Redes neurais convolucionais aplicadas na detecção de pneumonia através de imagens de raio-x

Luan Silva
Faculdade de Computação e Engenharia Elétrica
Universidade Federal do Sul e Sudeste do Pará
Marabá, Pará 68507-590
Email: luansilvatec@unifesspa.edu.br

Leandro Araújo
Faculdade de Computação e Engenharia Elétrica
Universidade Federal do Sul e Sudeste do Pará
Marabá, Pará 68507-590
Email: leandronetlink@unifesspa.edu.br

Victor Souza
Faculdade de Computação e Engenharia Elétrica
Universidade Federal do Sul e Sudeste do Pará
Marabá, Pará 68507-590
Email: victoor.soouza@unifesspa.edu.br

Adam Santos
Faculdade de Computação e Engenharia Elétrica
Universidade Federal do Sul e Sudeste do Pará
Marabá, Pará 68507-590
Email: adam.dreyton@gmail.com

Raimundo Neto
Faculdade de Computação e Engenharia Elétrica
Universidade Federal do Sul e Sudeste do Pará
Marabá, Pará 68507-590
Email:netobarros@unifesspa.edu.br

***Resumo*—Segundo a Organização Mundial da Saúde (OMS), a pneumonia mata cerca de 2 milhões de crianças menores de 5 anos e é constantemente estimada como a principal causa de mortalidade infantil, matando mais crianças do que o HIV/AIDS, a malária e o sarampo juntos. A aplicação de técnicas de aprendizagem profunda para classificação de imagens médicas cresceu consideravelmente nos últimos anos. Esta pesquisa apresenta três implementações de redes neurais convolucionais (RNCs): ResNet50, VGG-16 e InceptionV3. Essas RNCs são aplicadas com o objetivo de solucionar o problema de classificação de radiografias médicas de pessoas com pneumonia, a fim de auxiliar no diagnóstico da doença. As três arquiteturas utilizadas nesta pesquisa obtiveram resultados satisfatórios. A ResNet50 superou InceptionV3 e VGG-16, alcançando a maior porcentagem de *precision* em treinamento e teste, assim como resultados superiores no *recall* e *f1-score*. O *f1-score* da ResNet50 para a classe normal foi de 88,42%, comparado a 81,54% para a InceptionV3 e 81,42% para a VGG-16. Para a classe de pneumonia foi 95,10% contra 92,82% da InceptionV3 e 92,54% da VGG-16.**

***Keywords*—Aprendizagem profunda; Reconhecimento de padrões; Redes neurais convolucionais.**

I. Introdução

O reconhecimento de objetos em imagens é uma atividade simples para humanos e está se tornando cada vez mais fácil para os computadores, devido aos avanços em estudos da área de visão computacional que utilizam redes neurais de aprendizagem profunda. A aprendizagem profunda (*deep learning*) é uma sub-área da aprendizagem de máquina (*machine learning*), a qual emprega algoritmos para processar dados, se inspirando no processamento feito pelo cérebro humano, simulando um conjunto de neurônios conectados que são organizados em camadas, tal que cada um deles faz o processamento de informações e passa o resultado obtido a partir de uma dada entrada para a camada seguinte [1].

Esses algoritmos com base na operação de convolução tiveram grandes avanços a partir do ano de 2012 e vêm sendo utilizados em diversos problemas de classificação que envolvem imagens no geral, até mesmo imagens médicas. A aplicação de técnicas de aprendizagem profunda para classificação de imagens médicas teve um crescimento considerável nos últimos anos. Várias pesquisas abordam este tema, como realizar a classificação de imagens para ajudar no diagnóstico precoce da tuberculose, e classificar lesões através de radiografia torácica [2]; além de serem usadas para a classificação de imagens de raio-x de pessoas com pneumonia e outras lesões [3]–[5]. Especificamente, a pneumonia é uma forma de infecção respiratória aguda que afeta os pulmões. Esses órgãos respiratórios são compostos de pequenos sacos chamados alvéolos, que se enchem de ar quando uma pessoa saudável respira. Quando um indivíduo tem pneumonia, os alvéolos ficam cheios de pus e líquido, o que torna a respiração dolorosa e limita a ingestão de oxigênio [6].

Esta pesquisa apresenta três implementações de redes neurais convolucionais (RNCs): ResNet50, VGG-16 e InceptionV3, amplamente utilizadas na literatura. Elas possuem paradigmas diferentes, sobretudo em relação à sua arquitetura, para a extração de características de imagens. Portanto, são boas escolhas para estudos comparativos dessa área. Essas RNCs são aplicadas com o objetivo de resolver o problema de classificação de imagens médicas a partir de raio-x de pessoas com pneumonia, a fim de ajudar no diagnóstico da doença. O objetivo deste trabalho é realizar um estudo comparativo entre as implementações, avaliando qual delas têm o melhor desempenho na classificação dessas imagens. Este artigo está estruturado da forma como segue. Na seção 2 é apresentado o conceito de RNC, sua estrutura e disposição das camadas.

Além disso, nessa seção são apresentadas as arquiteturas que foram utilizadas (ResNet50, VGG-16 e InceptionV3). Na seção 3 é discutida a metodologia empregada neste trabalho, bem como é especificada a base de dados que foi utilizada. Na seção 4 são apresentados os resultados das implementações das três arquiteturas supracitadas. Por fim, na seção 5, são abordados as considerações finais e possíveis trabalhos futuros a serem desenvolvidos a partir deste estudo.

II. Problema

Segundo a Organização Mundial da Saúde (OMS), a pneumonia mata cerca de 2 milhões de crianças menores de 5 anos a cada ano e é constantemente apontada como a principal causa da mortalidade infantil, matando mais crianças do que HIV/AIDS, malária e sarampo combinados.

