordados os fundamentos que servirão de base para a pesquisa, apresentando conceitos sobre a realização de análises de exames radiológicos e os erros que podem surgir dessas análises. Com isso, será apresentada a arquitetura de geração de laudos focada na arquitetura *encoder-decoder* (codificador-decodificador) e como o uso de redes *encoder-decoder* para a geração automática de laudos, juntamente com a explicabilidade das palavras geradas, pode auxiliar os médicos na prevenção desses erros. Para isso, serão utilizadas as seguintes condições: *lung Hypoinflation* (hipoinsuflação pulmonar), *Lung Hyperdistention* (hiperdistensão pulmonar), *Cardiomegaly* (cardiomegalia), *Aorta tourtuos* (aorta tortuosa) e *Spine degenerative* (degeneração da coluna vertebral). Esses termos serão utilizados como achados base para a geração e validação da arquitetura *encoder-decoder* proposta.

2.1 Erros Radiológicos

A análise de exames radiológicos é um processo complexo que envolve tomadas de decisão sob condições de incerteza. O radiologista é responsável por identificar, avaliar e interpretar os achados clínicos presentes na imagem com o objetivo de produzir um diagnóstico. No entanto, a produção de laudos com base nesses achados pode apresentar problemas de interpretação e diagnóstico [6, 7, 10]. É importante destacar que a interpretação dos exames não é realizada de forma binária, ou seja, não se limita à identificação de achados normais ou presença de alterações. O profissional de radiologia pode ter acesso a informações como o histórico familiar e do próprio paciente, o que pode impactar na interpretação dos resultados. Apesar da importância do processo de análise e interpretação médica dos exames radiológicos, ele é suscetível a erros [10].

O ciclo de produção e análise de uma imagem radiológica pode ser dividido em quatro fases [28]: pré-procedimento, procedimento, pós-procedimento e clínica. A fase pré-procedimento ocorre durante a consulta médica, na qual o radiologista define se há a necessidade de realizar um exame e, caso haja, qual o tipo de exame e outros fatores. A fase do procedimento é o momento em que o paciente é informado sobre o exame que será realizado, incluindo instruções sobre a preparação necessária, como jejum, medicamentos

ou contraste. Na fase pós-procedimento, o radiologista interpreta o exame em busca de achados clínicos e produz um laudo que descreve essas informações. Por fim, na fase clínica, todos os dados gerados durante a consulta, incluindo os resultados do exame, são utilizados em conjunto, para determinar o tratamento e as ações que devem ser realizadas em benefício da saúde do paciente.

Dentre as quatro fases, as que mais apresentam erros com impacto no diagnóstico final são as que ocorrem durante a realização do procedimento e após o procedimento [28]. Neste trabalho, iremos focar na fase de pós-procedimento para entender as causas desses erros e propor uma abordagem baseada em inteligência artificial que pode, juntamente com desenvolvimento subsequente, potencialmente contribuir para mitigar sua ocorrência.

Alguns fatores contribuem para ocorrerem erros nessas fases do ciclo. Dentre estes erros, destacam-se a má interpretação dos exames [8, 10, 7, 60], mudança de turnos no trabalho [17, 42, 12], e potencialmente a pouca experiência com análise radiológica, no caso de profissionais iniciantes [11].

Os erros que ocorrem no processo de análise e interpretação dos exames são classificados e caracterizados por [60]. Esses erros estão listados na Tabela 2.1. Entre os 12 tipos de erros, os mais frequentes são a subleitura (42%), satisfação da procura (22%) e raciocínio defeituoso (9%). Esses três erros mais frequentes estão associados à interpretação dos exames.

Para reduzir a chance desses erros, o uso da inteligência artificial pode ser uma ferramenta de suporte para auxiliar e orientar o diagnóstico médico [10]. Portanto, este trabalho apresentará um modelo de inteligência artificial para a geração de laudos que ofereça explicabilidade das palavras em regiões dos exames de raio-X. A conexão entre a explicabilidade das palavras e os tipos de erros está relacionada à capacidade de detectar e corrigir possíveis equívocos. Ao fornecer explicabilidade, o modelo é capaz de justificar as decisões e inferências que faz, tornando o processo mais transparente para os médicos e pacientes.

2.2 Arquitetura do modelo de inteligência artificial

Arquitetura de um possível modelo de inteligência artificial para geração automática de laudos de exames radiológicos. A composição de uma rede neural para geração de laudos pode ser constituída a partir de dois componentes principais: um *encoder* e um *decoder* [14, 39]. O arranjo destes dois componentes em conjunto é conhecido como arquitetura *encoder-decoder*. O *encoder* é responsável por aprender e extrair características do exames de imagem, e o *decoder* é responsável por traduzir essas características em texto, permitindo assim, a geração do laudo que descreve o conteúdo presente no exame [39].

Tabela 2.1. Diferentes tipos de erros radiológicos que podem ocorrer durante a interpretação de exames. Fonte: [60].

Tipo de erro	Descrição
Complacência	O achado é identificado, mas é atribuído a uma causa errada.
Raciocínio defeituoso	O achado é encontrado e interpretado como anormal, porém é atribuído a uma causa errada.
Falta de conhecimento	O radiologista não consegue interpretar o exame.
Comunicação pobre	O achado clínico é identificado e interpretado de forma correta, porém a mensagem descrita no laudo falha em informar a relevância clínica.
Complicações	Complicações que podem ocorrer durante o exame.
Sub Leitura	O achado não foi encontrado.
Técnica	O achado não foi encontrado devido à técnica aplicada.
Localidade	O achado não é encontrado devido à lesão não estar na área de interesse da imagem.
Falta de conhecimento	O achado não foi encontrado devido ao exame anteriormente realizado ou não informado no laudo.
Exame prévio	O achado não foi encontrado devido ao exame anteriormente realizado ou não informado no laudo.
Histórico	O achado não é encontrado devido a histórico clínico incompleto.
Satisfação de procura	O achado não é encontrado devido não continuar o processo de identificação após uma primeira anormalidade ser encontrada.

A organização do *encoder* é normalmente formada por uma *Convolutional Neural Network* (CNN), em que a imagem é representada em vetores densos [43], que são elementos do chamado espaço latente [29]. Já a arquitetura do *decoder* é formada por uma *Recurrent Neural Network*, que associa o espaço latente gerado pelo *encoder* a uma sequência de palavras [43]. Porém, antes de entrar em detalhes da arquitetura *encoder-decoder* é importante entender do que uma rede neural artificiais é composta e como é seu processo de aprendizagem.

2.3 Rede Neurais Artificiais

As Redes Neurais Artificiais Rede neural artificial (RNA) são estruturas computacionais inspiradas no sistema neural biológico. As RNAs são modeladas matematicamente para se assemelharem aos neurônios biológicos, e essa modelagem é conhecida

como neurônio artificial. No processo de sinapse, as células nervosas são conectadas através de axônios e dendritos, permitindo o processo de transmissão de informações entre elas. De maneira semelhante, os neurônios artificiais se conectam para permitir que um sinal de entrada passe por eles e produza uma resposta de saída [29].

O trabalho apresentado por McCulloch e Pitts modelou um sistema computacional baseado na lei do *tudo ou nada* do sistema nervoso [38], que afirma que um impulso não será gerado a menos que tenha uma determinada intensidade, conhecida como limiar de excitação. Com base na modelagem de McCulloch e Pitts, Rosenblatt desenvolveu uma estrutura chamada de perceptron, que é o modelo matemático que representa o neurônio artificial [46].

2.3.1 Perceptron

O *perceptron* é a unidade mais simples de uma rede neural artificial, originalmente utilizado para tarefas de identificação de padrões de separação binária, produzindo regiões de identificação de padrões presentes nos sinais de entrada. Na Figura 2.1, é apresentada a modelagem de um *perceptron*, no qual alguns componentes constituem essa unidade: $X_1, X_2, X_3, ...X_n$ são os componentes do vetor de entrada X, $W_0, W_1, W_2, ...W_n$ compõe o vetor W de pesos sinápticos aprendidos durante o treinamento e b é o *bias* ou viés da rede [46, 29, 44].

Nos perceptrons, as entradas de cada camada são multiplicadas por pesos sinápticos e os resultados dessas multiplicações são somados de forma gerar a saída dada por

$$z = bW_0 + \sum_{i=1}^{\infty} X_i W_i.$$

A saída do perceptron é obtida após a aplicação de uma função não linear f na variável z ou seja,

$$y = f(z).$$

As equações anteriores representam matematicamente o funcionamento dos *perceptrons*. Na primeira equação, temos a multiplicação dos sinais de entrada pelos pesos sinápticos, seguida da soma de todos esses valores. O resultado obtido é então somado ao viés do neurônio, gerando assim o valor de z. As operações realizadas até essa etapa são do tipo linear. Para que o neurônio seja capaz de aprender relações não lineares entre entradas e saídas, é necessário adicionar uma operação não linear, chamada de função de ativação. Essa função gera o valor de y, que é a saída da rede. Como os *perceptrons*

têm por característica realizar separações binárias, eles são utilizados principalmente para tarefas de classificação [44].

Similarmente aos neurônios biológicos, os perceptrons podem ser conectados entre si, gerando múltiplas camadas [44]. Utilizando apenas um neurônio, as regiões de tomada de decisão podem ser vistas como semi-planos. Com a adição de dois neurônios, podemos obter regiões convexas e, com múltiplos neurônios conectados, podemos obter regiões arbitrárias que melhor se adaptam às características dos sinais. Na Figura 2.2, é demonstrado um exemplo de como essas regiões são formadas [44].

Para que o *perceptron* possa gerar os limites de decisão desejados, é necessário ajustar os valores dos pesos sinápticos de forma apropriada. Esses valores são atualizados continuamente durante o processo de treinamento, que é feito utilizando o algoritmo de *backpropagation* [32, 44]. A combinação dos *perceptrons* com o algoritmo de *backpropagation* possibilitou o desenvolvimento de arquiteturas de redes neurais artificiais (RNAs) mais complexas e eficientes. A seguir, é explicado como o *backpropagation* atua na atualização dos pesos sinápticos.

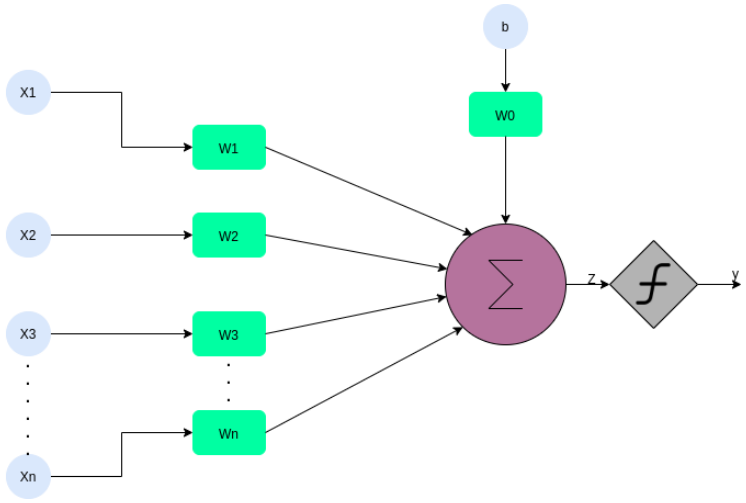

Figura 2.1. Representação de um neurônio artificial com seus componentes. Os sinais de entrada da rede são representados pelo vetor X, que representa a informação de entrada para classificação. Já o vetor W os pesos sinápticos que são aprendidos durante a fase de treinamento. Os valores dos pesos sinápticos são multiplicados com os valores de entradas e somados, e após esse estágio aplicada uma função de ativação que é definida de acordo com o propósito da rede. Fonte: Autoria própria.

2.3.2 Algoritmo de retropropagação backpropagation

O algoritmo de treinamento de uma RNA do tipo perceptron ou perceptron multicamada ajusta as conexões sinápticas que ocorrem entre os neurônios, a fim de permitir que o processo de aprendizado exemplos pré-rotulados, no caso do treinamento supervisionado. As sinapses são agregadas em redes multicamadas [32]. O algoritmo apresentado

por Rumelhart modifica os pesos da rede através de interações com o objetivo de encontrar os valores que melhor realizem o mapeamento entre os vetores de entrada do neurônio artificial e a saída esperada [47].

Quando existe uma relação linear entre os vetores de entrada e de saída, torna-se simples definir regras de aprendizagem que ajustam os pesos sinápticos com o intuito de reduzir a diferença entre o valor real e o valor obtido através da rede [47].

Redes neurais artificiais são treinadas em iterações chamadas de épocas. Uma época completa consiste em apresentar todas as amostras disponíveis para treinamento como entrada. Após a última amostra ser processada pela rede, o algoritmo de *backpropagation* avalia o quanto a alteração dos valores dos pesos sinápticos reduzirá o erro da rede, e ajusta os vetores de pesos na direção que mais reduz esse erro, porém com um fator de escala denominado taxa de aprendizagem.

A função de perda ou *loss function* é utilizada durante o treinamento da rede neural e corresponde ao erro das inferências em relação os valores reais esperados. Durante o treinamento com o uso do *backpropagation*, o modelo busca diminuir esse erro em relação a todo o conjunto de dados existentes. A função de perda originalmente utilizada é o erro quadrático dado por

$$E = \frac{1}{2}\sum_{i=1}(y_i - t_i)^2,$$

em que y_i representa o valor de saída do neurônio i, t_i é o valor desejado de saída para esse neurônio, e N é o número total de neurônios na camada. É importante notar que o fator $1/2$ é introduzido por conveniência matemática, auxiliando na simplificação das expressões durante as operações de derivação realizadas no algoritmo de minimização do erro.

O *perceptron*, juntamente com o algoritmo de *backpropagation*, possibilita o desenvolvimento e implementação de arquiteturas variadas de RNAs. Nas seções a seguir serão discutidas algumas dessas arquiteturas que serão utilizadas neste trabalho.

2.3.3 Cross-entropy loss

A função de perda do tipo entropia cruzada (do inglês, cross-entropy loss) é uma medida da diferença entre a distribuição de probabilidade prevista pelo modelo e a distribuição de probabilidade verdadeira para um determinado conjunto de dados. É comumente usada como uma função de perda para problemas de classificação, e é dada por:

$$L = -\sum_{i}(y_i \log(\hat{y}i) + (1 - yi) \log(1 - \hat{y}_i)),$$

Na equação y é o valor real da classe e y_i é a predição gerada pelo modelo tanto a classe predita quando a classe real são valores entre 0 e 1. A equação calcula a soma da perda para cada classe (i) e o objetivo é minimizar esse valor de perda durante o treinamento do modelo [54].

2.4 Convolution Neural Network (CNN)

As redes neurais convolucionais (CNN, do inglês *Convolutional Neural Network*), são uma classe de redes neurais artificiais estruturadas para processar informação normalmente organizada na forma de múltiplos vetores. Por exemplo, essa informação pode estar na forma de imagens compostas por três ou mais matrizes, correspondentes aos chamados canais vermelho, verde e azul (RGB, do inglês *red*, *green*, *blue*). A estrutura da CNN é composta por três tipos principais de camadas, denominadas camada de convolução, camada de *pooling* e camada completamente conectada (em inglês, *fully connected*, como é conhecida). [29, 2].

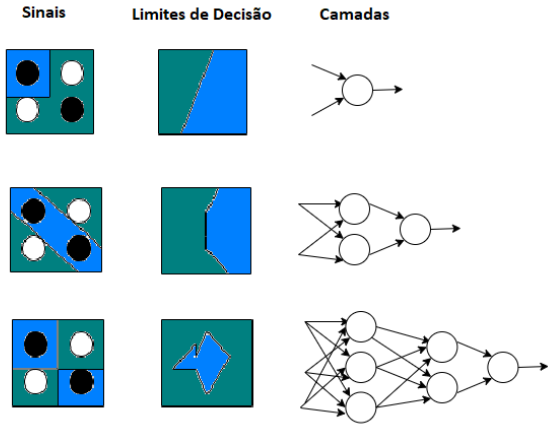

Figura 2.2. Exemplo de limites de decisão, os círculos pretos e brancos são tipos distintos de sinais, as regiões verdes e azuis representam como são separados os sinais de acordo com os limites de decisão. Na coluna Limites de Decisão temos a representação geométrica em um plano 2 dos limites gerados. Na coluna de camadas temos a quantidade de perceptrons que são conectados em respectivamente que geram os limites de decisão. Fonte: Adaptado de [44].

2.4.1 Camada Convolucional

As camadas convolucionais são um aspecto distintivo das CNNs em comparação com outros tipos de redes neurais [59, 29, 2]. Essas camadas permitem que as CNNs aprendam, por meio do treinamento, a extrair automaticamente as características relevantes para a classificação, eliminando a necessidade de uma etapa prévia de extração explícita de características. As camadas convolucionais em si são operações lineares que realizam a filtragem linear de sinais. No entanto, elas são comumente seguidas por funções de ativação não-lineares, como a convolução e as funções de ativação, respectivamente [59].

A convolução é uma operação linear aplicada a um sinal ou imagem. Ela consiste em multiplicar ponto a ponto o sinal pela resposta impulsional e somar os resultados, considerando todas as posições possíveis. Essa operação é aplicada em toda a imagem, gerando mapas de características que representam diferentes elementos presentes nela. A Figura 2.3 exemplifica o processo convolucional, no qual um kernel 3x3 é aplicado a uma imagem de entrada.

