UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Daniel Ambrósio Ferreira Júnior

# Estudo sobre Transfer Learning na Detecção de Distorção Arquitetural em Imagens Mamográficas

São Carlos

2020

**Daniel Ambrósio Ferreira Júnior**

# Estudo sobre Transfer Learning na Detecção de Distorção Arquitetural em Imagens Mamográficas

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Marcelo A. C. Vieira

**São Carlos**

**2020**

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Júnior, Daniel Ambrósio Ferreira
J184e Estudo sobre Transfer Learning na Detecção de Distorção Arquitetural em Imagens Mamográficas / Daniel Ambrósio Ferreira Júnior; orientador Marcelo Andrade da Costa Vieira. São Carlos, 2020.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2020.

1. Transferência de Aprendizado. 2. Extração de Características . 3. Distorção Arquitetural Mamária. 4. Redes Neurais Convolucionais . 5. Processamento de Imagens. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

# FOLHA DE APROVAÇÃO

**Nome: Daniel Ambrosio Ferreira Junior**

**Título: "Estudo sobre Transfer Learning na detecção de distorção arquitetural em imagens mamográficas"**

**Trabalho de Conclusão de Curso defendido e aprovado em 22/06/2020,**

**com NOTA 10,0 ( DEZ, ZERO ), pela Comissão Julgadora:**

***Prof. Associado Marcelo Andrade da Costa Vieira - Orientador - SEL/EESC/USP***

***Prof. Associado Adilson Gonzaga - (Sênior - SEL/EESC/USP)***

***Mestre Arthur Chaves Costa - Doutorando - SEL/EESC/USP***

**Coordenador da CoC-Engenharia Elétrica - EESC/USP:**
**Prof. Associado Rogério Andrade Flauzino**

# AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me guiado ao longo da minha trajetória acadêmica, desde a minha entrada na faculdade até o fim dela, além de ter me dado forças pra prosseguir e consolo nos momentos mais difíceis.

Agradeço a minha família, especialmente aos meus pais Daniel Ambrósio Ferreira e Eunice Reis Ferreira, por todo suporte através das orações, ligações por telefone, e conselhos nos momentos mais decisivos.

Agradeço ao Arthur Chaves Costa por todo suporte oferecido sempre que precisei de ajuda.

Agradeço ao Leandro Guilherme por todos os ensinamentos sobre redes neurais convolucionais e pela paciência ao me emprestar o computador para fazer testes.

Agradeço a todos os meus amigos, especialmente ao Rafael Pagliarini Bagagli e ao Leonardo Maronezi Pizani, por toda ajuda durante o desenvolvimento do trabalho e por todas as risadas que ajudaram a acalmar os momentos de estresse.

Por fim, agradeço ao Professor Doutor Marcelo Andrade da Costa Vieira e a todos os professores e funcionários desta universidade, por contribuírem continuamente para um ensino superior público de qualidade e excelência.

*"O estudo, a busca da verdade e da beleza são domínios em que nos é consentido sermos crianças por toda a vida."*

*Albert Einstein*

# RESUMO

Ambrósio, D. **Estudo sobre Transfer Learning na Detecção de Distorção Arquitetural em Imagens Mamográficas**. 2020. 65p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020.

O presente trabalho é um estudo sobre a utilização de *transfer learning* em redes neurais convolucionais profundas (CNN) pré-treinadas para a detecção de distorção arquitetural mamária em exames de mamografia digital. Foram utilizadas três arquiteturas distintas de CNN pré-treinadas no conjunto de imagens ImageNet da Universidade de Stanford, sendo elas VGG16, MobileNet e ResNet50. O estudo não apenas visou a verificação do comportamento das arquiteturas diferentes quando igualmente configuradas, como também teve por objetivo observar comparativamente o comportamento de modelos utilizando de uma a quatro camadas densas durante a técnica de extração de características. Os hiperparâmetros para todas as redes utilizadas no trabalho foram os mesmos: otimizador Adam e função de perda *categorical cross entropy*. Foi utilizada a técnica de validação cruzada *k-fold*, com k = 6, na tentativa de fazer com que os modelos gerados tivessem uma melhor capacidade de generalização. Portanto, para cada arquitetura diferente, as redes neurais foram treinadas 6 vezes para cada quantidade de camadas densas variando de um a quatro. Os modelos foram treinados utilizando um conjunto de exames clínicos de mamografia digital, recortados em janelas de 224x224 pixels com auxílio de um médico radiologista, com e sem a presença de distorção arquitetural. Foram realizados treinamentos com e sem a utilização de *data augmentation*. Foi possível observar que para a arquitetura MobileNet, as redes neurais treinadas com duas camadas densas apresentaram resultados ligeiramente superiores aos demais. Para o caso das acurácias médias, os modelos treinados com duas camadas foram cerca de 8% maior do que o segundo melhor caso no conjunto com *data augmentation*, possuindo diferença estatisticamente significativa comprovada pelo método de Wilcoxon em relação aos outros modelos com diferentes números de camadas. Já para a arquitetura VGG16, os resultados foram bem similares entre si. O melhor desempenho obtido não ultrapassa em 1% o segundo melhor desempenho em acurácia. A arquitetura ResNet50, para todas as diferentes configurações de camadas densas, não foi capaz de aprender a identificar as distorções arquiteturais.