A OMS relata que quase todos os casos (em média 95%) de crianças com pneumonia ocorrem em países em desenvolvimento, particularmente no Sudeste Asiático e na África. Patógenos bacterianos e virais são as duas principais causas de pneumonia, mas exigem formas de gerenciamento muito diferentes. A pneumonia bacteriana requer encaminhamento urgente para tratamento imediato com antibiótico, enquanto a pneumonia viral é tratada com precaução. Assim, o diagnóstico preciso e oportuno é indispensável. Um elemento chave no diagnóstico é o dado radiográfico, uma vez que as radiografias do tórax são rotineiramente obtidas como padrão de atendimento e podem ajudar a diferenciar os variados tipos de pneumonia [7]. Entretanto, uma interpretação radiológica rápida de imagens nem sempre está disponível, particularmente nos locais com poucos recursos em que a pneumonia infantil tem a maior incidência e maiores taxas de mortalidade.

O processo de interpretação de agentes (tumores cerebrais ou anomalias nos pulmões, por exemplo) é uma atividade complexa e, por isso, faz-se necessário o uso de técnicas de processamento de imagem, frequentemente combinadas à técnicas de aprendizado de máquina, para identificá-los [8].

Um sistema de triagem (classificação), assistido por computador, mitigaria essas questões, fornecendo a interpretação inicial do CXR (*Chest X-Ray*) para extrair uma classificação de normal/infectado por imagem. O conceito de Diagnóstico Auxiliado por Computador (*Computer-Aided Diagnosis - CAD*) para radiografias de tórax existe há cinquenta anos. Tradicionalmente, sistemas CAD usam a técnica de aprendizado de máquina para executar tarefas de análise das relações de dados existentes [2].

III. Redes Neurais Convolucionais

RNC são arquiteturas biologicamente inspiradas capazes de serem treinadas e aprenderem/generalizarem representações invariantes a escala, translação, rotação e transformações afins [9], [10]. As RNCs compõem um dos tipos de algoritmos da área conhecida como aprendizagem profunda e são projetadas para uso com dados de múltiplas dimensões, tornando-as boas candidatas para a solução de problemas envolvendo reconhecimento de imagens [11]. De maneira semelhante aos processos tradicionais de visão computacional, uma RNC é capaz de aplicar filtros em dados não-estruturados, mantendo a relação de vizinhança entre os *pixels* da imagem ao longo do processamento da rede. A Figura 1 demonstra a estrutura de uma RNC.

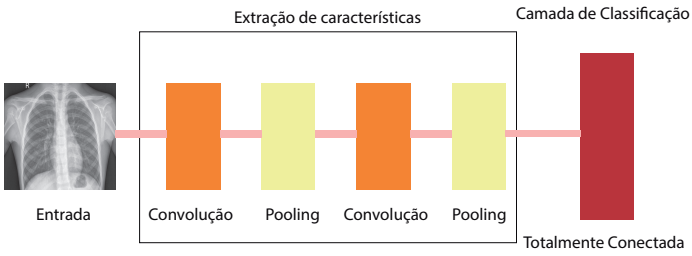

Figura 1. Estrutura de uma RNC.

As RNCs têm sido usadas pela comunidade científica em imagens médicas por causa de seu excelente desempenho demonstrado em visão computacional. Usualmente, três tipos de camadas são usadas para construir uma RNC: camada convolucional, camada de agrupamento (*pooling*) e camada totalmente conectada [12].

As camadas convolucionais usam um número de filtros sobre a imagem para obter uma série de mapas de características (*features*), um para cada filtro. Por sua vez, as camadas de agrupamento são responsáveis por sumarizar esses mapas de características evidenciando as características mais relevantes. A tarefa de classificação é feita pela camada totalmente conectada [12].

A principal camada dessas redes é a camada de convolução, sendo sua função a de aplicar máscaras nas imagens de entrada, com base em uma vizinhança de *pixels*. A saída produzida nessa operação são os filtros de convolução (matrizes) que armazenam os pesos das conexões entre os neurônios [13], [14]. A convolução é representada por um operador linear entre duas funções que produz uma terceira função. Esta última representa as áreas de superposição das máscaras que são aplicadas na imagem. A distinção de padrões de dados, com essa operação, é dada por campos receptivos locais (pequenas porções espaciais de *pixels*), que são representados por estruturas de neurônios localmente conectadas [15].

O compartilhamento de pesos na camada de convolução garante que os filtros sejam aplicados em diferentes posições na imagem, diminuindo significativamente o número de dados a serem aprendidos [16]. A ativação é uma estrutura muito importante na camada de convolução, realizando um ajuste entre um conjunto de neurônios $y_i(v)$, com uma função de ativação f sobre uma convolução $h(v)$. Isso produz os mapas de características que armazenam as informações aprendidas pelos filtros [14].

Outra camada que também é muito importante nas RNCs é a camada de agrupamento. Ela é responsável por reduzir a dimensionalidade dos mapas de características, diminuindo a largura e a altura. A operação de *pooling* possibilita uma invariância espacial. O agrupamento de características, na maioria das arquiteturas de convolução, utiliza uma função de

Max pooling, determinando o valor máximo do agrupamento em dada vizinhança [13], [17].

As próximas camadas das RNCs desempenham o papel de regressão das ativações. Em qualquer rede desse tipo, após a camada de agrupamento, é necessário ao menos uma camada totalmente conectada, servindo para criar caminhos de decisões, a partir dos filtros obtidos na camada anterior [18]. A última camada das redes convolucionais também é totalmente conectada, sendo responsável por realizar a classificação dos dados. Nessa situação, uma função de ativação determina a identificação das saídas em classes. A ativação mais utilizada é a função *Softmax* para problemas de múltiplas classes e *Sigmoid* para problemas binários. A *Softmax* determina a probabilidade de um conjunto de dados pertencer a uma classe de acordo com sua ocorrência [14]. O treinamento de uma RNC é realizado, na maioria dos casos, através do algoritmo *backpropagation*, que ajusta os pesos dos neurônios por intermédio da propagação e correção de erros relacionados com a classificação de dados da etapa de treinamento, tendo suporte da otimização proporcionada pelo gradiente descendente.