A convolução discreta é matematicamente representada pela seguinte equação:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k].$$

Nessa equação, $y[n]$ representa a saída da convolução, $x[k]$ é o sinal de entrada, $h[n-k]$ é a resposta impulsional (também conhecida como filtro) e a soma é realizada para todas as possíveis posições de sobreposição entre o sinal de entrada e a resposta impulsional. Essa operação de convolução discreta é amplamente utilizada em processamento de sinais e processamento de imagens para realizar filtragem, detecção de características e outras operações essenciais [45, 51].

O janelamento realizado pelo kernel entre as aplicações dos cálculos de convolução é chamado de *stride*, que define a distância entre as convoluções consecutivas [2, 59]. O tamanho do *stride* e o formato da matriz do kernel são definidos previamente durante a hiperparametrização da rede antes do treinamento. Quando a rede é projetada para realizar uma função específica, esses parâmetros são definidos para melhor atender ao objetivo final.

O processo de treinamento da CNN consiste em identificar o kernel que melhor se adapta ao contexto em que está sendo aplicado, de acordo com o objetivo da rede e o dataset trabalhado.

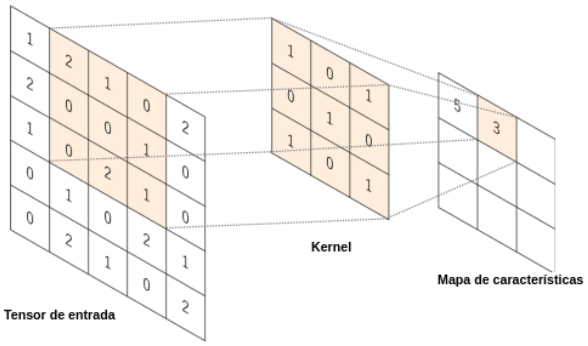

Figura 2.3. Exemplo de aplicação de convolução a o kernel exemplificado foi aplicado um kernel de 3x3 com stride de 1 resultando no número 9 no mapa de x características e está sendo aplicado no segundo janelamento. Fonte: [59].

2.4.2 Camada de Pooling

Outra estrutura que compõe a CNN é a camada de pooling [59]. Essa camada fornece uma redução na resolução do mapa de características gerado pela camada convolucional, reduzindo a complexidade para as camadas subsequentes. Nessa etapa, não há aprendizado de novos parâmetros e, assim como as camadas convolucionais, elas são hiperparametrizadas. As camadas de *pooling* geralmente utilizadas são a de *max pooling* e a de *average pooling*.

2.4.2.1 Maxpooling

A camada de *maxpooling* tem como objetivo reduzir a dimensionalidade, buscando extrair características relevantes para a classificação. Essa redução é feita por meio da análise de blocos 2x2 na matriz resultante da camada anterior, que corresponde à matriz de características construída nas camadas convolucionais. Nessa análise, apenas o valor máximo de cada bloco é mantido, descartando-se os demais valores.

Na Figura 2.4, é ilustrado o funcionamento da camada de *maxpooling*. A matriz de 4x4 representa um mapa de características originado das camadas convolucionais com *stride* de 2x2. A camada de *maxpooling* gera uma matriz reduzida a partir desse mapa de características, que será passada para as próximas camadas convolucionais, diminuindo a dimensionalidade dos dados para as camadas subsequentes da rede e facilitando o aprendizado dos parâmetros seguintes.

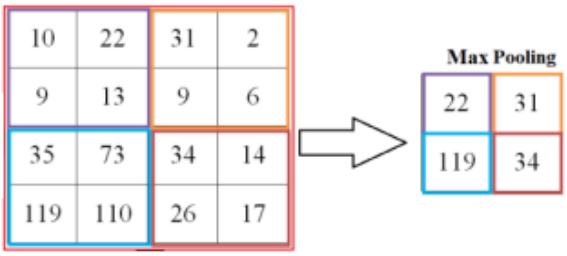

Figura 2.4. Aplicação de maxpooling em uma matriz, a camada de pooling aplicada é de 2x2. Em cada bloco, o valor máximo presente é selecionado. Ou seja, entre os quatro valores na submatriz 2x2, apenas o valor máximo é mantido. Os valores máximos selecionados são então usados para construir uma nova matriz que é uma versão reduzida da matriz de entrada. Cada valor na nova matriz representa o valor máximo de um bloco na matriz de entrada original. Fonte: [1].

2.4.2.2 Average Pooling

O *average pooling* funciona de maneira semelhante ao *max pooling*. Os mapas de características gerados pelas camadas convolucionais também são divididos em blocos de tamanho 2x2. No entanto, ao contrário do *max pooling*, em vez de considerar apenas os valores máximos de cada bloco, é calculada a média dos valores de cada bloco. As médias obtidas formam uma nova matriz, que representa a informação extraída e é transmitida para as camadas subsequentes.

2.4.2.3 Global Average Pooling

A camada de *Global Average Pooling* (GAP) realiza uma redução de dimensionalidade ainda maior em comparação ao *maxpooling* e *average pooling*. Essa camada recebe o tensor de características e reduz o tamanho das características para um único escalar para cada matriz desse tensor. Esse escalar é obtido calculando a média dos componentes de cada matriz. A Figura 2.5 ilustra o resultado dessa operação, onde é gerado um vetor final.

A camada de GAP é aplicada antes da camada totalmente conectada (*fully connected layer*), trazendo duas vantagens para o processo de aprendizagem: reduz a quantidade de parâmetros que precisam ser aprendidos durante o treinamento e permite que a CNN

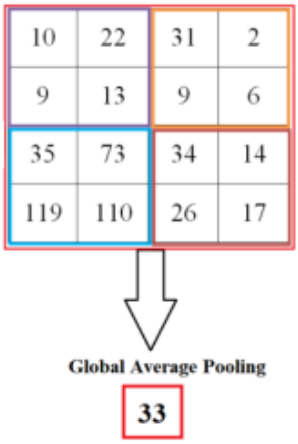

Figura 2.5. Aplicação de Global Average Pooling em uma matriz, a camada de pooling aplicada é de 2x2. Em cada bloco, a média dos valores presentes na submatriz 2x2 é calculada. Os valores médios calculados são usados para construir um vetor de saída. Cada valor no vetor representa a média dos valores em um bloco da matriz de entrada original. Fonte: [1].

possa receber entradas com tamanhos variados.

2.4.3 Função de ativação

Em uma rede convolucional, as saídas geradas pelas operações de convolução são submetidas a funções não lineares, conhecidas como funções de ativação [59]. Essas funções determinam se um neurônio artificial será ativado ou não. Elas são aplicadas após as camadas lineares, no caso da CNN, as camadas do tipo *fully connected* e convolucionais.

A aplicação de funções de ativação torna a relação entre os dados de entrada da rede neural e a saída obtida não linear, permitindo que a CNN aprenda padrões mais complexos. No entanto, as funções de ativação devem ser diferenciáveis, para que o erro do *backpropagation* possa ser usado durante o treinamento [3]. A seguir, serão apresentadas as funções de ativação comumente usadas em CNNs e que também serão aplicadas neste trabalho.

- **Função sigmoidee** corresponde a um número real x, e a saída gerada é um valor entre 0 e 1 dado por:
$$f(x) = \frac{1}{1 + e^{-x}}.$$
- **Função tangente hiperbólica** corresponde a um número real x, e a saída gerada

é um valor entre -1 e 1 dado por:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

- **Função do tipo unidade linear retificada (ReLU)** Converte qualquer valor de entrada para um valor positivo dado por:

$$f(x) = \max(0, x).$$

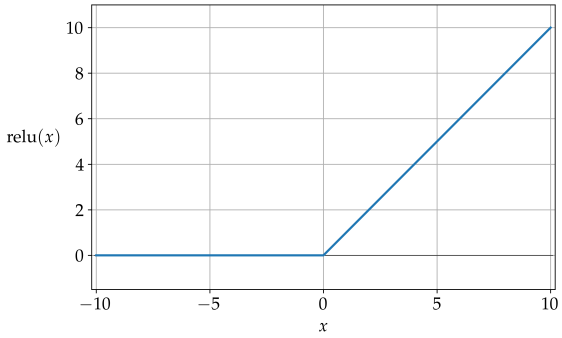

Figura 2.6. A função ReLU é caracterizada por um eixo y que se estende de 0 a 10, e um eixo x que varia de -10 até 10. Na ReLU, observamos um comportamento de valores constantes até alcançar o valor máximo real de x.

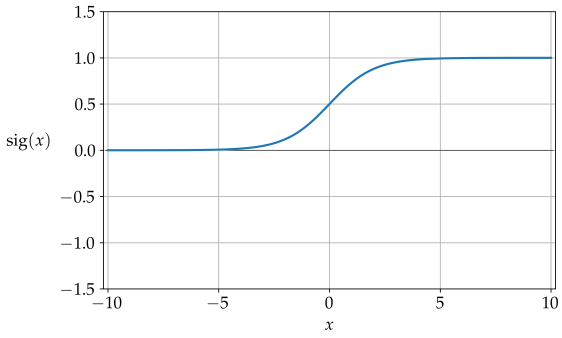

Figura 2.7. A função sigmoide mapeia seus valores de entrada para um intervalo entre 0 e 1, apresentando um formato de curva em S.

Nas Figuras 2.6, 2.7 e 2.8 é possível observar os diferentes comportamentos de cada uma das funções de ativação. É possível notar os contornos suaves existentes nas funções sigmoide e tanh, enquanto a ReLU apresenta um crescimento a uma taxa constante, para valores positivos de entrada.

Além das funções de ativação mencionadas anteriormente, neste trabalho será utilizado também a função softmax. Normalmente, a softmax é aplicada em problemas de classificação multiclasse, em que para cada amostra de entrada da rede neural é gerada uma classificação com pelo menos duas classes. A função softmax, aplicada a uma entrada z_i, é dada por:

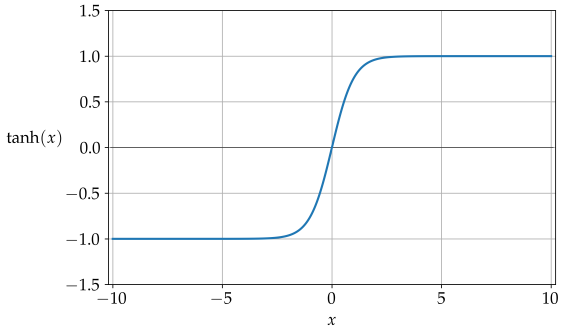

Figura 2.8. transforma os valores de entrada para um intervalo entre -1 e 1, exibindo um formato de curva similar ao da função sigmoide.

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1} e^{z_j}}.$$

No cálculo do valor da softmax, z_i é o vetor com as predições geradas pelo modelo.

2.4.4 Fully Connected Layer

Após a geração dos mapas de características pela última camada convolucional ou pela camada de *pooling*, eles são convertidos em de matrizes para vetor-coluna e conectados a um ou mais modelos de *fully connected layers*, também conhecidos como camadas densas, em que cada entrada é multiplicada por um peso individualizado, antes que se gere a soma poderada de saída daquela camada.

Na última camada do *fully connected layer*, é aplicada a função de ativação que permite alcançar o objetivo de aplicação da CNN. Por exemplo, em uma aplicação de classificação binária, pode ser aplicada a função sigmoide, onde os parâmetros aprendidos pelas *fully connected layer* são normalizados, que faz com que os resultados sejam entre 0 e 1. Para uma aplicação em que se busca identificar múltiplas classes em uma imagem, pode ser aplicada a softmax, que gera um vetor contendo as probabilidades de todas as classes presentes na imagem.

A composição e o arranjo de todos esses componentes permitem uma grande variedade de arquiteturas de CNN, seja devido ao empilhamento das camadas, ao tamanho dos kernels ou à seleção de qual pooling será aplicado. Dentre essas diversas arquiteturas, temos a densenet-121, comumente utilizada em tarefas de classificação de imagens. Na Figura 2.9 é ilustrado como as camadas se conectam.

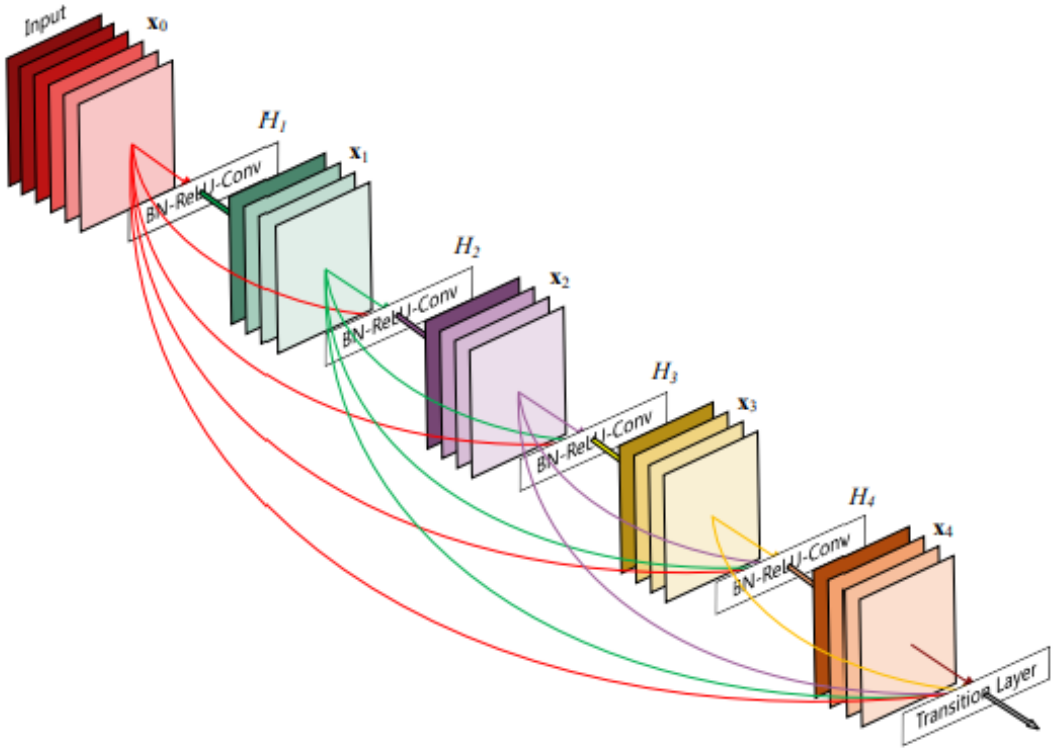

Figura 2.9. Representação do fluxo de dados na *densenet*. As camadas são interligadas em si, as características aprendidas nas camadas anteriores são concatenadas e seguem juntas para as próximas camadas. Fonte: [22].

2.4.5 Dropout

Dropout é uma técnica de regularização aplicada em modelos de inteligência artificial que consiste em descartar aleatoriamente neurônios em cada camada durante as épocas de treinamento. Isso evita que o modelo aprenda a mapear somente os exemplos de treinamento às classes correspondentes, sem generalização para outros padrões semelhantes, o que corresponderia à situação indesejada de overfitting [53]. A taxa de descarte, ou seja, a porcentagem de unidades descartadas, é geralmente definida como um hiperparâmetro a ser otimizado.

2.4.6 DenseNet

A densenet é outra arquitetura de rede convolucional[22]. Nela existem blocos chamados de *denseblock*, formados por múltiplas camadas convolucionais conectadas entre si [22].

Na Figura 2.9 é possível observar como é realizada a passagem de informação entre os blocos. X_0 é o primeiro conjunto de mapas de características de entrada no *denseblock*,

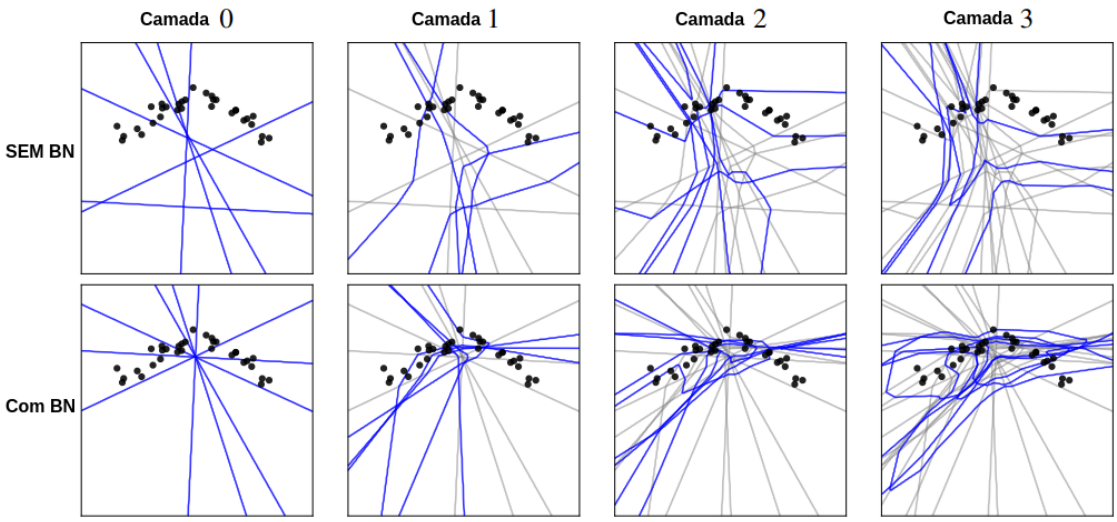

Figura 2.10. Exemplo de aplicação de batchnormalization entre em camadas de um CNN, as linhas azuis são os hiperplanos aprendidos e inseridos pela camada atual, as linhas cinza são os planos inseridos pelas camadas anteriores e os pontos escuros é o dado utilizado. Fonte: [4].

gerado pela camada convolucional fora do bloco. Nesse primeiro mapa, são aplicadas as operações H_l, que consistem em *batch normalization* (BN), *ReLU* e *pooling*, onde l é o índice da camada. O resultado de H_l é passado para a próxima camada convolucional interna, gerando um novo mapa de características que é concatenado com o mapa anterior, e a operação é reaplicada após a passagem de cada camada.