**Palavras-chave**: Transferência de Aprendizado; Redes Neurais Convolucionais; Processamento de Imagens; Extração de Características; Distorção Arquitetural Mamária

## ABSTRACT

Ambrósio, D. **Study on Transfer Learning in Detecting Architectural Distortion in Mammographic Images**. 2020. 65p. Undergraduate Final Project, Sao Carlos School of Engineering, University of Sao Paulo, Sao Carlos, Brazil, 2020

This paper is a study on the use of learning transfer in pre-trained deep convolutional neural networks (CNN) for the detection of breast architectural distortion in digital mammography. Three different architectures of CNN, pre-trained on Stanford University's ImageNet dataset, were used: VGG16, MobileNet and ResNet50. The study not only aimed at verifying the behavior of different architectures when equally configured, but also aimed at comparatively observing the behavior of models using one to four dense layers during the feature extraction technique. The hyperparameters for all networks used in the work were the same: Adam optimizer and categorical cross entropy loss function. The k-fold cross-validation technique was used, with k = 6, in an attempt to increase the generalization capacity of the trained models. Therefore, for each different architecture, neural networks were trained 6 times for each number of dense layers ranging from one to four. The data set used consists of digital mammography clinical exams, cut out in 224x224 pixel windows, with the help of a radiologist, with and without architectural distortion. The training process was performed with and without the use of data augmentation. It was possible to observe that for the MobileNet architecture, the neural networks trained with two dense layers presented results slightly superior to the others. When analyzing the average accuracy, the models trained with two layers were about 8% higher than the second best case in the set with data augmentation, with a statistically significant difference confirmed by the Wilcoxon method in relation to the other models with different numbers of layers. As for the VGG16 architecture, the results were very similar to each other. The best performance obtained does not surpass by 1% the second best performance in accuracy. The ResNet50 architecture, for all different configurations of dense layers, was not able to learn to identify architectural distortions.



**Keywords**: Transfer Learning; Convolutional Neural Network; Image Processing; Featyre Extraction; Breast Architectural Distortion

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

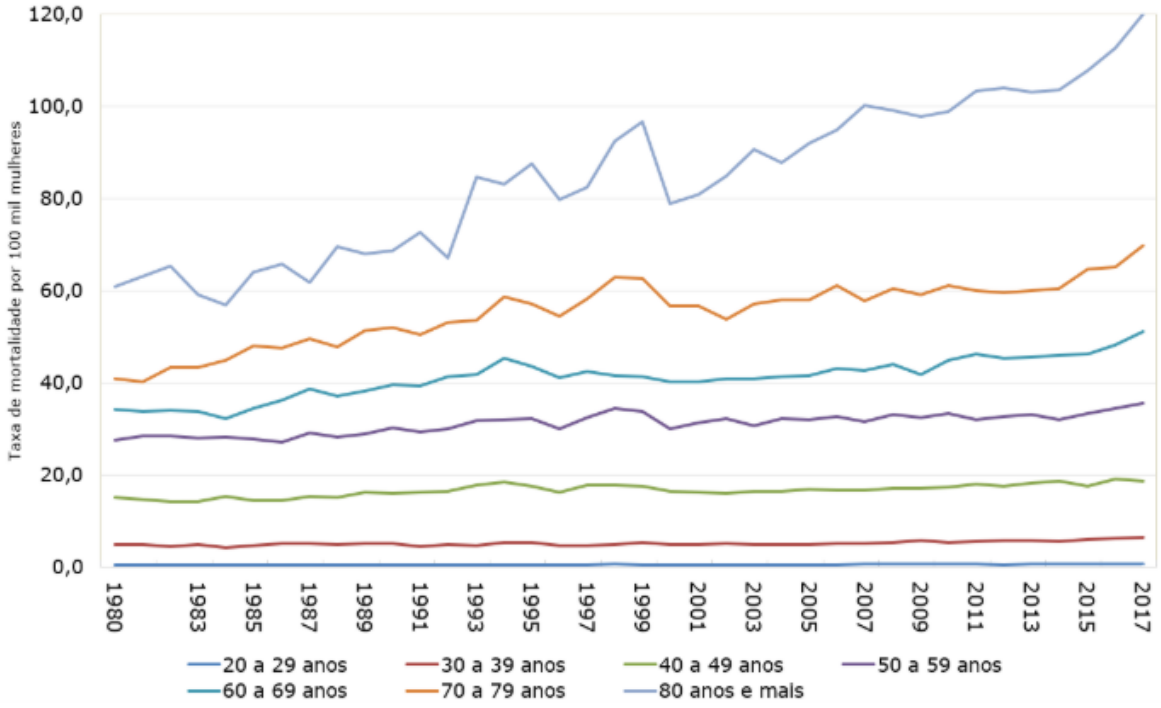
| | |
|---|---|
| CNN | *Convolutional Neural Network* |
| TL | *Transfer Learning* |
| DA | *Data Augmentation* |
| ROC | Característica de Operação do Receptor |
| FE | Extração de Características |
| FT | *Fine Tuning* |