As arquiteturas de RNCs são criadas para se adequar a uma quantidade de imagens e classes, proporcionando um reconhecimento de padrões mais robusto. Vários estudos investigam o tamanho dessas arquiteturas e a sua eficácia. Nesse sentido, o *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) avalia abordagens de identificação e classificação que utilizam redes neurais profundas. Uma das suas motivações é promover arquiteturas que obtém grandes progressos [13]. Neste estudo, três dessas arquiteturas de RNC amplamente utilizadas pela literatura são consideradas [2], [4], [5]: VGG-16, ResNet50 e a InceptionV3.

A. VGG-16

A VGG-16 é um modelo de RNC proposto por K. Simonyan e A. Zisserman [17]. Foi um dos famosos modelos submetidos ao ILSVRC-2014. Ele fez melhoria sobre a AlexNet substituindo grandes filtros convolucionais (11 e 5 na primeira e segunda camadas convolucionais, respectivamente) por múltiplos filtros de tamanho 3 × 3 um após o outro.

A arquitetura da VGG-16 é mostrada na Figura 2. A entrada para a primeira camada de convolução tem o tamanho padrão de 224 x 224, aceitando outros tamanhos. A imagem é passada através de uma pilha de camadas convolucionais, onde os filtros foram usados com um campo receptivo muito pequeno de 3 × 3. O agrupamento espacial é realizado por cinco camadas considerando uma função de máximo. Esse agrupamento é executado em uma janela de 2 × 2 pixels [17].

Três camadas totalmente conectadas (FC) seguem uma pilha de camadas convolucionais (que tem uma profundidade diferente em diferentes arquiteturas): as duas primeiras têm 4096 neurônios cada, a terceira executa a classificação. A camada final é aplica a função *Softmax*. Todas as camadas ocultas estão equipadas com a não linearidade do tipo ReLU.

B. ResNet50

A ResNet ou rede residual é um modelo de RNC clássica usada como *backbone* para muitas tarefas de visão computacional. Esse modelo foi o vencedor do desafio ImageNet em 2015. A ResNet50 é uma rede residual com 50 camadas. O avanço na arquitetura da ResNet nos permite treinar redes neurais extremamente profundas, com mais de 150 camadas. A arquitetura da ResNet50 possui duas características principais:

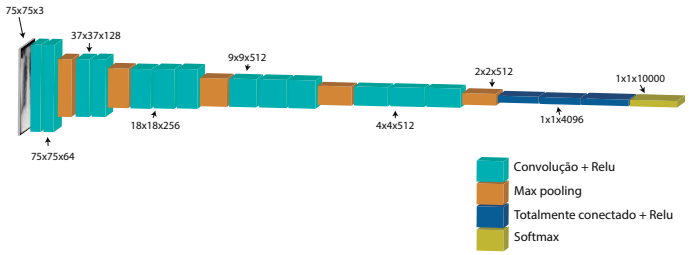

Figura 2. Arquitetura da VGG-16.

- "*Identity shortcut connections*": uma estratégia de "atalhos" ou "conexões de salto", que pulam pares de grupos de camadas convolucionais. São também chamadas *gated units* ou *gated recurrent units*, apesar de não apresentarem uma recorrência no sentido tradicional dos modelos de redes neurais recorrentes;
- foco pesado em normalização de lotes (*batch normalization*).

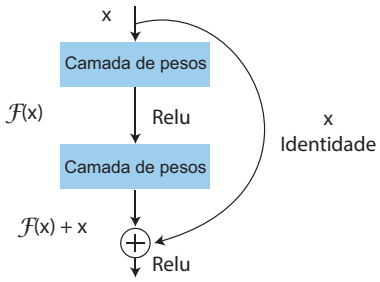

Figura 3. Bloco Residual da rede ResNet.

De forma simplificada, a ideia dos atalhos na ResNet50 é evitar que a rede, muito profunda, morra por esvanecimento de gradientes através do empilhamento de mapeamentos de identidades que, do ponto de vista matemático, estão simplesmente empilhando camadas que não fazem nada. Com isso, como mostra a Figura 3 , a ResNet50 utiliza, num determinado ponto, um sinal que é a soma do sinal produzido pelas duas camadas convolucionais anteriores somado ao sinal transmitido diretamente do ponto anterior a estas camadas, juntando um sinal processado com um sinal de uma etapa anterior no processamento [19].

A arquitetura básica de uma ResNet qualquer é descrita na Figura 4. É aplicada na entrada um bloco de preenchimento zero de (3,3). No Estágio 1, a Convolução 2D tem 64 filtros de forma (7,7) e usa uma passada de (2,2). O *BatchNorm* é aplicado ao eixo dos canais da entrada. O *MaxPooling* usa uma janela (3,3) e uma passada (2,2). No Estágio 2, o bloqueio convolucional usa três conjuntos de filtros. Os 2 blocos de identidade usam três conjuntos de filtros. Estágio 3, o bloqueio

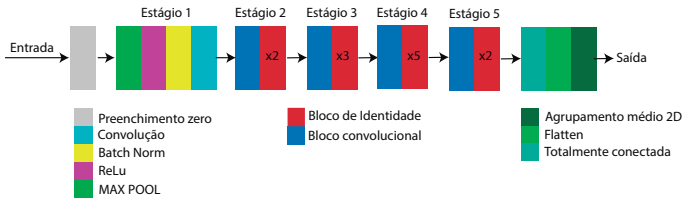

Figura 4. Arquitetura da rede ResNet.

convolucional usa três conjuntos de filtros. Os 3 blocos de identidade usam três conjuntos de filtros. No Estágio 4, o bloco convolucional usa três conjuntos de filtros de tamanho. Os 5 blocos de identidade usam três conjuntos de filtros. Estágio 5, o bloqueio convolucional usa três conjuntos de filtros. Os 2 blocos de identidade usam três conjuntos de filtros. O *Average Pooling* (Agrupamento médio) 2D usa uma janela de forma (2,2). O *flatten* (nivelamento da camada) não possui nenhum hiperparâmetro ou nome. E a camada totalmente conectada (Densa) reduz sua entrada para o número de classes usando uma ativação *Softmax* [19], [20] .