Os *denseblocks* resolvem o problema de *vanishing gradient* que afeta as redes neurais que utilizam o *backpropagation* para calcular o valor do gradiente. Durante a aplicação da regra da cadeia, as sucessivas multiplicações de valores numéricos em ordens de grandeza menores do que os pesos envolvidos geram problemas numéricos relacionados à resolução numérica finita. De fato, um número arbitrariamente alto de camadas impõe problemas de convergência e estabilidade à medida que o algoritmo de *backpropagation* é aplicado na sucessão a partir da camada mais externa e em direção às camadas mais internas [41]. À medida que a rede neural se torna mais profunda, esse problema é intensificado. Os parâmetros são adquiridos ao identificar mudanças nas saídas da rede, mas quando essas mudanças são excessivamente pequenas, os valores de saída diminuem, resultando em um aprendizado lento por parte das redes [22]. Esse problema acontece devido à escolha da função de ativação aplicada em cada camada, normalmente funções como sigmoide que recebem um valor muito grande e o definem em um range entre 0 e 1.

Os componentes dos *denseblocks* são responsáveis por corrigir esse problema. O *batchnormalization* (BN) é uma camada adicionada à rede neural que adapta os planos para melhor se adequar aos dados. O *batchnormalization* reduz a distância entre os dados e os planos gerados pela rede neural. Essa diminuição permite uma inicialização mais efici-

Tabela 2.2. Variações das camadas de acordo com as arquiteturas da densenet.

Camadas	Saída	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-265
Convolução	112 x 112	7 x 7 conv, stride 2			
Pooling	56 x 56	3 x 3 max pool, stride 2			
Bloco Denso (1)	56 x 56	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 6$
Camada de Transição (1)	56 x 56	1 x 1 Conv			
	28 x 28	2 x 2 AVG Pooling, stride 2			
Bloco Denso (2)	28 x 28	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 12$
Camada de Transição (2)	28 x 28	1 x 1 conv			
	14 x 14	2 x 2 Avg pooling, stride 2			
Bloco Denso (3)	14 x 14	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 64$
Camada de Transição (3)	14 x 14	1 x 1 conv			
	7 x 7	2 x 2 Avg pooling, stride 2			
Bloco Denso (4)	7 x 7	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 48$
Camada de Transição (4)	1 x 1	7 x 7 Global Avg pooling			
Camada de Classificação		Fully connection,softmax			

ente dos parâmetros, fazendo com que a rede treine mais rapidamente, e também corrige o problema de mudança de covariância interna [4, 23]. Durante cada época de treinamento, os pesos são atualizados, fazendo com que a distribuição das entradas dos neurônios seja diferente. O *batchnormalization* fixa a distribuição das entradas dos neurônios [23].

O *batchnormalization* faz com que os dados do *batch* de treinamento se adaptem para terem média 0 e variância igual 1 [23]. A Figura 2.10 ilustra a diferença entre as camadas que aplicam o BN e aquelas que não o fazem, mostrando como os planos gerados a partir dos dados de treinamento são mais próximos dos dados reais quando o BN é utilizado. Isso é especialmente importante para a inicialização eficiente dos parâmetros e para corrigir o problema da mudança de covariância interna durante o treinamento da rede neural [23]. Durante cada época de treinamento, os pesos são atualizados, o que pode alterar a distribuição das entradas dos neurônios. O *batchnormalization* fixa a distribuição das entradas dos neurônios, garantindo que a rede neural seja mais robusta e treine mais rapidamente [23].

A função de ativação ReLU é o componente final presente na *DenseNet* que ajuda a contornar o problema de *vanishing gradient* [22]. Esse fenômeno ocorre quando as derivadas das funções de ativação, notadamente aquelas como a sigmoide e a tangente hiperbólica, atingem valores exponencialmente reduzido à medida que a informação é *backpropagation* pela rede. Isso resulta em gradientes quase nulos, ocasionando uma atualização minimamente expressiva dos pesos da rede. Como resultado, a capacidade de aprendizado das camadas iniciais é comprometida, dificultando a captura de padrões complexos [41].

Os pesos das camadas de transição, que ficam fora do bloco dos *denseblocks*, também

são compartilhados dentro do bloco, pois as primeiras características aprendidas são utilizadas. Os *denseblocks* melhoram o fluxo de informação entre as camadas, permitindo que as camadas subsequentes acessem os mapas de características das camadas anteriores, o que possibilita o reuso de características e gera modelos menores que necessitam de menos parâmetros de aprendizado. Os pesos entre as camadas são compartilhados entre si, de forma que as características aprendidas pelas primeiras camadas sejam efetivamente utilizadas pelas camadas mais profundas.

A *DenseNet* é formada por quatro blocos de *denseblocks* e três blocos de transição, sendo que as variações arquiteturais das redes são definidas pelos terceiro e quarto blocos, que possuem 121, 169, 201 e 264 camadas. As arquiteturas dessas redes podem ser observadas em detalhes na Tabela 2.2.

2.5 Recurrent Neural Network (RNN)

A *Recurrent Neural Network* é um tipo de arquitetura de rede neural que encontra sua aplicação em dados temporais ou sequenciais [50]. As RNN permitem que dados sequenciais sejam organizados de forma que seja possível capturar a correlação entre os dados próximos [50]. Suas aplicações podem ser encontradas em análises de textos, séries temporais, mercado de ações, genomas ou dados numéricos em geral.

Além dessas aplicações, as RNN podem ser utilizadas em imagens ou vídeos, sendo que no caso das imagens podem ser decompostas em uma série e tratadas como uma sequência. Em aplicações com vídeo, quadros anteriores podem ser utilizados para entender o contexto do quadro atual e auxiliar na predição do próximo.

Quando se trata de aplicações textuais, as RNN exercem atividades de modelagem de linguagem, geração de texto, reconhecimento de fala e classificação textual. [49] A estrutura sequencial presente nos textos permite que sejam modeladas de maneira semelhante aos vídeos, sendo que as palavras anteriores ajudam a entender o contexto atual e predizer o próximo. Uma arquitetura que representa bem essas características da rede é a *long short term memory* (LSTM) [21].

2.5.1 Long Short Term Memory

Na arquitetura da LSTM apresentada em [21], as camadas recorrentes contêm blocos de memória compostos por células que armazenam o estado temporal. O fluxo de informações é gerenciado dentro dessas células com o auxílio de três estruturas: *input gate*, *output gate* e *forget gate*.

A Figura 2.11 apresenta a célula de memória existente na LSTM, bem como as divisões

dos *gates* que definem o fluxo de informação. O *forget gate* define quais informações devem ser mantidas ou descartadas. O vetor de entrada X_t, a saída do bloco anterior (também conhecida como *hidden state*) e a memória do bloco anterior são somados, e o resultado é submetido, juntamente com o termo de viés (bias), à função de ativação sigmoide. Caso um valor próximo de 1 seja obtido da célula anterior, a informação é considerada necessária e utilizada no bloco seguinte.

O *input gate* é utilizado para atualizar o estado da célula atual. Os valores de X_t e H_{t-1} são passados por uma segunda sigmoide, onde valores próximos de 1 são importantes e próximos de 0 não. Para regularizar os valores da célula, são passados pela função tangente hiperbólica. Esses valores são multiplicados e somados com C_t, que representa a memória da célula anterior, gerando assim a saída de memória da célula atual.

O *output gate*, por outro lado, determina *hidden state*. O *hidden state* da célula anterior passa através de uma nova sigmoide, sendo multiplicado com o resultado da função tangente hiperbólica aplicada no estado atual da célula. Isso define quais informações devem ser passadas para as próximas camadas. Cada célula de memória da LSTM pode ser observada como um *perceptron*.

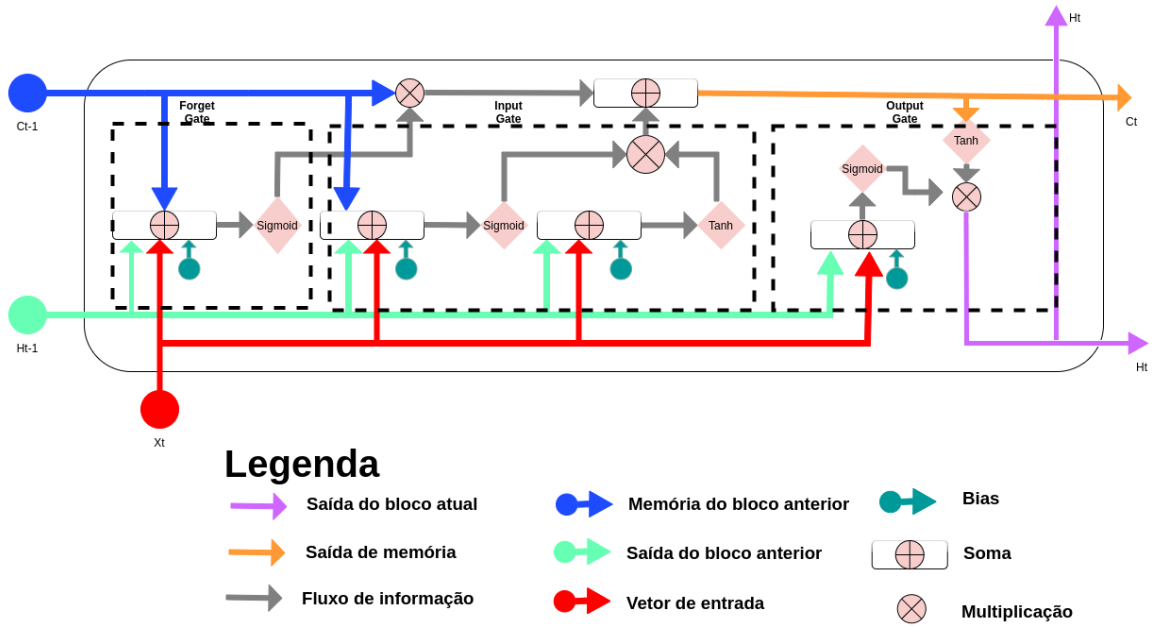

Figura 2.11. Representação do funcionamento da célula de memória da LSTM. O termo X_t representa a vetorização do textual de entrada para a célula, enquanto H_t refere-se aos estados ocultos da célula anterior, os quais encapsulam os padrões aprendidos e as informações prévias. Por sua vez, C_t representa os estados de memória, determinando se a informação atual deve ser incorporada para modificar o estado da célula atual.

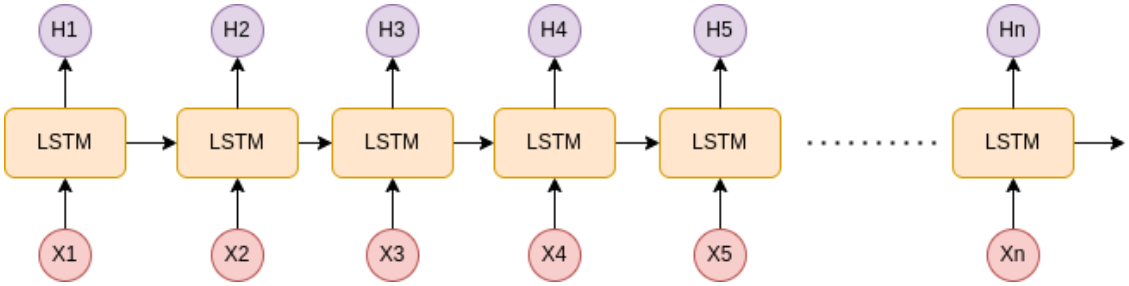

Figura 2.12. União de blocos de memória da LSTM. Na Figura é representado o fluxo de informação, onde as células compartilham as informações entre si, onde cada unidade define quais informações devem prosseguir adiante. Fonte: Autoria própria.

Neste trabalho, a LSTM é implementada como um componente do *decoder* que recebe as características extraídas pela CNN e as converte para texto. A estrutura sequencial de um dado textual permite utilizar a característica das células de memória [25] para entender as associações entre as imagens e os textos, permitindo realizar tarefas de geração textual [50]. A palavra a ser gerada depende da saída gerada pela célula anterior [23]. No trabalho [48], é apontada a necessidade de inserir informações de contextualização para geração de texto. Neste trabalho, iremos utilizar os *transformers* para auxiliar na melhoria e adicionar contextualização, o que é discutido na seção 2.5.2.

2.5.2 Transformers

O modelo *transformers* [57] é baseado na estrutura de *self-attention* [34], que possibilita extrair múltiplas representações de uma sequência. A Figura 2.13 ilustra a estrutura do transformers, que também utiliza os denominados embeddings, que são vetores de representação textual. Já o *positional encoding* é extraído para indicar a posição de cada palavra na sequência, sendo concatenado com o *embedding*.

A arquitetura dos *transformers* também é uma arquitetura *encoder-decoder*. O *encoder* é composto por 6 blocos idênticos, que são compostos por uma camada de *multi-head attention* juntamente do bloco residual [19]. O *decoder* utiliza os blocos do *encoder*, fazendo uso adicional da camada de *masked multi head attention*. Nesta camada os *embeddings* são movidos uma posição à esquerda.

A operação base dos *transformers* é a atenção, que utiliza três vetores: Q, K e V. O vetor Q representa a pergunta que o modelo está tentando responder, ou seja, a palavra ou conjunto de palavras para as quais o modelo está tentando encontrar um contexto. O vetor K representa as palavras do texto de entrada que o modelo usa para encontrar o contexto da palavra de Q, e o vetor V representa as informações de contexto que o modelo usa para produzir a saída. A Equação matemática é dada por,

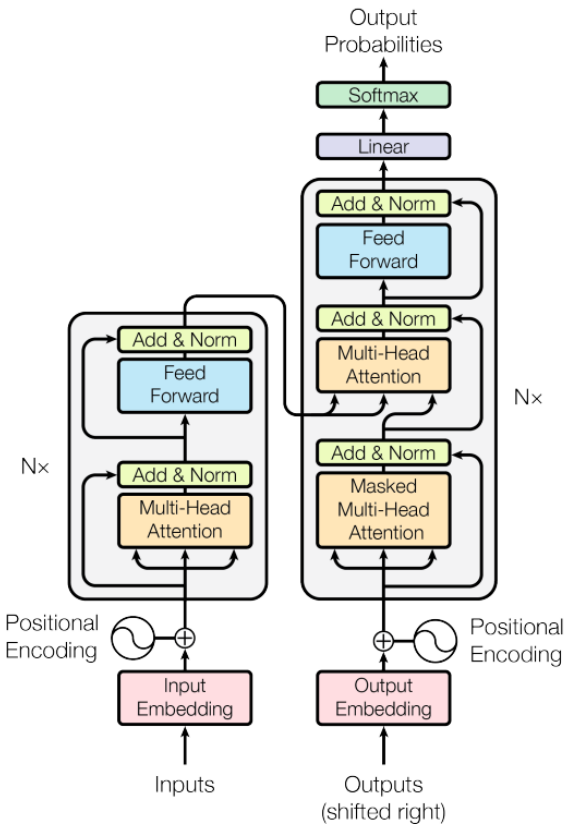

Figura 2.13. Exemplo de célula de processamento dos transformers que utiliza o mecanismo de atenção multi-head para examinar relações complexas entre palavras em diferentes perspectivas. Além disso, incorpora embeddings de posição para capturar a ordem das palavras na sequência e embeddings de entrada para compreender o significado das palavras. Essa combinação de técnicas permite que o Transformer compreenda tanto o contexto semântico quanto a estrutura sequencial, tornando-o altamente eficaz em várias tarefas, desde tradução até geração de texto. Figura adaptada de [57].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

A camada de atenção opera por meio da multiplicação do vetor Q por uma matriz K, seguida pela divisão do resultado pela raiz quadrada da dimensão do vetor K. Em seguida, aplica-se uma função softmax para obter as probabilidades de cada posição na matriz K. Essas probabilidades são então multiplicadas pela matriz V para obter a saída final da camada de atenção.

A camada de atenção é responsável por permitir que o modelo considere as relações entre as palavras em um texto de entrada, o que é essencial para entender o contexto e produzir uma saída mais precisa.

2.5.3 Embedding

O processo de *embedding* consiste em representar cada palavra de um texto por meio de um vetor de valores contínuos reais. Essa representação mapeia palavras com contextos e significados similares para vetores próximos entre si [58]. Cada vetor gerado pelo *embedding* é capaz de resumir de forma sucinta tanto a sintaxe quanto a semântica da palavra. Essas características do texto representadas pelos vetores de *embedding* são utilizadas no treinamento de redes neurais [58].