## SUMÁRIO

# 1 INTRODUÇÃO

O câncer de mama corresponde a um grupo diverso de doenças que podem se manifestar com morfologias diferentes, assinaturas genéticas diferentes e até, por consequência, respostas terapêuticas diferentes. Mais comumente, seus sintomas estão associados ao aparecimento de nódulo, geralmente indolor, duro e irregular na mama, além de alterações no bico do peito e pele avermelhada (Instituto Nacional de Câncer, 2020a). De acordo com a International Agency for Research on Cancer, em 2018 o câncer de mama correspondeu à quinta maior causa de mortalidade por câncer em geral e à maior causa de mortalidade apenas em mulheres no mundo todo, com cerca de 626679 casos (International Agency for Research on Cancer, 2018). O Instituto Nacional de Câncer (INCA) estima para 2020 um total de 66280 casos novos, ou seja, uma taxa de incidência de 43,74 casos por 100000 mulheres (Instituto Nacional de Câncer, 2020b). Na figura 1 é possível observar o crescimento da taxa de mortalidade nos últimos anos.

Figura 1: Taxas de mortalidade por câncer de mama feminina no Brasil, específicas por faixas etárias, por 100.000 mulheres




Fonte: Instituto Nacional de Câncer (2020b)

Desde o surgimento do exame de mamografia convencional através de chapas, um grande problema associado aos seus resultados apareceu: o alto risco de diagnósticos falso-positivos ou falso-negativos.

Diagnósticos falso-positivos podem acontecer por diversos motivos, dentre os quais o principal é a própria característica do tecido mamário que, em alguns casos, por ser muito denso, dificulta a diferenciação do tumor. Suas consequências são o de postergar o tratamento do câncer e o de gerar um falso senso de tranquilidade na paciente. Já os diagnósticos falso-positivos ocorrem normalmente pela sobreposição de estruturas normais nas imagens ou pela presença de outras doenças na mama. Suas consequências são o alto impacto psicológico que recai sobre a paciente. Estudos recentes mostram que tais efeitos alteram a ansiedade da paciente a curto prazo, porém não a longo prazo. Vale ressaltar que não foram mensuráveis a longo prazo decrementos na saúde psicológica e física dos pacientes estudados (Tosteson et al., 2014). Entretanto, tais resultados ainda não são considerados como verdades universais e estudos ainda são feitos a fim de obterem resultados mais concretos.

Desde a década de 1990, foram desenvolvidos sistemas para auxiliar radiologistas a melhorarem a acurácia dos exames mamográficos. Tais sistemas foram denominados CAD - *Computer-Assisted Detection*. Entretanto, os sistemas CAD não geraram melhorias significantes de performance da classificação dos exames e se mantiveram estagnados por mais de décadas (Fenton et al., 2007). Entretanto, atualmente, com o avanço dos estudos em inteligência artificial no uso de técnicas como *deep learning* e a grande quantidade de pesquisas em redes convolucionais para tratamento de imagens, os sistemas CAD tem conseguido obter resultados semelhantes aos dos médicos na detecção de câncer de mama e tem melhorado a performance de radiologistas ao atuar como suporte. (Rodriguez-Ruiz et al., 2019).