C. InceptionV3

A rede GoogLeNet foi vencedora do ILSVRC no ano de 2014 [21]. Sua principal contribuição foi o desenvolvimento de um módulo de Inception que reduziu drasticamente o número de parâmetros na rede para 4 milhões, comparado a rede AlexNet com 60 milhões. O objetivo principal do módulo Inception é atuar como um extrator de características em vários níveis, computando convoluções 1x1, 3x3 e 5x5 dentro do mesmo módulo da rede, conforme mostra a Figura 5.

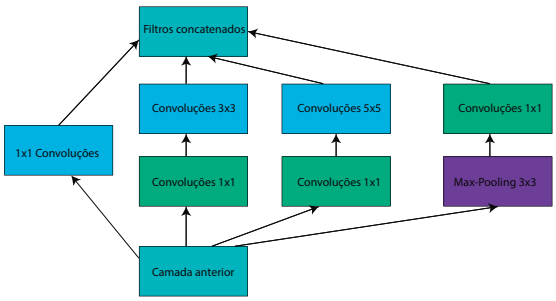

Figura 5. Módulo Inception.

O InceptionV3 é um modelo de RNC voltada para a resolução do problema de classificação de imagens, que foi treinada no conjunto de dados ImageNet. Esta rede, que apresentou bom desempenho com baixo custo computacional no ano de 2015, reduziu os parâmetros estimados pela mesma, fazendo com que ela possua desempenho computacional superior, em relação as redes VGG, durante seu treinamento [21], [22]. A arquitetura da InceptionV3 é mostrada na Figura 6. O modelo é composto de componentes simétricos e assimétricos, incluindo convoluções, *Average Pooling* (AvgPool), *Max Pooling* (MaxPool), concatenações (Concat), desistências (Droupout) e camadas totalmente conectadas. A normalização em lote é usada extensivamente em todo o modelo e aplicada às entradas de ativação. A classificação é realizada na última camada, utilizando a *Softmax* [23].

IV. Metodologia

A linguagem de programação utilizada para a codificação das arquiteturas de Redes Neurais Convolucionais foi o Python, utilizando as bibliotecas Keras e o TensorFlow. O Keras é uma biblioteca que fornece blocos de construção altamente poderosos e abstratos para construir redes de aprendizagem profunda [24], [25]. O TensorFlow é um sistema de aprendizado de máquina que opera em grande escala e em ambientes heterogêneos [26].

A infraestrutura utilizada para a realização das simulações e geração dos modelos da VGG-16, ResNet50 e InceptionV3, com os pesos treinados, foi o Google Colab (Colaboratory), no qual oferece 12GB de RAM e uma Tesla K80 GPU. Ele fornece tempos de execução do Python 2 e 3 pré-configurados com as bibliotecas essenciais de aprendizado de máquina e inteligência artificial, como o TensorFlow, Matplotlib e Keras. A máquina virtual sob o tempo de execução (VM) é desativada após um período de tempo, e todos os dados e configurações do usuário são perdidos. No entanto, o notebook é preservado e também é possível transferir arquivos do disco rígido da VM para a conta do Google Drive do usuário [27].

Antes de abordarmos acerca dos resultados obtidos, precisamos fazer uma análise sobre a base de dados, matriz de confusão, uma técnica utilizada durante os treinamentos (*data augmentation*) e como foi definido os parâmetros para os testes. Para uma análise mais precisa, foram realizadas 10 simulações para cada rede, com o objetivo de obtermos um comportamento médio de cada arquitetura.

A. Base de Dados

Para este trabalho foi utilizada uma base de dados com imagens de raio-x do peito de pacientes com diferentes tipos de pneumonia, como os exemplos da Figura 6 [7]. São 3 classes (normal, pneumonia viral e pneumonia bacterial). As imagens foram coletadas e etiquetadas de um total de 5.856, sendo 5.232 imagens de raio-x do peito de crianças, incluindo 3.883 categorizadas como representação de pneumonia (2.538 bacterina e 1.345 viral) e 1.349 normal; e as outras 624 imagens são de pacientes de diversas faixas etárias de idade [7].

A Figura 7 mostra alguns exemplos de imagens presentes na base de dados. A radiografia de tórax normal (painel da esquerda) mostra pulmões limpos sem nenhuma área de opacificação anormal na imagem. A pneumonia bacteriana

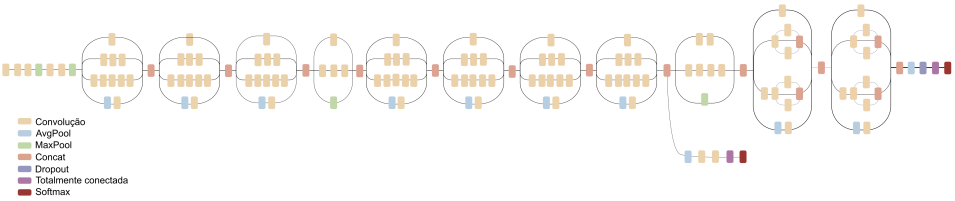

Figura 6. Arquitetura da InceptionV3.