2.6 Explicabilidade de Inteligência Artificial

A explicabilidade de modelos de inteligência artificial tornou-se um tema de interesse crescente entre os pesquisadores [56]. A explicabilidade busca fornecer uma interpretação humana do que a rede neural ou outro modelo de inteligência artificial está observando durante a realização da tarefa para a qual foi treinada.

2.6.1 Explicabilidade Encoder-Decoder

No trabalho apresentado por [36], é proposto um método de explicabilidade para arquiteturas *encoder-decoder* que é incorporado como um componente durante o treinamento da rede. Dessa forma, a arquitetura *encoder-decoder* é treinada desde o início utilizando o módulo que posteriormente irá produzir a relação entre as palavras e as imagens. A camada de explicabilidade [36] foi desenvolvida para redes que utilizam uma combinação de CNN e RNN para a geração de texto, e é chamada de "atenção espacial", que relaciona a imagem com as palavras geradas.

O modelo de atenção espacial produz um vetor de contexto que é dado por

$$c_t = g(V, ht),$$

em que V denota o espaço latente das características extraídas pela rede convolucional, enquanto h_t representa os estados ocultos aprendidos pela rede neural recorrente (RNN). Em seguida, é apresentado como a atenção é aplicada às características extraídas pela CNN em conjunto com os estados ocultos da RNN, gerando o resultado

$$z_t = \text{attention}(\tanh(V + ht)1^T).$$

O 1 representa um vetor onde todos os elementos são iguais a 1. Ao final é aplicada a função $softmax$ no valor de z_t obtendo assim o valor de α,

$$\alpha_t = \text{softmax}(z_t).$$

O valor de α representa os pesos da atenção sobre as características de V [36], Para das palavras, é aplicado a função *sigmoide* sobre o valor de h_t obtendo assim um valor g_t,

$$g_t = \text{sigmoide}(W_t \cdot h_t).$$

Tendo obtido o valor de g_t, ele é multiplicado com z_t obtendo assim $z_t h$,

$$z_t h = g_t \cdot z_t.$$

Finalmente, o valor $z_t h$ é concatenado com o *embedding* textual, o que fornece o termo

$$\beta = \text{softmax}([z_t h; ew_h)]).$$

Dessa forma, o símbolo β representa o conteúdo textual juntamente com a atenção obtida a partir do vetor de características da imagem. Esse valor é utilizado na célula de memória da rede neural recorrente LSTM para gerar a próxima palavra e produzir um novo vetor h_t. Esse processo é iterativo e continua até que a palavra gerada represente o fim da sentença.

A camada de *spatial attention* tem como objetivo destacar automaticamente as regiões mais relevantes da imagem, permitindo que o sistema de descrição de imagens foque em gerar descrições precisas dessas áreas específicas. Essa abordagem resulta em descrições mais precisas e informativas da imagem como um todo, uma vez que está direcionando sua atenção para as áreas mais relevantes. Dessa forma, a camada de *spatial attention* permite que o sistema de descrição de imagens seja mais eficiente no processo de geração de descrições precisas.

2.7 Métricas de avaliação

As métricas de avaliação de aprendizado são medidas quantitativas que permitem avaliar o desempenho de um modelo em relação a um conjunto de dados. A escolha adequada da métrica depende da tarefa para a qual o modelo será empregado. Neste caso, as métricas serão divididas em duas categorias: as métricas de avaliação do *encoder* e as de avaliação do *decoder*.

2.7.1 Métricas do encoder

As métricas utilizadas para avaliar a qualidade dos modelos de *encoder* incluem o f1-score e o AUC. O f1-score é uma métrica que combina precisão, que na epidemiologia é conhecida como especificidade, e recall, que na epidemiologia corresponde à sensibilidade. A precisão (ou especificidade na epidemiologia) mede a proporção de previsões corretas positivas em relação ao número total de previsões positivas, enquanto o recall (ou sensibilidade na epidemiologia) avalia a proporção de previsões corretas positivas em relação ao número total de casos positivos verdadeiros. Essas métricas desempenham um papel essencial na avaliação da eficácia e da confiabilidade dos modelos de *encoder*.

$$f1 = \frac{2 \times (\text{especificidade} \times \text{sensibilidade})}{(\text{especificidade} + \text{sensibilidade})}$$

Os valores do f1-score variam entre 0 e 1, onde 1 representa a melhor qualidade do modelo e 0 a pior. Quanto mais próximo de 1 for o valor da métrica, melhor é a qualidade do modelo.

O AUC é uma métrica utilizada na identificação de classificações binárias, avaliando a capacidade do modelo de distinguir entre duas classes. Ele é calculado com base na curva de *Receiver Operating Characteristics* (ROC), que demonstra a relação entre os verdadeiros positivos e a taxa de falsos positivos. Os valores do AUC variam entre 0 e 1, sendo que 1 indica que o modelo é capaz de diferenciar perfeitamente entre as duas classes e 0 indica que o modelo é incapaz de distinguir entre as classes.

2.7.2 Métricas do decoder

A métrica utilizada para avaliar a qualidade do texto gerado pelo *decoder* é a *Recall-Oriented Understudy for Gisting Evaluation* (ROUGE) [33]. Essa métrica foi originalmente desenvolvida para comparar sistemas de geração de resumos, mas é comumente utilizada para avaliar a qualidade da geração de texto em tarefas de descrição de imagens.

A métrica ROUGE compara o resumo gerado pelo modelo com o resumo verdadeiro em termos de similaridade lexical e cobertura de conteúdo, levando em consideração diferentes componentes, como o conjunto de palavras comuns e similaridade lexical. A pontuação da métrica ROUGE varia de 0 a 1, sendo que quanto maior a pontuação, maior é a similaridade entre o resumo gerado pelo modelo e o resumo verdadeiro.

A métrica ROUGE é amplamente utilizada em tarefas de geração de texto, além da geração de resumos, pois permite comparar a similaridade entre os textos gerados e os textos originais [24]. O cálculo da métrica é baseado na proporção de sequências de palavras geradas pelo modelo em relação ao total de sequências de palavras presentes nas

sentenças originais.

3 Estado da Arte em Geração de Laudos Radiológicos

Neste capítulo, serão apresentados alguns trabalhos concentrados na geração de laudos radiológicos, abordando as arquiteturas utilizadas para a extração de características e para a geração de texto.

Os autores Gasimova et al. [14] desenvolveram uma arquitetura de geração de laudos focada em fraturas de joelhos. Para extrair as características visuais, foi utilizada a rede GoogLeNet [55] em uma arquitetura de *encoder-decoder*, usando camadas até a última camada de *pooling*. Além disso, um CNN foi treinado para criar *bounding boxes* nas fraturas presentes na imagem, visando aprimorar o *encoder*. Cada fratura foi capturada em três ângulos diferentes, e as três visualizações foram passadas pelo *encoder* e agregadas para alimentar o *decoder*, que foi construído com a LSTM [21].

Jing et al. [26] utilizaram uma estrutura hierárquica de LSTMs para a geração automática dos laudos. Essa estrutura consiste em duas LSTMs, onde a primeira é responsável por gerar o tópico a partir do *embedding* representativo e a segunda é encarregada de gerar as palavras e compor o laudo utilizando o tópico gerado pela primeira LSTM.

Os autores propuseram a aplicação de uma camada de co-atenção para melhorar a identificação e associação das características presentes nos exames. Essa camada é responsável por associar as características visuais e semânticas, a fim de melhorar a compreensão do modelo. No *encoder*, foi utilizada a VGG-19 [52], que foi previamente treinada em uma tarefa de classificação multilabel. Além disso, o conteúdo das *tags* presentes nos laudos foi utilizado para melhorar a contextualização do conteúdo textual.

No trabalho de Zhang [61], um processo semelhante ao de Jing [26] foi seguido, em que o *encoder* foi treinado em uma classificação multilabel utilizando a DenseNet e um grafo convolucional. Desse componente, são criados dois fluxos: um para a geração do texto e outro para a classificação. A geração do laudo pelo *decoder* ocorre com um bloco hierárquico de LSTMs para a geração de tópicos e palavras.

No trabalho de Noorallahzadeh et al. [39], que utiliza exames de raios-x de tórax, foi

proposta uma arquitetura que emprega a ResNet [19] e um bloco modificado de transformers como *encoder*. Para o *decoder*, os autores utilizaram o BART [30]. Nessa abordagem, as características extraídas pela ResNet são passadas pelo *transformer* e utilizadas pelo BART para gerar o conteúdo textual do laudo.

Algumas abordagens utilizam o aprendizado por reforço para a geração de laudos. Em [31], é utilizado um modelo de template para padronizar os laudos, sendo implementado um módulo de recompensa para cada palavra gerada pela LSTM, visando adaptar-se melhor às características extraídas das imagens. Já outras abordagens utilizam a VGG-19 como *encoder* e a LSTM como *decoder*. Em [35], também é utilizado o aprendizado por reforço, mas com a concatenação das características extraídas pelo *encoder* (que utiliza a DenseNet) e a saída da última camada de *pooling* como *embedding* do laudo. Isso é feito aplicando a recompensa para cada palavra gerada, melhorando assim a predição da LSTM.

A maioria dos trabalhos adota a arquitetura *encoder-decoder*, que consiste em uma CNN para extrair as características da imagem e uma rede neural para gerar o texto correspondente. Entretanto, existem variações quanto à geração do texto, como a utilização de LSTMs hierárquicas para gerar tópicos e palavras, a utilização de camadas de co-atenção para associar características visuais e semânticas, e a aplicação de aprendizado por reforço para aprimorar a qualidade da geração textual.

O principal desafio desse tipo de trabalho é como enriquecer os dados para aprimorar a qualidade da geração textual. Para isso, é possível identificar ou gerar tópicos presentes no *embedding*, ou utilizar tags predefinidas nos laudos originais para melhor tratar o contexto, como a indicação da patologia presente ou se o exame é normal.

Mecanismos de atenção são aplicados com o propósito de melhorar a geração textual, permitindo que a rede de geração textual decodifique as informações presentes nos *embeddings* com maior qualidade e precisão.

Neste trabalho, não só examinaremos a geração do texto radiológico que descreve o exame de imagem, mas também a explicabilidade do exame em relação às palavras geradas. Isso permitirá que a arquitetura seja utilizada como suporte aos radiologistas durante a escrita e interpretação do laudo. No próximo capítulo, apresentaremos detalhadamente a estrutura da arquitetura *encoder-decoder* que utilizamos, bem como a abordagem adotada para obter a explicabilidade das palavras geradas.

4 Materiais e Métodos

Neste capítulo, apresentamos a metodologia aplicada, incluindo os materiais utilizados e a seleção das amostras utilizadas no desenvolvimento do modelo de inteligência artificial.

4.1 Base de dados

Para o desenvolvimento do modelo de inteligência artificial, utilizamos a base de dados IU-Xray [13], que contém 7470 exames de raio-X e 3955 laudos. A base de dados foi obtida a partir de exames de dois sistemas hospitalares dentro do banco de dados da *Indiana Network for Patient Care*, embora o nome específico dos hospitais não tenha sido mencionado no artigo. Foi selecionado apenas um exame por paciente, sendo que os exames correspondiam a pacientes ambulatoriais. Para garantir a conformidade com a Lei de Portabilidade e Responsabilidade de Seguro de Saúde dos EUA, foram removidas todas as informações que identificassem os pacientes.

As anotações dos exames foram realizadas pelos pesquisadores em duas etapas. Na primeira etapa, os exames foram categorizados como normais ou anormais, considerando a presença ou ausência de achados de doenças. Na segunda etapa, os exames classificados como anormais na fase anterior foram categorizados conforme a alteração específica presente.

Os diagnósticos descritos nos laudos são associados a imagens frontais do tórax e, opcionalmente, imagens laterais. Na Figura 4.1, apresentamos exemplos de imagens do exame de raio X frontal e lateral.

Os laudos são divididos em 3 seções, sendo elas:

- Indicação: Seção que transmite a razão médica pela qual o paciente terá que realizar o exame [43].
- Achados: Contém observações informativas evitando interpretações inadequadas, evitando uso excessivo de termos que remetem a percepção do radiologista e redundâncias [18].

Figura 4.1. Exemplo de imagem do exame raio-X, na esquerda imagem fronta da região torácica e na direita região lateral de um mesmo exame marcado como normal, sem presença de alterações clínicas. Fonte: [13].

- Impressão: É a síntese do significado do achado que permite formalizar os diagnósticos [18].

A base de dados contém 1717 condições, das quais foram selecionadas as cinco com a maior quantidade de exames de imagem e laudos: *lung hypoinflation*, *lung hyperdistention*, *cardiomegaly*, *aorta tortuous* e *spine degenerative*. A Tabela 4.1 apresenta a quantidade de imagens de exames de raio-X utilizadas para cada patologia. As seções de indicações e impressão foram desconsideradas, pois representam a motivação do paciente para realizar o exame e o possível diagnóstico final, respectivamente, que podem ser influenciados por informações externas ao exame [43]. Portanto, consideramos apenas a seção de achados no laudo.

Tabela 4.1. Amostragem por condições utilizadas no processo de validação cruzada

Patologia	**Quantidade de Amostras**
Lung Hypoinflation	245
Lung Hyperdistention	164
Cardiomegaly	157
Aorta tourtuos	126
Spine degenerative	115

As Figuras 4.2, 4.3, 4.4, 4.5, 4.6 apresentam uma representação das características dos textos presentes nos laudos. É possível observar que os laudos alterados normalmente contêm menos texto do que os laudos normais. No entanto, em todos os conjuntos de dados divididos por condições, há uma variabilidade limitada de palavras, sendo que as palavras mais frequentes são comuns entre todas as condições selecionadas.

4.2 Rede de extração de características *Encoder*

Para aplicar a arquitetura proposta de *encoder-decoder*, é necessário treinar uma rede capaz de extrair características da imagem e formar o *encoder*.

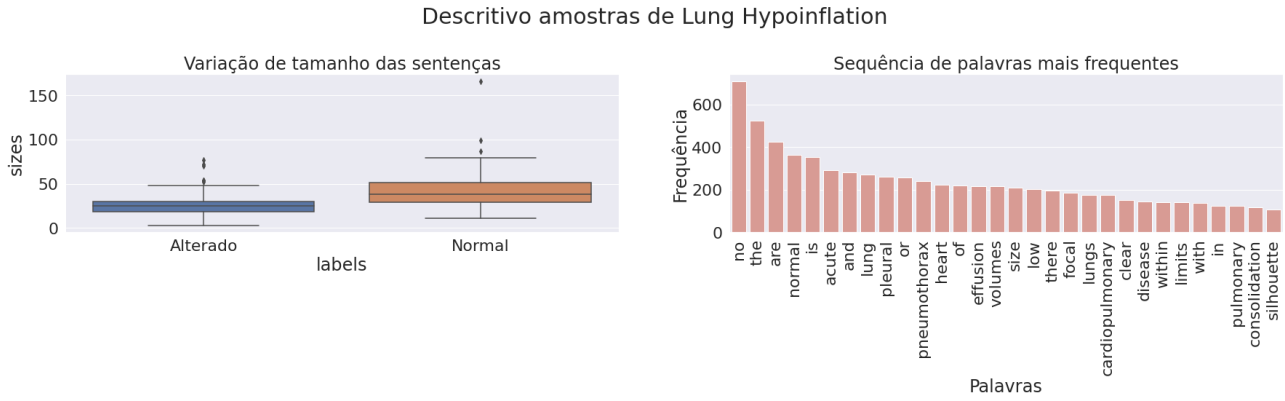

Figura 4.2. Característica dos laudos de *Lung hypoinflation*. Laudos que representam alterações tem são menores em número de palavras em comparação com normais. Fonte: Autoria própria.

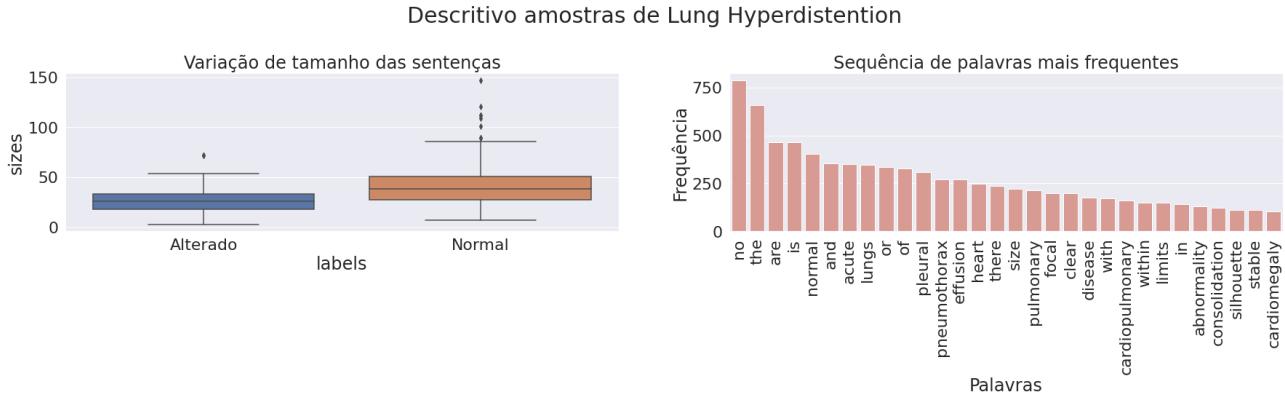

Figura 4.3. Característica dos laudos de *Lung hyperdistantion*. Laudos que representam alterações tem são menores em número de palavras em comparação com normais.