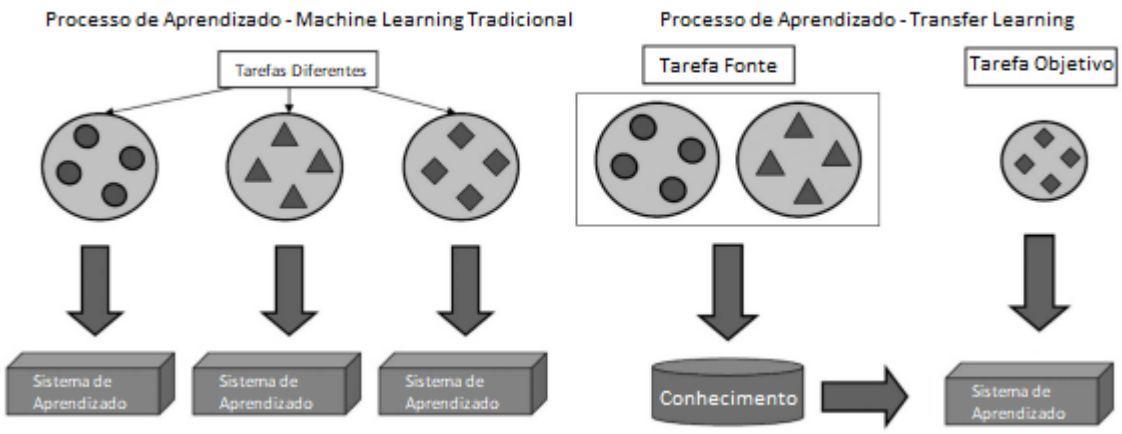
No entanto, no caso da detecção de distorção arquitetural em exames mamográficos, o processo de classificação é mais complicado. Isso se deve ao fato de que para gerar um classificador, existe a necessidade de haver uma grande quantidade de dados disponíveis para treinamento da rede de modo que uma grande quantidade de informações sejam extraídas (Goodfellow, 2016). Infelizmente, é extremamente difícil conseguir um banco de dados de exames de mamografia clínicos laudados por terem acesso restrito (Greenspan et al., 2016), ou por haverem poucos casos positivos disponíveis (Bordás et al., 2004). Logo, estratégias tem sido desenvolvidas para prosseguir no estudo do processamento de imagens mamográficas. Para contornar o problema da falta de dados, comumente se utiliza a técnica de *data augmentation*, que consiste em executar uma série de transformações como rotações ou translações em uma mesma imagem, gerando novas com o objetivo de fazer com que a rede neural entenda que as imagens geradas são diferentes, aumentando, portanto, o tamanho do conjunto de dados. Outra técnica que tem sido estudada para auxiliar nos estudos de processamento de imagens médicas é a transferência de aprendizado ou *transfer learning*.

No mundo real é possível observar diversos exemplos de transferência de aprendizado.

O aprendizado humano é permeado de experiências que auxiliam no aprendizado de outras. Quando se aprende a identificar um copo, por exemplo, há uma maior facilidade para compreender o que é uma xícara. De forma similar, aprender a tocar violão facilita aprender a tocar guitarra ou baixo. O estudo do aprendizado de transferência foi motivado justamente pela naturalidade com que os seres humanos tem de resolver problemas com base em conhecimentos previamente adquiridos, gerando novas soluções mais fáceis ou melhores. Tal motivação foi primeiramente estudada na Conferência e Workshop sobre Sistemas de Processamento de Informação Neural de 1995, com a temática intitulada "Aprendendo para Aprender"(*Learning to Learn*, focada justamente na necessidade de métodos computacionais que retivessem e reutilizassem aprendizados prévios (NIPS*95 Post-Conference Workshop, 1995).

O estudo sobre a transferência de aprendizado progrediu ao longo dos anos até, por fim, possuir o objetivo de extrair conhecimento de uma ou mais tarefas fonte e aplicá-lo em uma tarefa alvo que, normalmente, não apresenta uma grande quantidade de dados bons para treinamento. A figura 2 mostra a diferença entre o aprendizado de máquina tradicional e a proposta da transferência de aprendizado.

Figura 2: Diferentes processos de aprendizado




Fonte: Adaptado de Pan and Yang (2010)

A proposta da utilização do *transfer learning* para classificação de exames mamográficos consiste em treinar um modelo de *deep learning* em uma grande base de dados para alguma tarefa qualquer. Caso tal tarefa esteja relacionada com a radiologia, o modelo utilizará os pesos de características já aprendidas para mais facilmente se adequar ao aprendizado da detecção de tumor na mama. Mesmo que a tarefa não esteja associada com a radiologia, ainda assim a transferência de aprendizado pode ser benéfica. Uma vez que os pesos do modelo já estão inicializados para detectar características como bordas ou

texturas, a adaptação para detecção de uma nova tarefa se torna mais fácil, garantindo menos tempo de treinamento e, possivelmente, até melhoras de performance (Oquab et al., 2014).