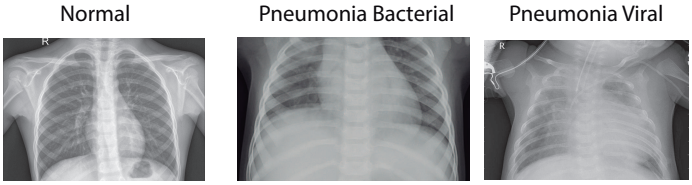

Figura 7. Exemplos ilustrativos de raios X torácicos em pacientes.

(média), geralmente exibe consolidação lobar focal, neste caso, no lobo superior direito (setas brancas), enquanto a pneumonia viral (direita) se manifesta com um padrão "intersticial"mais difuso em ambos os pulmões [7].

Porém, resolvemos realizar os testes apenas com duas classes (normal e pneumonia), com o objetivo de encontrar uma arquitetura ideal para posteriormente aumentar o número de classes. A divisão ficou na seguinte forma: 4273 imagens de raio x de pacientes com pneumonia e 1583 imagens de pacientes em condições normais.

B. Matriz de Confusão

Uma matriz de confusão compara as previsões de um modelo com o padrão verdadeiro. A diagonal principal da matriz representa classes que foram corretamente previstas (do inglês, *True Positive* - TP (Verdadeiro Positivo) e *True Negative* - TN (Verdadeiro Negativo)), enquanto elementos da diagonal secundária representam indivíduos que foram classificados de forma errada (do inglês, *False Positive* - FP (Falso Positivo) e *False Negative* - FN (Falso Negativo)) [22]. A partir da matriz de confusão, é possível calcular as métricas para avaliar se o algoritmo está ou não conseguindo bons resultados. A Tabela 1 mostra um exemplo de matriz de confusão que será utilizadas nas seções seguintes para fazer os cálculos das métricas.

Tabela I
EXEMPLO DE UMA MATRIZ DE CONFUSÃO.

	Normal	**Pneumonia**
Normal	10 (TP)	5 (FP)
Pneumonia	4 (FN)	6 (TN)

C. Data augmentation

O *Data Augmentation* é uma forma utilizada para reduzir o *overfitting* em redes neurais profundas. Essa técnica cria vários exemplos sintéticos de um mesmo conjunto de dados, permitindo um melhor aprendizado, que é oferecido com o aumento artificial de informações . [28] Ela faz modificações nas imagens a cada iteração durante o treinamento com o objetivo de fazer as redes não conhecerem os dados de treinamento completamente, para que assim, as redes não percam sua capacidade de generalização, se tornando especialista apenas nos dados de treinamento [22].

D. Parâmetros para os testes

A base de dados, com 5856 imagens, foi dividida em dados de treino (4684 imagens) e dados de teste (1172 imagens). Algumas delas possuem resoluções diferentes, podendo chegar a ser 1917x1432 pixels. Foi necessário a padronização delas para que pudessem passar pela rede, tendo assim uma resolução padrão de 75x75 pixels cada uma delas. A arquitetura da ResNet50 utilizada foi a ResNet50v1 do Keras. Alguns parâmetros utilizados nos teste por ambas as arquiteturas:

- Função de ativação da camada de saída: Sigmoid
- Tamanho do Lote (*batch*): 16
- Função de ativação nas camada escondidas: ReLU (*Rectified Linear Units*)
- Otimizador: Adam
- Quantidade de épocas: 200.
- Droupout: 0.2.
- Taxa de aprendizado: A taxa de aprendizado inicial é: $1x10^{-3}$, sendo multiplicada por taxas menores no decorrer das épocas, a partir da Época 81: $1x10^{-1}$ / Época 121: $1x10^{-2}$ / Época 161: $1x10^{-3}$/ Época 181: $0.5x10^{-3}$
- Função de Custo: *binary cross-entropy* (entropia cruzada)

E. Métricas de avaliação dos resultados

Alguns parâmetros de avaliacão foram utilizados para melhorar a visualização dos resultados.

1) precision: A precisão é a relação (1) em que TP é o número de verdadeiros positivos e FP o número de falsos positivos. A precisão é, intuitivamente, a capacidade do classificador de não rotular como positiva uma amostra que é negativa. [29]

$$precision = TP/(TP + FP) \quad (1)$$

2) recall: O *recall* é a relação (2) em que TP é o número de verdadeiros positivos e FN o número de falsos negativos. O recall é, intuitivamente, a capacidade do classificador de encontrar todas as amostras positivas. [30]

$$recall = TP/(TP + FN) \quad (2)$$

3) f1-score: O *f1-score* pode ser interpretado como uma média ponderada do *precision* e do *recall*, em que um *f1-score* alcança seu melhor valor em 1 e o pior escore em 0. A contribuição relativa de *precision* e *recall* para o *f1-score* é igual. A fórmula para a pontuação de F1 é (3). [31]

$$f1 - score = 2 * (precision * recall)/(precision + recall) \quad (3)$$

V. Resultados e Discussões

A Tabela 2 apresenta os resultados da média das 10 simulações das métricas de *precision*, *recall* e *f1-score*. A ResNet50 obteve resultados superiores (*precision* - Normal: 87,32%, Pneumonia:95,72% / *recall* - Normal: 89,70%, Pneumonia:94,52%) as outras duas arquiteturas (InceptionV3 (*precision* - Normal: 85,08%, Pneumonia: 91,52% / *recall* - Normal: 78,59%, Pneumonia: 94,22%) e VGG-16 (*precision* - Normal: 82,80%, Pneumonia: 91,94% / *recall* - Normal: 80,11%, Pneumonia: 93,15%)), em ambas as classes (normal e pneumonia). Isso mostra que a ResNet50 é mais precisa em classificar imagens de pacientes em condições normais e pacientes com pneumonia, além de ter uma capacidade maior de encontrar todas as amostras positivas e negativas, enquanto a InceptionV3 obtém um desempenho superior à VGG-16 na classe normal do *precision* e na classe de Pneumonia do *recall* e a VGG-16 leva uma pequena vantagem sobre a InceptionV3 na classe de Pneumonia do *precision* e da classe normal do *recall*.