4.2.1 Treinamento do Encoder

Para treinar a classificação de imagens normais e alteradas em cada classe, utilizamos todas as imagens frontais disponíveis dos exames. Devido à escassez de amostras de exames de imagens, aplicamos técnicas de aumento de dados, criando um conjunto expandido de imagens para cada imagem original. Essas novas imagens foram geradas por meio de quatro tipos diferentes de transformações: rotação aleatória, rotação vertical, rotação horizontal e normalização da imagem. A escolha dessas transformações foi baseada em experimentos anteriores, nos quais observamos não apenas as melhores métricas de classificação geradas pela rede de encoder, mas também a qualidade do texto gerado pela rede de decoder. Como resultado desse processo, foram criadas quatro novas imagens para cada imagem original, garantindo que o modelo pudesse classificar exames normais e alterados de forma consistente, independentemente da posição ou qualidade da imagem. Exemplos dessas amostras geradas a partir dos exames de imagem podem ser visualizados nas Figuras 4.7, 4.8, 4.9 e 4.10.

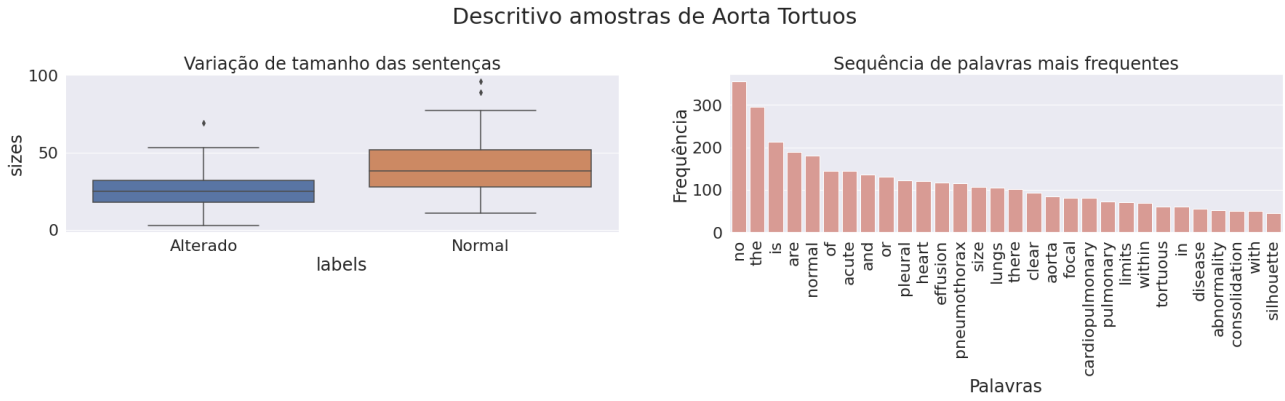

Figura 4.4. Característica dos laudos de *Aorta tortuos*. Laudos que representam alterações tem são menores em número de palavras em comparação com normais.

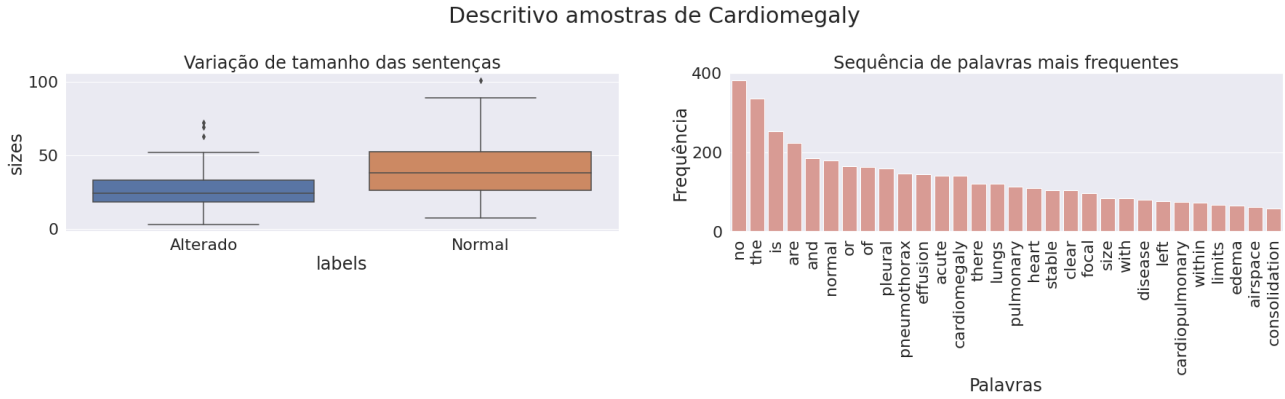

Figura 4.5. Característica dos laudos de *Cardiomegaly*.Laudos que representam alterações tem são menores em número de palavras em comparação com normais.

A arquitetura escolhida para construir o *encoder* foi a densenet, especificamente a versão 121. A densenet é amplamente empregada na extração de características de imagens de raio-X devido à sua característica única de unir todas as camadas convolucionais. Essa união permite que os padrões presentes nas imagens sejam captados de forma abrangente, possibilitando uma tradução mais precisa em texto. Esta variação arquitetural demonstrou resultados significativamente melhores na transformação dessas características em texto, superando outras variações existentes. Além disso, ela foi modificada incluindo a camada *AdaptiveAvgPool* antes da camada linear com a função de ativação, responsável pela classificação das imagens.

Para avaliar a qualidade da rede que será utilizada como *encoder*, foi empregado o processo de validação cruzada. Essa técnica é um método de avaliação de modelos de inteligência artificial que envolve a divisão dos dados em N conjuntos, denominados *folds*, onde cada conjunto é utilizado como dados de teste uma vez. A qualidade do modelo é determinada pela média das métricas de desempenho obtidas em todos os *folds* de treinamento.

Os dados foram divididos em quatro *folds*. A Figura 5.3 apresenta como os con-

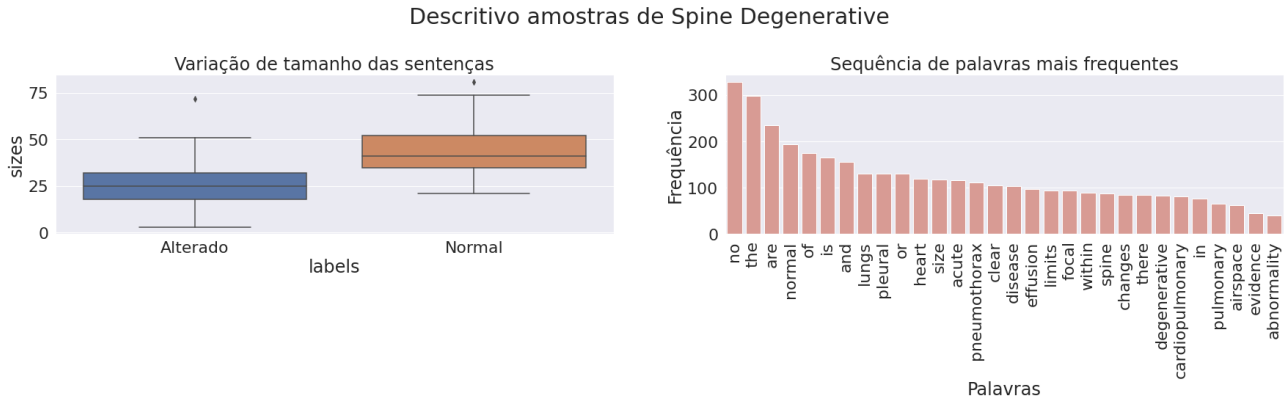

Figura 4.6. Característica dos laudos de *Spine degenerative*. Laudos que representam alterações tem são menores em número de palavras em comparação com normais.

juntos de exames de imagem foram separados para cada uma das condições observadas, mantendo as proporções entre exames normais e alterados nos grupos.

As classificações binárias propostas pela rede neural obtiveram resultados significativos na identificação das classes, distinguindo entre exames normais e exames alterados. As métricas indicam que os modelos são capazes de reconhecer as características das imagens e convertê-las em texto. O objetivo é que as redes sejam capazes de extrair características dos pesos obtidos pela rede e aplicá-las para a extração de características nas imagens de exames. Para isso, a última camada responsável pela classificação da rede é removida, utilizando apenas o espaço latente obtido a partir dos pesos da rede.

4.3 Rede de Geração textual *Decoder*

A arquitetura proposta do *decoder* tem como objetivo gerar descrições textuais explicativas juntamente com a imagem. A camada de *spatial attention* [36] foi adicionada para permitir ao *decoder* focar em regiões mais relevantes da imagem, resultando em descrições mais precisas e informativas. O *decoder* utiliza as características extraídas pelo *encoder*, selecionando aquelas que melhor descrevem as informações presentes na imagem para gerar o texto correspondente.

4.3.1 Treinamento do Decoder

Durante o treinamento do *decoder*, é realizada a associação de cada palavra a um valor numérico, possibilitando a representação do conteúdo textual por uma sequência de números. Essa sequência representa todo o vocabulário existente, incluindo todas as palavras e pontuações presentes nos laudos.

Figura 4.7. Exemplo de raio-X levemente rotacionado após o processo de *augmentation*, à esquerda na imagem original da base dados, na direita a imagem inclinada após *augmentation*.

Figura 4.8. Exemplo de raio-X verticalmente rotacionada após o processo de *augmentation*, à esquerda na imagem original da base dados, na direita a imagem verticalmente rotacionada após *augmentation*.

Figura 4.9. Exemplo de raio-X horizontalmente rotacionada após o processo de *augmentation*, à esquerda na imagem original da base dados, na direita a imagem horizontalmente rotacionada após *augmentation*.

Figura 4.10. Exemplo de raio-X histogramas normalizados após o processo de *augmentation*, à esquerda na imagem original da base dados, na direita a imagem com os histogramas normalizados rotacionada após *augmentation*.

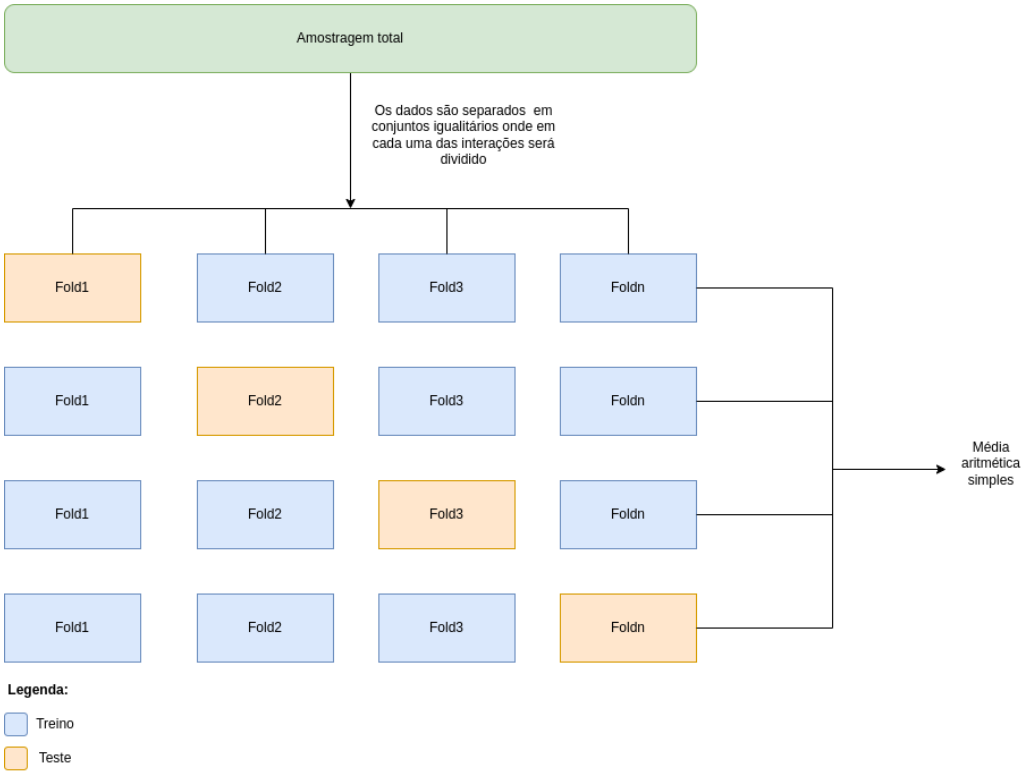

Figura 4.11. Divisão dos dados utilizados para treinamento em quatro folds. Os dados de treinamento foram divididos em quatro grupos (folds). Os grupos em azul representam os conjuntos utilizados para o treinamento, enquanto os conjuntos em amarelo contêm as imagens de teste. Essa divisão permite avaliar o modelo em diferentes cenários.

Os laudos presentes no dataset [13] passaram por um processo de anonimização, no qual informações pessoais, como nome, idade e sexo dos pacientes, foram substituídas por sequências fixas de caracteres. Essas sequências, juntamente com espaços em branco e vazios excedentes, foram removidas dos laudos. A representação numérica dos laudos é aplicada a camada de embedding do decoder.

A fim de utilizar as redes de classificação binárias dos exames em relação a normalidade e alterações como *encoder*, foram removidas as duas últimas camadas: a camada de *pooling* e a camada linear responsável por gerar as classificações. Dessa forma, todas as camadas convolucionais que aprenderam as características das imagens que diferenciam exames normais de exames alterados foram mantidas, e as informações obtidas por essas camadas convolucionais são denominadas espaço latente.

A representação textual do laudo é concatenada com o espaço latente obtido pelo *encoder*, resultando em uma representação que inclui tanto o texto quanto as características relevantes dos exames. Esse vetor de representação, obtido através da concatenação do *embedding* textual e do espaço latente da imagem, é então submetido a um bloco de atenção da arquitetura *transformers* [57]. Essa etapa adiciona informações de contexto ao vetor de representação, permitindo capturar relações semanticamente fortes presentes

no conteúdo.

Após a aplicação do bloco de atenção dos *transformers* [57], é aplicado o bloco de *spatial attention* [36] em seguida. Com o *spatial attention*, obtemos os valores de alpha, que representam as regiões de destaque nos exames de imagem. Os pesos de atenção obtidos são concatenados com os valores de alpha, que são utilizados como entrada para a LSTM, gerando assim a palavra que relaciona as regiões de atenção da imagem. A Figura 4.12 demonstra o fluxo da arquitetura da *encoder-decoder* proposta.

O processo descrito acima foi aplicado em cada uma 5 condições utilizadas neste trabalho mantendo os mesmos *folds* utilizados no treinamento das redes de *encoder* para evitar erros na avaliação ondem o *encoder* pudesse apresentar um falso desempenho em extrair características devido as imagem anteriormente estarem no conjunto de treino.

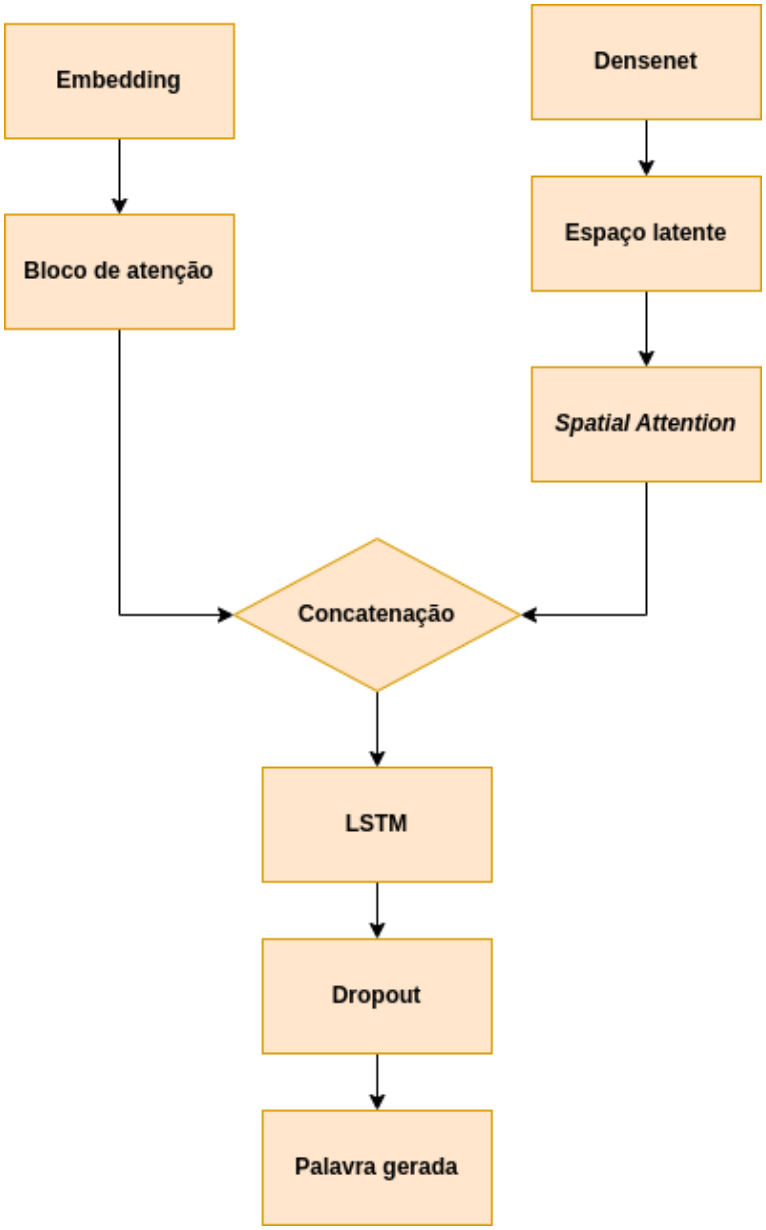

Figura 4.12. Demonstração do fluxo do modelo *encoder-decoder proposto*. A *densenet* extrai as características das imagens gerando produzindo o espaço latente, nele é aplicado o *spatial attention*, já no *embedding* dos laudos é aplicado um bloco de atenção dos *transformers*, essas informações são concatenadas passando alimentando a LSTM e aplicando uma camada de dropout, ao final gerando uma nova palavra.