Tabela II
Média das 10 simulações de *precision*, *recall* e *f1-score*.

Arquitetura	Classe	precision	recall	f1-score
ResNet50	Normal	87,32%	89,70%	88,42%
	Pneumonia	95,72%	94,52%	95,10%
InceptionV3	Normal	85,08%	78,59%	81,54%
	Pneumonia	91,52%	94,22%	92,82%
VGG-16	Normal	82,80%	80,11%	81,42%
	Pneumonia	91,94%	93,15%	92,54%

O *f1-score* é a média ponderada das métricas acima, na qual a ResNet50 é naturalmente superior, sendo que a InceptionV3 vem logo em seguida.

As Figuras 8 e 9 demonstram o comportamento da acurácia das arquiteturas durante a etapa de treinamento e teste. Tanto na etapa de treinamento como de teste, a ResNet50 (curva azul) obteve uma convergência mais rápida e superior a 90% antes da época 50, enquanto a InceptionV3 (curva preta) só conseguiu atingir essa acurácia no treinamento na epoca 75 e a VGG-16 (curva vermelha) na época 100. A curva da acurácia da ResNet50 foi amplamente superior, enquanto as curvas da InceptionV3 e da VGG-16 demonstram um certo equilíbrio e proximidade, principalmente ápos a época 100.

Em relação ao *loss* (Perda), nas Figuras 10 e 11, é interessante perceber que em ambas as etapas, de treinamento e teste, a Função de Custo da ResNet50 começa com valores altos e é minimizada aos poucos, o que não compromete o seu desempenho. Os custos da InceptionV3 e da VGG-16 começam

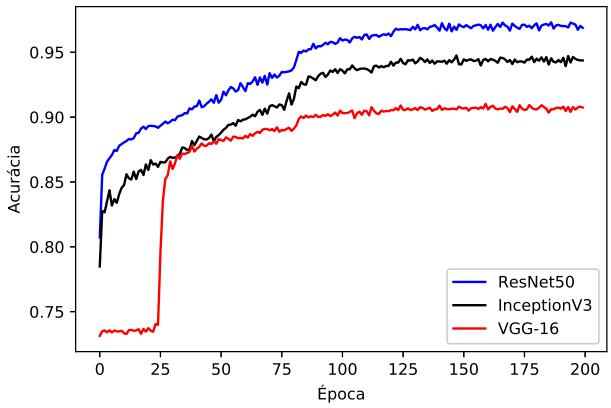

Figura 8. Média de 10 Simulações da Acurária no treinamento das três arquiteturas.

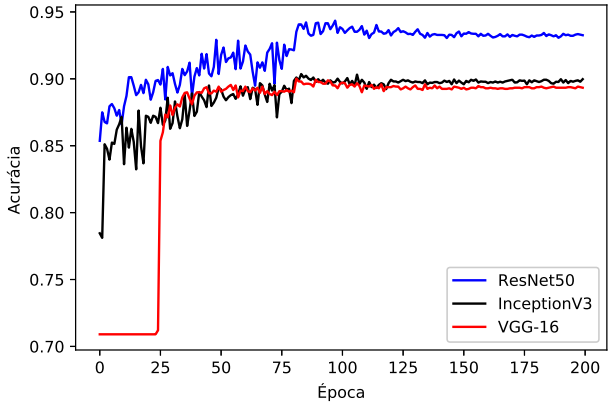

Figura 9. Média de 10 Simulações da Acurária no teste das três arquiteturas.

baixos, mas não tendem a diminuir tanto quanto a ResNet50 no decorrer das épocas. As arquiteturas possuem comportamentos parecidos no treinamento, mas divergem muito na fase do teste. Os altos picos no comportamento médio das arquiteturas é graças a taxa de aprendizado que é otimizada e diminuída no decorrer das épocas e ao otimizador Adam por ser mais lento.

As Figuras 12, 13 e 14 mostram as matrizes médias de confusão da ResNet50, InceptionV3 e VGG-16. Nelas estão dispostas a quantidade de acertos que as arquiteturas obtiveram em relação a classe normal, ou seja, representa os Verdadeiros Positivos (ResNet50: 305 imagens/ InceptionV3: 268 imagens / VGG-16: 273 imagens), que são imagens de raio-x na qual os pacientes estão com os pulmões em condições normais, e os Verdadeiro Negativos (ResNet50: 785 imagens/ InceptionV3: 783 imagens / VGG-16: 774 imagens), que representa a classe de imagens de pacientes que estão com pneumonia e foram assim corretamente classificados.

Já em relação aos erros, é possível observar os Falsos Positivos (ResNet50: 35 imagens/ InceptionV3: 73 imagens

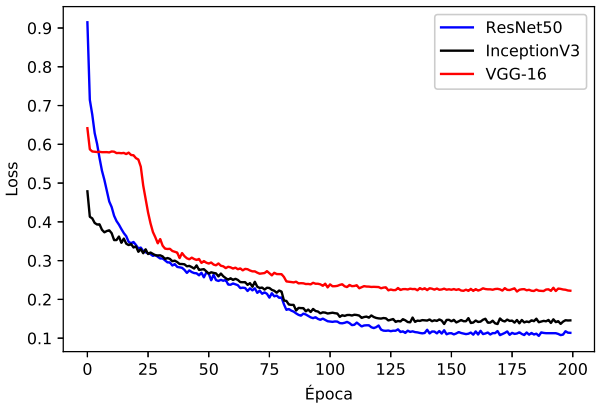

Figura 10. Média de 10 Simulações de Loss no treinamento das três arquiteturas.