5 Resultados e Discussões

Este capítulo apresenta os resultados obtidos, incluindo os treinamentos das redes de *encoder* para classificação binária e os resultados da geração de texto.

5.1 Resultados Rede de *Encoder*

Nas Figuras 5.4, 5.5, 5.6, 5.7 e 5.8 apresentam as evoluções do treinamento das redes *encoder* em relação às épocas nos 4 folds. O círculo vermelho indica a época do presente *fold* que obteve os melhores resultados das métricas f1score e AUC, dentre as 100 épocas de treinamento.

Tabela 5.1. Média das métricas dos 4 *folds* para cada uma das condições.

Modelo	F1score	Auc
Lung Hypoinflamation	0.76	0.76
Lung Hyperdistention	0.76	0.72
Cardiomegaly	0.80	0.81
Aorta tortuos	0.59	0.63
Spine degenerative	0.63	0.58

Na Tabela 5.1, apresentam-se os resultados da média dos 4 *folds* após o treinamento de 100 épocas. Todas as redes foram treinadas com a mesma arquitetura e parâmetros, variando apenas os dados das condições de treinamento. Os *encoders* para *hypoinflation*, *hyperdistention* e *cardiomegaly* obtiveram resultados expressivos, sendo capazes de distinguir entre exames normais e alterados e permitindo a aprendizagem de características que alimentam o *decoder*.

5.2 Resultados da Rede de *Decoder*

Para gerar os laudos com as marcações nas imagens, foram utilizadas como *encoders* as redes que obtiveram as melhores métricas de AUC e F1-score em suas respectivas épocas, além da rede treinada durante as 100 épocas completas. Os resultados da métrica ROUGE para geração textual são apresentados na Tabela 5.2, em que o nome do modelo

indica a patologia, o fold e a época em que se obteve a melhor métrica.

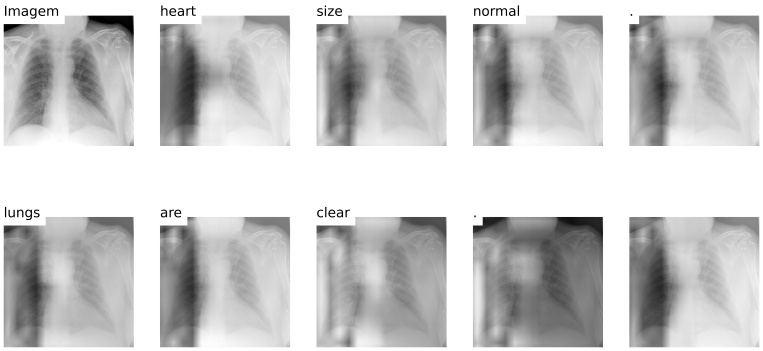

Figura 5.1. Exemplo de geração de texto para um laudo médico, utilizando o sistema que foi desenvolvido com base em um exame de raios-X original. Durante o percurso das 10 imagens, é possível identificar regiões de destaque. O sistema automaticamente gera uma sequência de palavras que corresponde a cada uma dessas áreas destacadas, proporcionando uma descrição detalhada do exame.

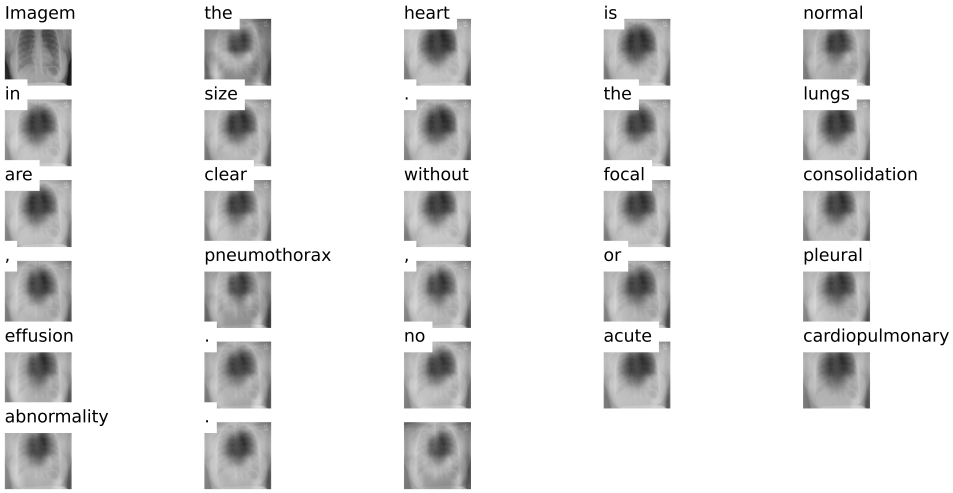

Figura 5.2. Exemplo de geração de texto para um laudo médico, utilizando o sistema desenvolvido, a partir de um exame de raios-X (imagem original mostrada). O sistema gera uma sequência de palavras correspondentes a cada região automaticamente destacada nas 27 imagens subsequentes à imagem original. É importante notar que, para cada região destacada, o sistema gera uma palavra na sequência textual do laudo, permitindo ao profissional de saúde não apenas observar o resultado diagnóstico, mas também identificar as regiões da imagem que embasam esse resultado.

As Figuras 5.1, 5.2 e 5.3 apresentam alguns resultados das marcações geradas juntamente com as palavras correspondentes. Em cada figura, a primeira imagem exibe o exame frontal original. As regiões mais claras indicam as áreas de atenção produzidas pela camada de *spatial attention*, as quais se relacionam com a palavra gerada e são adicionadas ao exame. As imagens foram equalizadas para proporcionar uma visualização mais nítida das regiões de atenção geradas pelo modelo. No 6, é possível encontrar mais exemplos de marcações feitas pelo modelo, desta vez concentradas em palavras cruciais para um diagnóstico ou identificação de doenças.

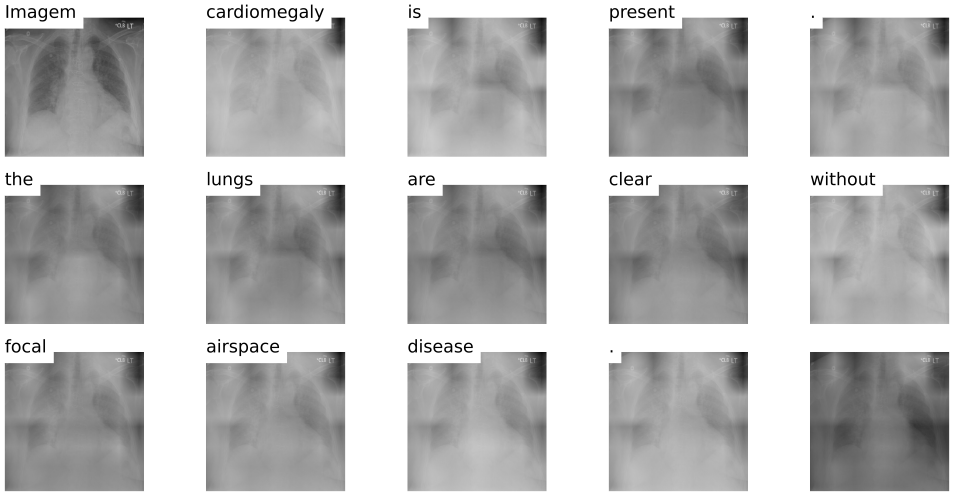

Figura 5.3. Exemplo de geração de texto para um laudo médico, com base em um exame de raios-X original, utilizando o sistema desenvolvido. Contudo, é crucial notar que as áreas destinadas à descrição das palavras não apresentam definição nítida, resultando em áreas com visualização opaca e esfumaçada. Isso impede a identificação de regiões de interpretabilidade específica, abrangendo a imagem como um todo, ao longo de um conjunto de 10 imagens.

Para cada uma das Figuras a seguir (Figuras 5.1, 5.2 e 5.3), são apresentados os conteúdos dos laudos originais:

- Figura 5.1: *Heart upper limits normal. Lungs clear*
- Figura 5.2: *There are midline sternotomy and mediastinal clips consistent with prior CABG. The heart is enlarged with unfolding of the aorta. There is prominence of the interstitial markings with fluid in the fissures consistent with interstitial edema. There is no focal airspace opacity, large pleural effusion, or pneumothorax. There multilevel degenerative spine changes. Interstitial pulmonary edema. Cardiomegaly.*
- Figura 5.3: *Heart size normal and lungs are clear. No edema or pneumonia. No effusion*

Ao comparar os conteúdos originais com os gerados, é possível observar que a rede *encoder-decoder* apresentou um comportamento mais orientado ao contexto do que às palavras originais. As relações entre as palavras e os exames mostram que a rede considerou a imagem como um todo, havendo pequenas variações entre palavras mais descritivas e palavras mais genéricas, como artigos e preposições.

Na primeira predição da Figura 5.1, o modelo se aproxima da sentença original. No entanto, a palavra "limits", que não é uma das mais frequentes, não foi inferida pelo modelo, o que indica que a baixa variabilidade das palavras presentes no dataset influencia na qualidade do texto gerado.

Na Figura 5.2, o modelo se aproxima do contexto na metade da sentença, porém deixa muitas informações pendentes. As palavras mais frequentes presentes nos laudos foram previstas, mas aquelas de menor frequência não foram previstas pelo modelo.

A baixa variabilidade das palavras presentes nos laudos dificulta a geração de palavras com menor frequência, resultando em uma geração textual previsível baseada em palavras mais frequentes, como visto na Figura 5.3. Embora o modelo tenha se aproximado do contexto com base nas palavras geradas, ele não conseguiu acertar exatamente todas as palavras. Na seção 6, são apresentados alguns exemplos focalizados na geração do laudo em comparação com o texto original. As marcações de atenção não são exibidas, apenas o texto gerado em relação à imagem correspondente.

Por fim, analisamos o desempenho das redes de codificação em relação à extração de características para a geração de laudos em uma determinada condição. Utilizamos a rede de *encoder* de *hypoinflation* e selecionamos aleatoriamente uma imagem de cada uma das diferentes condições para observar como seriam os laudos gerados. A seguir, apresentamos alguns exemplos de laudos:

- *No focal airspace disease* para *hyperdistention.*
- *Bilateral* para *spine generative.*
- The heart size and cardiomediastinal silhouette are normal. There is no focal airspa*ce opacity, pleural effusion or pneumothorax. The osseous structures are intact. No acute cardiopulmonary finding* para *aorta tortuos.*
- Heart size normal para *cardiomegalia.*

E as redes de *encoder* para as demais condições resultaram em laudos nulos ou com a presença do token *unk*, que, no conjunto de dados de laudos utilizados, foi definido para representar palavras ou codificações desconhecidas.

Esses exemplos destacam que, ao utilizar a rede de *encoder* treinada em *hypoinflation* para outras condições, os laudos gerados apresentaram conteúdo normal e, em alguns casos, conteúdo semântico ausente. O codificador especializado em condições patológicas demonstra limitações ao extrair características de outras condições e traduzi-las em texto. Isso ocorre porque o vocabulário e as informações mais relevantes presentes na imagem diferem entre as redes responsáveis pela extração de características.s na imagem diferem entre as redes responsáveis pela extração de características.

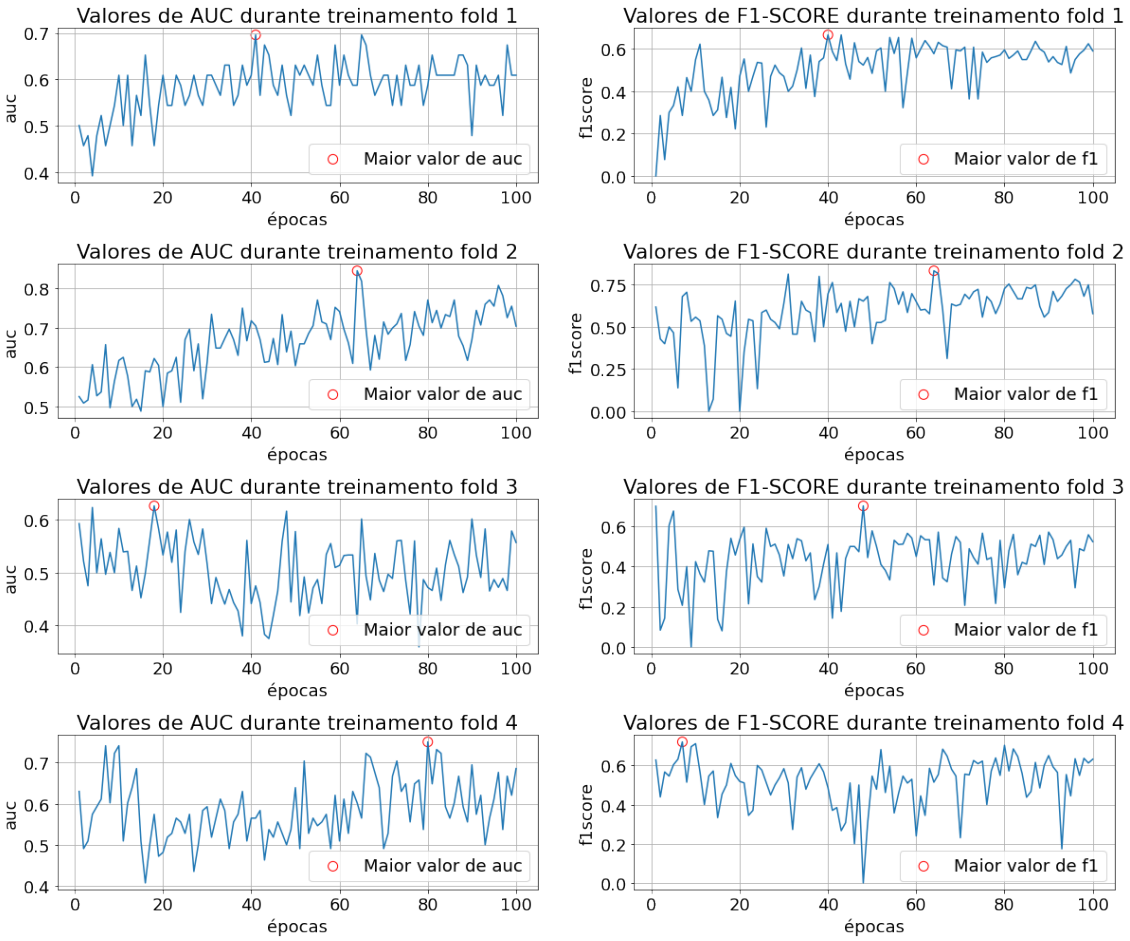

Figura 5.4. Resultados dos folds de treinamento para Aorta tortuos. Avaliação do modelo ao longo de 100 épocas, com foco na análise do desempenho da rede encoder em relação às métricas AUC e F1-Score. Durante esse período de treinamento, acompanhamos de perto o progresso do modelo e observamos seu melhor desempenho, bem como seu comportamento até a conclusão do ciclo completo de treinamento para cada um dos 4 folds.