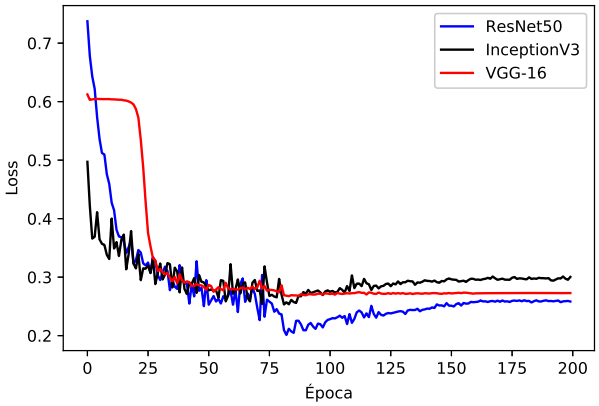

Figura 11. Média de 10 Simulações de Loss no teste das três arquiteturas.

/ VGG-16: 67 imagens), que são as imagens de pacientes em condições normais, mas que foram confundidas como sendo imagens de pacientes com pneumonia, e os Falsos Negativos (ResNet50: 45 imagens/ InceptionV3: 48 imagens / VGG-16: 56 imagens) nos quais representam as imagens de pacientes que possuem pneumonia, mas que foram classificadas em condições normais, o que, obviamente, é uma classificação bastante perigosa.

VI. Considerações Finais e Trabalhos Futuros

As três arquiteturas utilizadas nesta pesquisa obtiveram bons resultados e, como foi abordado, a ResNet50 obteve um desempenho superior à InceptionV3 e a VGG-16. Ela conseguiu o maior percentual de acucária no treinamento e teste, além de resultados superiores nas métricas de *precision*, *recall* e *f1-score*. A ResNet50 conseguiu diminuir os erros Falso Negativos, que são extremamente perigosos. Minimizou a função de custo, tendo o menor percentual próximo à época 75. Uma possibilidade para a quantidade de erros que as arquiteturas obtiveram está relacionada com a falta de

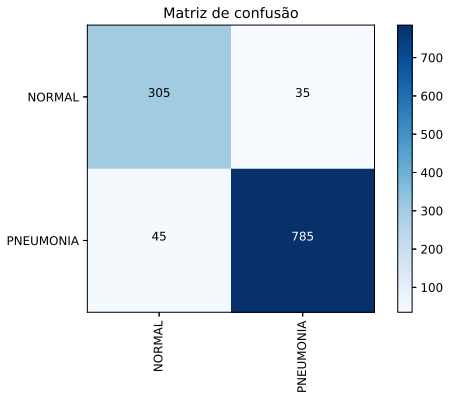

Figura 12. Média de 10 Simulações das Matrizes de Confusão da ResNet50.

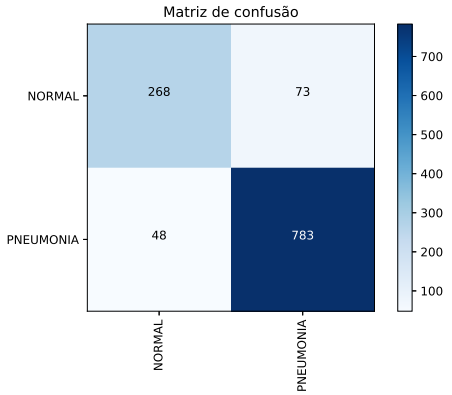

Figura 13. Média de 10 Simulações das Matrizes de Confusão da InceptionV3.

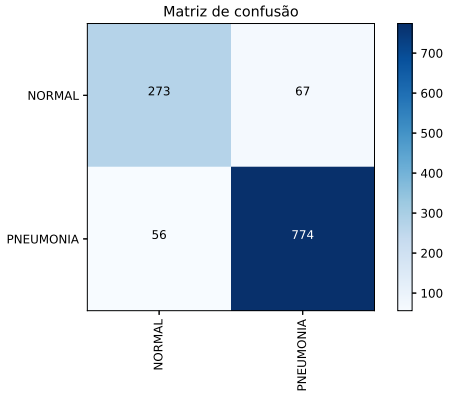

Figura 14. Média de 10 Simulações das Matrizes de Confusão da VGG-16.

balanceamento da quantidade de imagens de crianças e adultos presente na base de dados. Outra possibilidade está relacionada com a matriz de confusão, visto que ocorreram erros Falso Positivos e Falso Negativos.

Mas, a partir desta pesquisa, novas possibilidades podem ser exploradas. Arquiteturas como VGG-19, Xception, DenseNet podem ser utilizadas para efeito comparativo de desempenho,

podendo ser estudado o quão profundo uma rede neural convolucional precisa ser para de fato gerar um modelo ideal para classificação desses tipos de imagens.

Em relação à classificação de imagens de raio-x, novas imagens podem ser coletadas, possibilitando o aumento das classes de 2 (normal e pneumonia) para 4 (normal, pneumonia viral, pneumonia bacteriana e pneumonia fúngica), tornando o modelo mais robusto para classificação de imagens desse tipo.

Técnicas como *transfer learning* podem ser utilizadas para, tanto auxiliar na classificação de imagens de raio-x com pneumonia utilizando pesos pré-treinados de distribuições diferentes, como também para outros tipos de doenças, tais como: câncer ósseo, câncer de mama, tumores e assim por diante.

Outra possibilidade é a utilização da base de dados da *National Institutes of Health Chest X-Ray* para aprimorar os estudos com essas e outras arquiteturas de CNN's, como já citadas. É uma base de dados ampla, tanto de imagens (112.120 imagens) quanto de classes (15 classes) [4], [5].