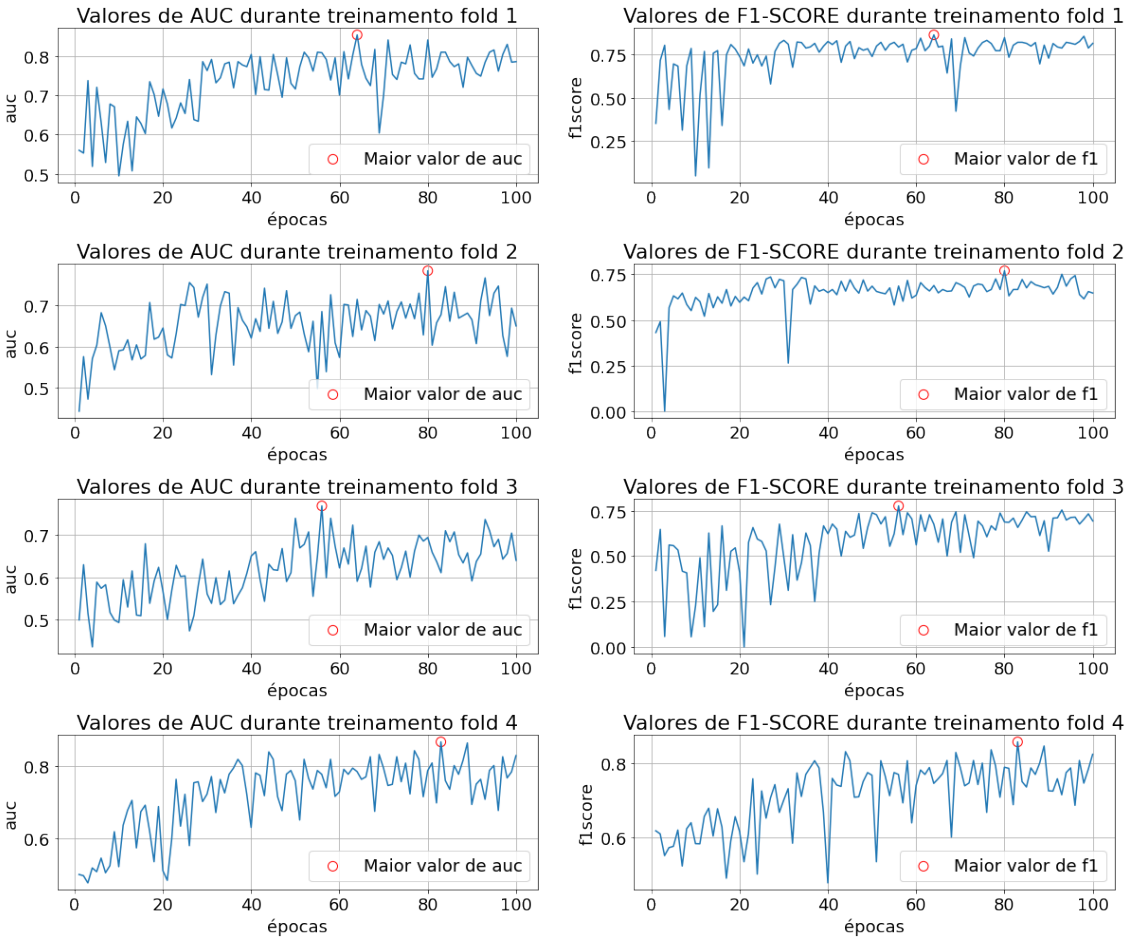

Figura 5.5. Resultados dos folds de treinamento para *lung hyperdistation*. Avaliação do modelo ao longo de 100 épocas, com foco na análise do desempenho da rede encoder em relação às métricas AUC e F1-Score. Durante esse período de treinamento, acompanhamos de perto o progresso do modelo e observamos seu melhor desempenho, bem como seu comportamento até a conclusão do ciclo completo de treinamento para cada um dos 4 folds.

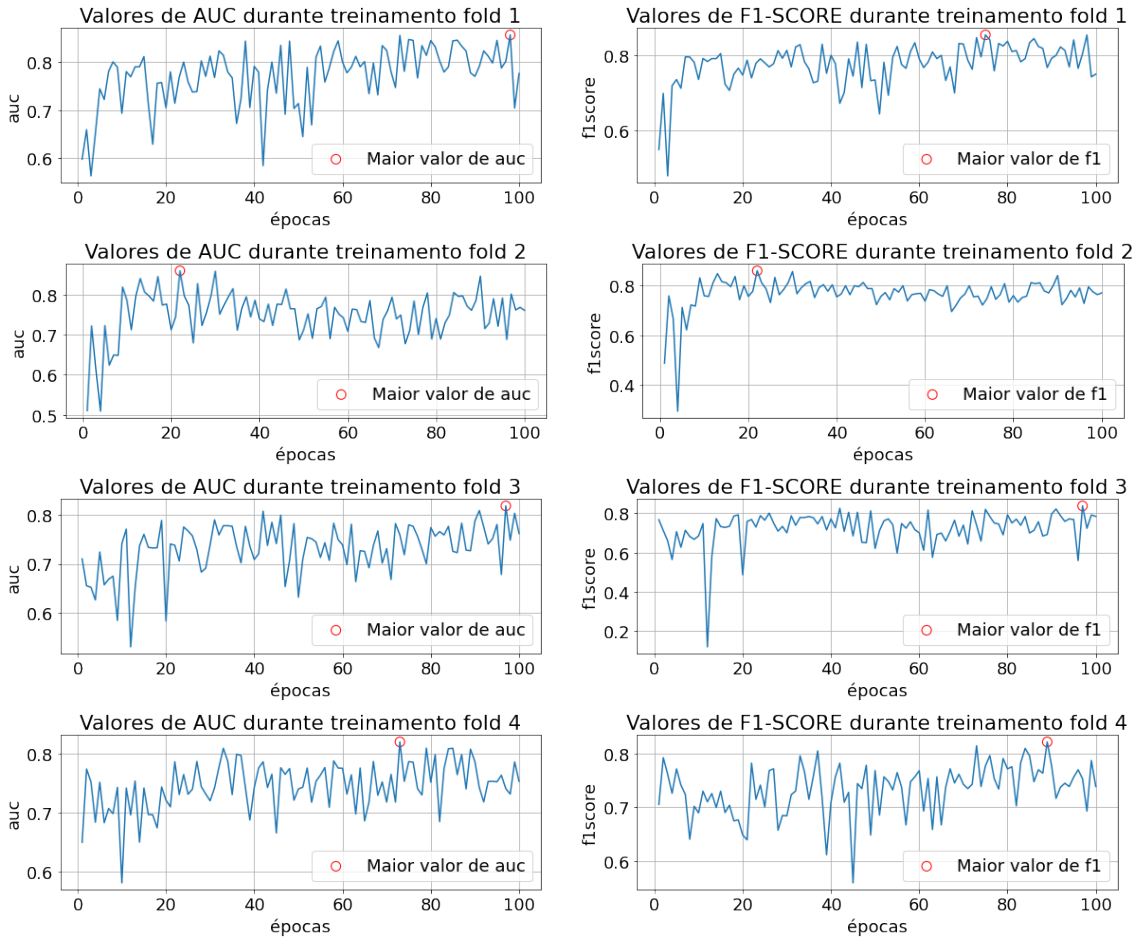

Figura 5.6. Resultados dos folds de treinamento para *lung hypoinflamation*. Avaliação do modelo ao longo de 100 épocas, com foco na análise do desempenho da rede encoder em relação às métricas AUC e F1-Score. Durante esse período de treinamento, acompanhamos de perto o progresso do modelo e observamos seu melhor desempenho, bem como seu comportamento até a conclusão do ciclo completo de treinamento para cada um dos 4 folds.

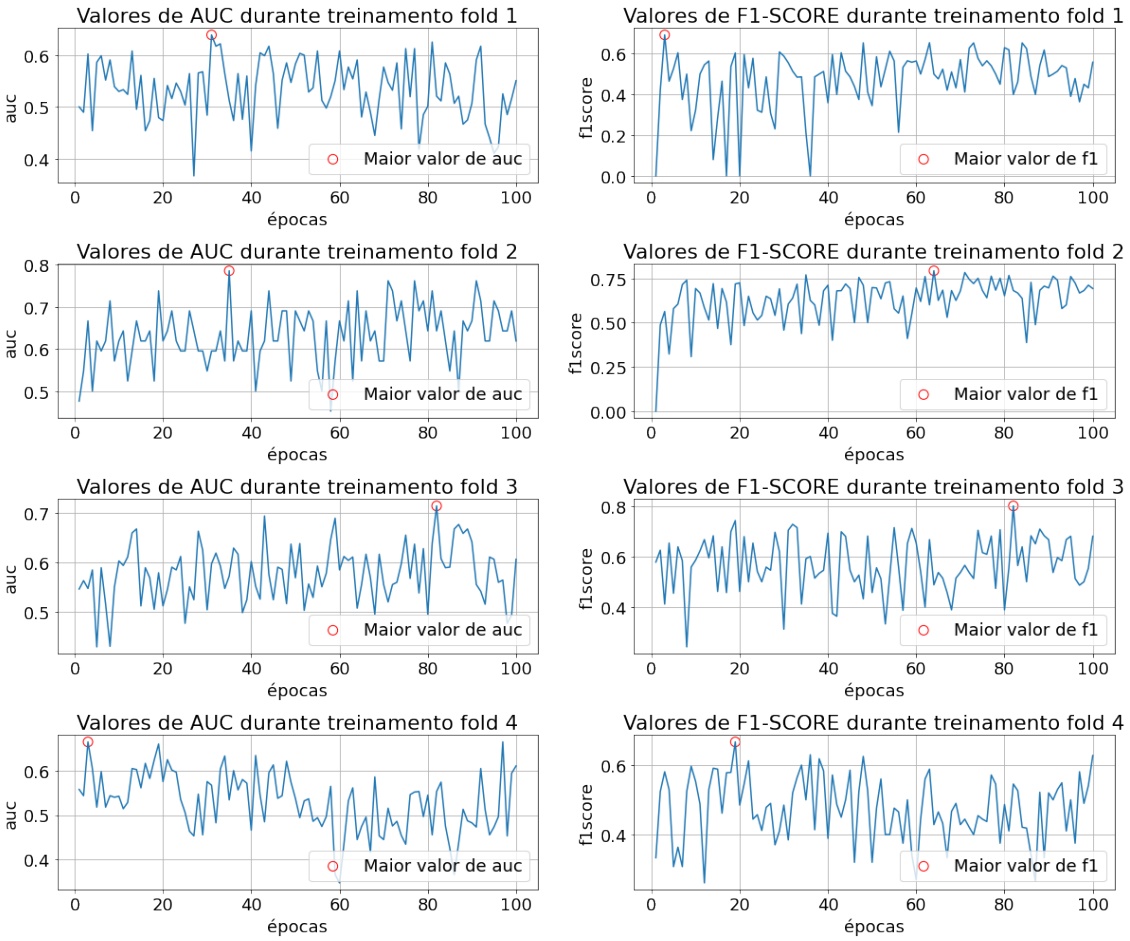

Figura 5.7. Resultados dos folds de treinamento para *spine degenerative*. Avaliação do modelo ao longo de 100 épocas, com foco na análise do desempenho da rede encoder em relação às métricas AUC e F1-Score. Durante esse período de treinamento, acompanhamos de perto o progresso do modelo e observamos seu melhor desempenho, bem como seu comportamento até a conclusão do ciclo completo de treinamento para cada um dos 4 folds.

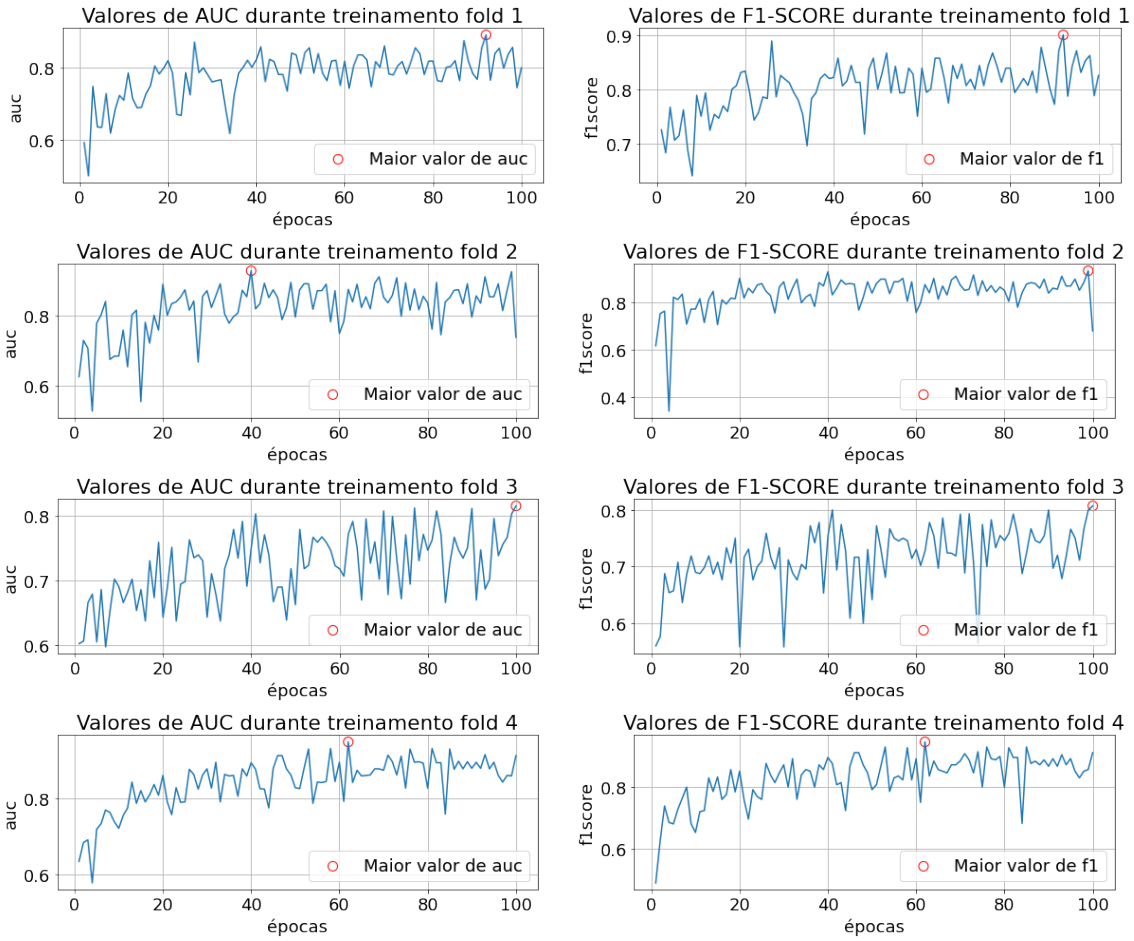

Figura 5.8. Resultados dos folds de treinamento para *cardiomegaly*. Avaliação do modelo ao longo de 100 épocas, com foco na análise do desempenho da rede encoder em relação às métricas AUC e F1-Score. Durante esse período de treinamento, acompanhamos de perto o progresso do modelo e observamos seu melhor desempenho, bem como seu comportamento até a conclusão do ciclo completo de treinamento para cada um dos 4 folds.

Tabela 5.2. Média das métricas dos 4 *folds* para cada uma das condições.

Modelo	ROUGE
lung/Hypoinflamation_fold1_epoca75_f1	0.33
lung/Hypoinflamation_fold1_epoca92_auc	0.35
lung/Hypoinflamation_fold2_epoca22_f1	0.34
lung/Hypoinflamation_fold2_epoca22_auc	0.34
lung/Hypoinflamation_fold3_epoca97_f1	0.32
lung/Hypoinflamation_fold3_epoca97_auc	0.32
lung/Hypoinflamation_fold4_epoca89_f1	0.32
lung/Hypoinflamation_fold4_epoca73_auc	0.34
lung/Hyperdistention_fold1_epoca64_f1	0.31
lung/Hyperdistention_fold1_epoca64_auc	0.31
lung/Hyperdistention_fold2_epoca80_f1	0.32
lung/Hyperdistention_fold2_epoca80_auc	0.32
lung/Hyperdistention_fold3_epoca56_f1	0.43
lung/Hyperdistention_fold3_epoca56_auc	0.43
lung/Hyperdistention_fold4_epoca83_f1	0.37
lung/Hyperdistention_fold4_epoca83_auc	0.37
cardiomegaly_fold1_epoca92_f1	0.30
cardiomegaly_fold1_epoca92_auc	0.30
cardiomegaly_fold2_epoca99_f1	0.28
cardiomegaly_fold2_epoca40_auc	0.26
cardiomegaly_fold3_epoca100_f1	0.30
cardiomegaly_fold3_epoca100_auc	0.30
cardiomegaly_fold4_epoca62_f1	0.31
cardiomegaly_fold4_epoca62_auc	0.31
spine_fold1_epoca3_f1	0.28
spine_fold1_epoca31_auc	0.30
spine_fold2_epoca64_f1	0.28
spine_fold2_epoca35_f1	0.31
spine_fold3_epoca82_auc	0.32
spine_fold3_epoca82_f1	0.33
spine_fold4_epoca19_auc	0.26
spine_fold4_epoca3_f1	0.30
aorta_fold1_epoca40_f1	0.35
aorta_fold1_epoca41_auc	0.37
aorta_fold2_epoca64_f1	0.32
aorta_fold2_epoca64_auc	0.32
aorta_fold3_epoca48_f1	0.31
aorta_fold3_epoca18_auc	0.34
aorta_fold4_epoca7_f1	0.34
aorta_fold4_epoca80_auc	0.35

6 Conclusão

A explicabilidade é uma característica crucial a ser considerada na construção de modelos de geração textual, especialmente quando aplicados a laudos radiológicos. Essa abordagem pode ser uma aliada valiosa para aprimorar a eficiência do processo de interpretação de imagens médicas, oferecendo suporte aos médicos na tomada de decisões clínicas e evitando possíveis erros que poderiam impactar a saúde dos pacientes. O modelo apresentado neste estudo é baseado em uma arquitetura encoder-decoder, o que requer a extração eficiente de características a partir de exames de imagem para um funcionamento adequado.

No entanto, é importante ressaltar uma limitação significativa deste sistema: ele foi treinado especificamente para abordar apenas cinco condições médicas específicas. Essa limitação significa que o modelo só é capaz de gerar laudos radiológicos dentro do escopo das cinco patologias para as quais foi treinado, e qualquer outra condição médica está fora de seu domínio de conhecimento.

Cinco redes distintas foram treinadas para abordar essas condições, com um encoder baseado na arquitetura DenseNet e um decoder que incorpora elementos de transformers e LSTM. Os textos radiológicos gerados foram enriquecidos com a técnica de atenção espacial, permitindo a implementação de uma função de explicabilidade que destaca as regiões da imagem que influenciaram a geração do conteúdo textual.