Além disso, com um modelo ótimo e devidamente treinado, uma aplicação de tempo real e com Realidade Aumentada pode ser desenvolvida, como sendo um ferramenta de auxílio no diagnostico, para a classificação de imagens físicas de raio-x, captadas por uma câmera de celular, apresentando índices de probabilidade, no qual pode ajudar um médico no seu diagnostico precoce de alguma anomalia.

REFERÊNCIAS

[1] HAYKIN, S. S. Neural networks and learning machines. [S.l.]: Pearson Upper Saddle River, NJ, USA:, 2009. v. 3.

[2] Karnkawinpong, T., & Limpiyakorn, Y. (2018, December). Chest X-Ray Analysis of Tuberculosis by Convolutional Neural Networks with Affine Transforms. In Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence (pp. 90-93). ACM.

[3] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., ... & Dong, J. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5), 1122-1131.

[4] Wang, Xiaosong & Peng, Yifan & Lu, Le & Lu, Zhiyong & Bagheri, Mohammadhadi & Summers, Ronald. (2017). ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. arXiv:1705.02315.

[5] Wang, Xiaosong & Peng, Yifan & Lu, Le & Lu, Zhiyong & Bagheri, Mohammadhadi & Summers, Ronald. (2017). ChestX-ray14: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases.

[6] Who.int. (2016). Pneumonia. [online] Available at: https://www.who.int/en/news-room/factsheets/ detail/pneumonia [Accessed 25 May 2019].

[7] Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F. and Dong, J., 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5), pp.1122-1131.

[8] Neto, Crespo, and Sergio Arthur de La Hidalga. "Reconhecimento de tumores cerebrais utilizando redes neurais convolucionais."(2017).

[9] LIU, T.; ROSENBERG, C.; ROWLEY, H. A. Clustering billions of images with large scale nearest neighbor search. In: Proceedings of the Eighth IEEE Workshop on Applications of Computer Vision. Washington, DC, USA: IEEE Computer Society, 2007.

[10] JURASZEK, Guilherme De freitas; SILVA, Alexandre Gonçalves; DA SILVA, André Tavares. Reconhecimento de produtos por imagem utilizando palavras visuais e redes neurais convolucionais. Joinville: UDESC, 2014.

[11] AREL, I.; ROSE, D.; KARNOWSKI, T. Deep machine learning - a new frontier in artificial intelligence research [research frontier]. Computational Intelligence Magazine, IEEE, v. 5, n. 4, p. 13–18, 2010. ISSN 1556-603X.

[12] Wafa Mousser and Salima Ouadfel. 2019. Deep Feature Extraction for Pap-Smear Image Classification: A Comparative Study. In Proceedings of the 2019 5th International Conference on Computer and Technology Applications (ICCTA 2019). ACM, New York, NY, USA, 6-10. DOI: https://doi.org/10.1145/3323933.3324060

[13] SANTOS, Alan et al. Uma abordagem de classificação de imagens dermatoscópicas utilizando aprendizado profundo com redes neurais convolucionais. In: 17º Workshop de Informática Médica (WIM 2017). SBC, 2017.

[14] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In BMVC, volume 1, page 6.

[15] Shi, Z. and He, L. (2011). Current status and future potential of neural networks used for medical image processing. Journal of multimedia, 6(3):244–251.

[16] Kawahara, J., BenTaieb, A., and Hamarneh, G. (2016). Deep features to classify skin lesions. In Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on, pages 1397–1400. IEEE.

[17] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[18] Vedaldi, A. and Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In Proceedings of the 23rd ACM international conference on Multimedia, pages 689– 692. ACM.

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[20] Rizwan, Muhammad. "Residual Networks (Resnets) | Engmrk". Engmrk, 2019, https://engmrk.com/residual-networks-resnets/.

[21] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[22] SILVA, Rodrigo Emerson Valentim da. Um estudo comparativo entre redes neurais convolucionais para a classificação de imagens. 2018. 51 f. TCC (Graduação em Sistemas de Informação) Universidade Federal do Ceará, Campus de Quixadá, Quixadá, 2018.

[23] Nivrito, A. K. M., Md Wahed, and Rayed Bin. Comparative analysis between Inception-v3 and other learning systems using facial expressions detection. Diss. Brac University, 2016.

[24] Ketkar, Nikhil, Introduction to keras. Deep Learning with Python, 2017.

[25] Chollet, F. (2015). Keras.

[26] Martín Abadi and Paul Barham and Jianmin Chen and Zhifeng Chen and Andy Davis and Jeffrey Dean and Matthieu Devin and Sanjay Ghemawat and Geoffrey Irving and Michael Isard and Manjunath Kudlur and Josh Levenberg and Rajat Monga and Sherry Moore and Derek G. Murray and Benoit Steiner and Paul Tucker and Vijay Vasudevan and Pete Warden and Martin Wicke and Yuan Yu and Xiaoqiang Zheng, TensorFlow: A System for Large-Scale Machine Learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016.

[27] Carneiro, Tiago and Da Nóbrega, Raul Victor Medeiros and Nepomuceno, Thiago and Bian, Gui-Bin and De Albuquerque, Victor Hugo C and Reboucas Filho, Pedro Pedrosa, Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. IEEE Access, 2018.

[28] Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2, ICDAR '03, pages 958–, Washington, DC, USA. IEEE Computer Society.

[29] $Scikit - learn.org.(2019)sklearn.metrics.precision_score - scikit - learn0.21.2documentation.$ [online] Available at: Scikit-learn.org [Accessed 24 Jun. 2019].

[30] $Scikit - learn.org.(2019).sklearn.metrics.recall_score^{-}scikit - learn0.21.2documentation.$ [online] Available at: Scikit-learn.org [Accessed 24 Jun. 2019].

[31] $Scikit - learn.org.(2019).sklearn.metrics.f1_score - scikit - learn0.21.2documentation.$ [online] Available at: Scikit-learn.org [Accessed 24 Jun. 2019].