A análise dos resultados revelou que o vocabulário utilizado nas seções geradas dos laudos é relativamente limitado, com uma tendência a priorizar palavras mais frequentes na produção do texto. Como resultado, a métrica ROUGE, usada para avaliar a qualidade dos textos gerados, apresentou um desempenho consistente em torno de 0,32 para todas as condições avaliadas.

É importante reconhecer a complexidade da estrutura de desenvolvimento do modelo encoder-decoder, que envolve o treinamento de duas redes distintas para etapas diferentes do processo: uma para aprender as características dos exames de imagem e outra para traduzi-las em texto e marcações. No entanto, apesar dos desafios, foi possível alcançar com sucesso a geração textual e a explicabilidade desejadas. Para uma avaliação mais precisa e geral da arquitetura proposta, seria necessário um conjunto de dados mais amplo,

abrangendo uma variedade maior de raios-X e seus respectivos laudos, a fim de alcançar uma melhor generalização. Como trabalhos futuros, são propostas algumas etapas, que podem potencialmente melhorar o desempenho do sistema proposto. Dentre as etapas, destacam-se:

1. Aplicar técnicas de *augmentation* de dados nos textos dos laudos para obter maior variabilidade textual.
2. Utilizar modelos de *visual transformer* para a rede de codificação.
3. Utilizar arquiteturas de geração textual gpt, t5 e llama permitindo realizar marcações nas imagens.
4. Ampliar a aplicabilidade do sistema para incluir laudos gerados a partir de outros tipos de imagens, como imagens de tomografia computadorizada, ressonância magnética, entre outros.

Lista de Referências

[1] A. Al-Sabaawi, H. M. Ibrahim, Z. M. Arkah, M. AlAmidie, e L. Alzubaidi. Amended convolutional neural network with global average pooling for image classification. 2021.

[2] S. ALBAWI, T. A. MOHAMMED, e S. AL-ZAWI. Understanding of a convolutional neural network. 2017.

[3] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, L. Farhan, e M. Al-Amidie. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. 2021.

[4] R. Balestriero e R. G. Baraniuk. Batch normalization explained. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[5] Leonard Berlin. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. 2017.

[6] A. Brady, R. Ó. Laoide, P. McCarthy, e R. McDermott. Discrepancy and error in radiology: Concepts, causes and consequences. 2012.

[7] A. P. Brady. Error and discrepancy in radiology: inevitable or avoidable? 2016.

[8] Adrian Brady, Risteárd Ó Laoide, Peter McCarthy, e Ronan McDermott. Discrepancy and error in radiology: Concepts, causes and consequences. 2011.

[9] Michael A Bruno, Eric A Walker, e Hani H Abujudeh. Understanding and confronting our mistakes: The epidemiology of error in radiology and strategies for error reduction. 2015.

[10] Andrew J. Degnan, Emily H. Ghobadi, Hardy Peter., Elizabeth Krupinski, Elena P. Scali, Lindsay Stratchko, Adam Ulano, Eric Walker, Ashish P Wasnik, e William F Auffermann. Perceptual and interpretive error in diagnostic radiology—causes and potential solutions. 2018.

[11] Trafton Drew, Melissa L. H. Vo, e Jeremy M. Wolfe. The invisible gorilla strikes again: Sustained inattentional blindness in expert observers. 2013.

[12] J. Elliott e K. Williamson. The radiology impact of healthcare errors during shift work. 2019.

[13] D. D. Fushman, Kohli M. D., Rosenman M. B., e Shooshan S. E. Preparing a collection of radiology examinations for distribution and retrieval. 2015.

[14] A. Gasimova, G. Montana, e D. Rueckert. Automated knee x-ray report generation. 2021.

[15] Barbara C. Goo, Lawrence A. Cooperstein, Georgine B. DeMarino, Linda M. Miketi, Rose C. GennarF, Howard E. Rockette, e David Gur. Does knowledge of the clinical history affect the accuracy of chest radiograph interpretation? 1990.

[16] T. N. Hanna, M. E. Zygmont, R. Peterson, D. Theriot, H. Shekhani, J.-O. Johnson, e E. A Krupinski. The effects of fatigue from overnight shifts on radiology search patterns and diagnostic performance. 2017.

[17] Tarek N. Hanna, Thomas Loehfelm, Faisal Khosa, Saurabh Rohatgi, e Jamlik-Omari Johnson. Overnight shift work: factors contributing to diagnostic discrepancies. 2015.

[18] I. C. Gaillard F. Jeffrey P. K. Hartung, M. P. Bickle. How to create a great radiology report. 2020.

[19] Kaiming He, X Zhang, S. Ren, e J. Sun. Deep residual learning for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[20] G. S. Heriot, P. McKelvie, e A. G. Pitman. Diagnostic errors in patients dying in hospital: Radiology's contribution. 2009.

[21] S. Hochreiter e J. Schmidhuber. Long short-term memory. 2017.

[22] Gao Huang, Zhuang Liu, Geoff Pleiss, Laurens Van Der Maaten, e Kilian Weinberger. Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[23] S Ioffe e C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[24] Qureshi S. Iqbal, T. The survey: Text generation models in deep learning. 2019.

[25] T. Iqbal e S. Qureshi. The survey: Text generation models in deep learning. 2020.

[26] B Jing, P. Xie, e E. P. Xing. On the automatic generation of medical imaging reports. 2018.

[27] C. D. Johnson, K. N. Krecke, C. C. Miranda, R. Roberts, e C. Denham. Quality initiatives developing a radiology quality and safety program: A primer. 2009.

[28] Thomas M. J. W. Mandel C. J. Grimm J. Hannaford N. Schultz T. J. Runciman W. Jones, D. N. Where failures occur in the imaging care cycle: Lessons from the radiology events register. 2010.

[29] Y. Lecun, Y. Bengio, e G. Hinton. Deep learning. 2015.

[30] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V Stoyanov, e L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[31] C. Y. Li, X. Liang, Z. Hu, e Xing E. P. Hybrid retrieval-generation reinforced agent for medical image report generation. 2021.

[32] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, e G. Hilton. Backpropagation and the brain. 2020.

[33] C. Y. Lin. Rouge: A package for automatic evaluation of summaries. 2004.

[34] Z. Lin, M. Feng, C. N. Santos, M Yu, B. Xiang, B. Zhou, e Y Bengio. A structured self-attentive sentence embedding. 2017.

[35] G. Liu, T. M. H. Hsu, M. McDermott, W. Boag, P. Szolovits, e M. Ghassemi. Clinically accurate chest x-ray report generation. 2018.

[36] Jiasen Lu. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. 2017.

[37] M. A. Makary e Daniel M. Medical error—the third leading cause of death in the us. 2016.

[38] PITTS W. MCCULLOCH, W. S. A logical calculus of the ideas immanent in nervous activity. butt. math. biophysics. 1943.

[39] F. Noorallahzadeh, N. P. Gonzalez, Frauenfelder T., e Fujimoto K. Progressive transformer-based generation of radiology reports. 2021.

[40] Lindsay P. Busby, Jesse L. Courtier, e Christine M. Glastonbury. Bias in radiology: The how and why of misses and misinterpretations. 2017.

[41] Razvan Pascanu, Tomas Mikolov, e Yoshua Bengio. On the difficulty of training recurrent neural networks. 2013.

[42] Anika G. Patel, Victor J. Pizzitola, C. Daniel Johnson, Nan Zhang, e Maitray D. Patel. Radiologists make more errors interpreting off-hours body ct studies during overnight assignments as compared with daytime assignments. 2020.

[43] J. Pavlopoulos, V. Kougia, I. Androutsopoulos, e D. Papamichail. Diagnostic captioning: a survey. 2022.

[44] M. C. POPESCU, L. P. POPESCU, V. E. BALAS, e N. MASTORAKIS. Multilayer perceptron and neural networks. 2009.

[45] M. Porsani e T. J. Ulrych. Discrete convolution by means of forward and backward modeling. 2020.

[46] F. ROSENBLATT. The perceptron: A probabilistic model for information storage and organization in the brain. 1958.

[47] D. E. Rumelhart, G. E. Hintont, e R. J. Williams. Learning representations by back-propagating errors. 1986.

[48] S. Santhanam. Context based text-generation using lstm networks. 2020.

[49] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview. 2019.

[50] A. Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. 2020.

[51] Ben. H. Niv Shocher, A. F e I. Michal. From discrete to continuous convolution layers. 2020.

[52] K. Simonyan e A. Zisserman. Very deep convolutional networks for large-scale image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[53] Hinton G. Krizhevsky A. Sutskever I. Salakhutdinov R. Srivastava, N. Dropout: A simple way to prevent neural networks from overfitting. 2014.

[54] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, e R. Fergus. Training convolutional networks with noisy labels. 2014.

[55] C. Szegedy, W. Liu, Y. Jia, Pierre S, S. Reed, A. Dragomir, D. Erhan, V. Vanhoucke, e A. Rabinovich. Going deeper with convolutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[56] E. Tjoa e C. Guan. A survey on explainable artificial intelligence (xai): towards medical xai. 2019.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, e L. Gomez, A. N. Kaiser. Attention is all you need. 2017.

[58] C. Wang, P. Nulty, e L. Lillis. A comparative study on word embeddings in deep learning for text classification. 2020.

[59] R. Yamashita, M. Nishio, e K. Richard Kinh Gogashi. Convolutional neural networks: an overview and application in radiology. 2018.

[60] Liem T. Mansfield Young W. Kim and. Fool me twice: Delayed diagnoses in radiology with emphasis on perpetuated errors. 2013.

[61] Y Zhang, X Wang, Z Xu, A Yu, Q Yuille, e D Xu. When radiology report generation meets knowledge graph. 2020, confec = Proceedings of the 56th Annual Meeting of the Association for Computational, doi =10.18653/v1/2021.findings-emnlp.241.

Anexo A

Exemplos de texto de laudos gerados

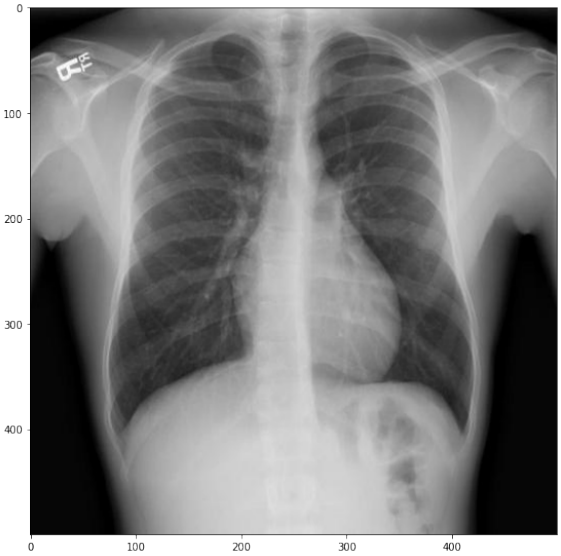

Figura A.1. Exemplo de imagem frontal de exame de raio-X disponível na base de dados, sem marcações de atenção destacadas.

- Laudo gerado pelo sistema: *heart cardiomediastinal silhouette is normal in clear. there lungs are clear. of the lungs lungs are clear. there lungs are clear unremarkable.no focal consolidation is unremarkable effusion no acute cardiopulmonary abnormality.*
- Laudo original: *The lungs appear clear. The heart and pulmonary are normal. Pleural spaces are clear. Mediastinal contours are normal. Patient status post sternotomy and CABG.No acute cardiopulmonary disease*

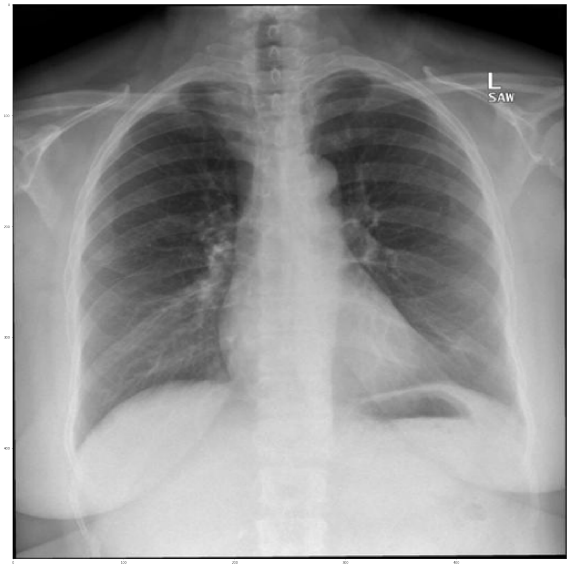

Figura A.2. Exemplo de imagem frontal de exame de raio-X disponível na base de dados, sem marcações de atenção destacadas.

- Laudo gerado pelo sistema: *the heart cardiomediastinal consolidation is acute limits no acute cardiopulmonary abnormality .*
- Laudo original: *The heart is normal in size. The mediastinum is unremarkable. The lungs are clear.No acute disease.*

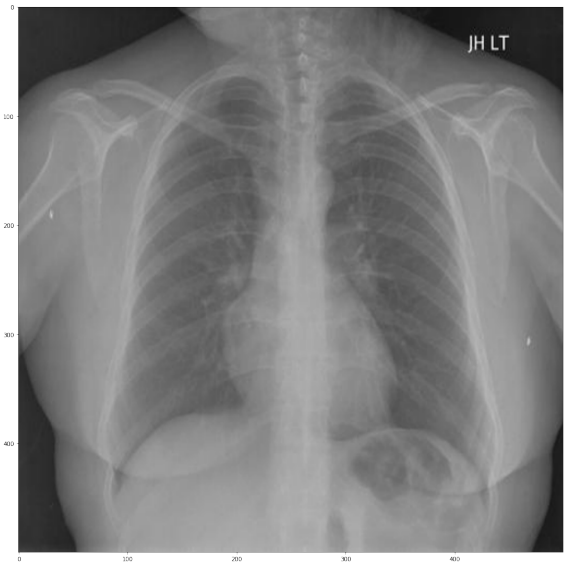

Figura A.3. Exemplo de imagem frontal de exame de raio-X disponível na base de dados, sem marcações de atenção destacadas.

- Laudo gerado pelo sistema: *the heart size within pneumothorax no acute limits effusion no acute cardiopulmonary abnormality no pneumothorax no acute cardiopulmonary abnormality .*
- Laudo original: *Heart size within normal limits. Negative for focal pulmonary consolidation, pleural effusion, or pneumothorax. No upper lobe airspace disease or cavitary lesions identified.. No acute abnormality. . No evidence of pulmonary tuberculosis.*

Anexo B

Exemplos de marcações e palavras geradas

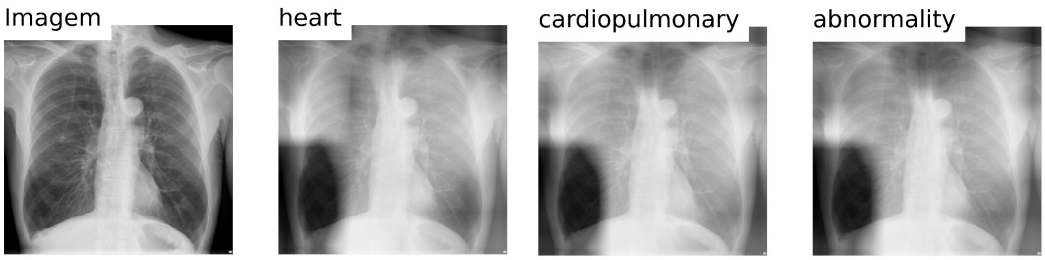

Figura B.1. A primeira figura, intitulada imagem, representa o exame original. As três imagens seguintes ilustram exemplos de marcações feitas em palavras relevantes do laudo, todas relacionadas ao termo mencionado no título. É observável que as regiões destacadas pelo modelo para gerar essas marcações são consistentes e idênticas entre si. A sentença do laudo gerada: *heart size is acute cardiopulomonary abnormality.*

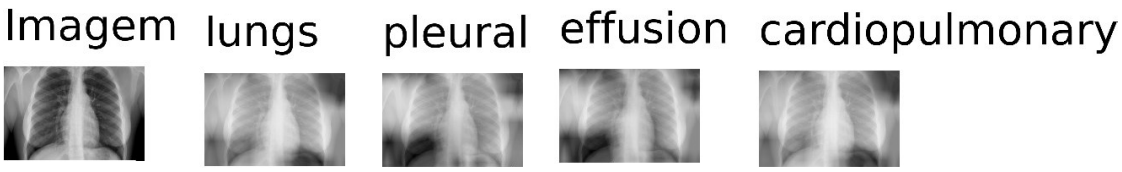

Figura B.2. A primeira figura exibe a imagem original do exame. É crucial observar que, para cada uma das palavras relevantes no laudo gerado, as regiões de atenção variam. No entanto, é interessante notar que a rede neural demonstrou uma tendência para focar em áreas quase que integralmente abrangentes da imagem do exame, mesmo com essa variação nas regiões de interesse. A sentença do laudo gerada: *the lungs are clear unremarkable.no focal consolidatio, pneumothorax, or pleural effusion. there lungs there is within acute limits.*

Figura B.3. A primeira imagem ilustra o exame original, enquanto as imagens subsequentes mostram marcações com as regiões de atenção. Cada uma dessas áreas específicas de atenção é meticulosamente selecionada para produzir a palavra mencionada no título. A sentença do laudo gerada: *the heart is normal in size. the mediastinum is unremarkeble. the lungs are clear.no acute disease.*