## 1.1 Objetivo

O presente trabalho tem como objetivo geral estudar a possibilidade de utilização da transferência de aprendizado, mais especificamente a técnica de extração de características, em redes neurais convolucionais para detecção de distorção arquitetural mamária em imagens mamográficas. O trabalho também tem por objetivo realizar testes em diferentes arquiteturas pré treinadas e observar o comportamento de cada uma delas em diferentes situações, comparando o comportamento delas com uma rede treinada especificamente para essa finalidade.

## 1.2 Estrutura do Trabalho

O presente documento está dividido nos seguintes capítulos descritos abaixo:

- *Conceitos e Teoria:* Nesse capítulo serão apresentadas os conceitos e teoria utilizados para a execução do trabalho. Primeiramente, haverá uma breve exposição dos conceitos de distorção arquitetural, salientando a importância de estudos nessa área. Em maiores detalhes haverá uma explanação sobre o conceito de transferência de aprendizado (*Transfer Learning)*, bem como uma exposição a respeito de redes neurais convolucionais, suas características, tipos de arquitetura e técnicas.

- *Materiais e Métodos:* Nesse capítulo são apresentadas as escolhas feitas para o trabalho com suas respectivas justificativas. Além disso, é apresentada a metodologia empregada para geração do banco de dados e os critérios utilizados para avaliar os resultados obtidos.

- *Resultados:* Nesse capítulo são expostos os resultados obtidos, bem como uma breve discussão sobre o que foi observado.

- *Conclusão:* Nesse capítulo o trabalho é concluído, apresentando mais os resultados de forma condensada, com uma breve discussão e indicação de possíveis trabalhos futuros.

# 2 CONCEITOS E TEORIA

## 2.1 Distorção Arquitetural Mamária

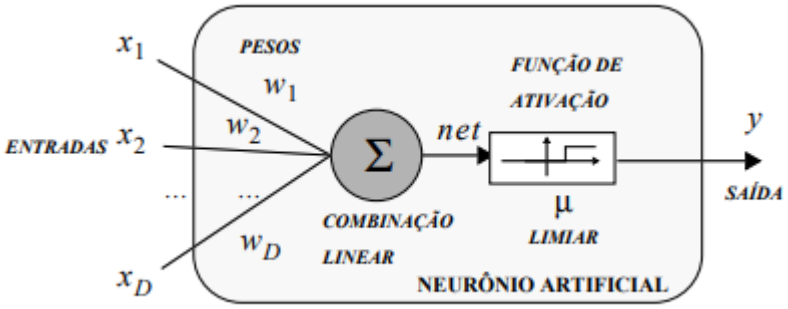
A distorção arquitetural mamária é definida como a distorção na arquitetura normal da mama sem o aumento da densidade da mama e sem necessariamente haver massa associada. Ela tem por característica a presença de linhas finas ou espiculadas, ou seja, linhas que são irradiadas a partir de determinado ponto focal. A distorção arquitetural pode conter o desvio da orientação das fibras que deveriam ir para o mamilo, chamado de retração ou distorção focal, podendo levar a convergência em outro ponto focal ou divergindo deste (of Radiology et al., 2003). Esse tipo de aparição na mama é o terceiro sinal mais comum do câncer de mama e, devido a sua variabilidade e sutileza, podendo até imitar uma aparência normal de tecidos de mama sobrepostos, esse sinal de anormalidade é frequentemente não detectado (Rangayyan et al., 2010). Vale ressaltar que, uma vez que as microcalcificações e nódulos são os casos mais observados na descoberta de câncer em exames radiológicos, muitos sistemas CAD foram desenvolvidos com foco de detecção nesses sinais (Nemoto et al., 2009). Entretanto, por conta disso, muitos sistemas CAD podem ter uma dificuldade maior em encontrar casos de distorção arquitetural, o que torna o estudo de formas de detecção desses casos essencial (Nemoto et al., 2009).

## 2.2 Redes Neurais Convolucionais

Para que os conceitos de redes neurais convolucionais possam ser compreendidos, primeiramente serão introduzidos alguns conceitos relativos à redes neurais artificiais em geral. Primeiramente é importante salientar que a forma como as redes neurais adquirem algum conhecimento é através de um processo adaptativo de observação, ou seja, após a exposição sucessiva de diversos exemplos da tarefa objetivo, há a atualização dos parâmetros da rede. O processo de atualização dos pesos da rede é chamado de retropropagação do erro (do inglês *backpropagation*) e consiste em pegar o erro encontrado na saída da rede neural e repassar esse valor para todas as camadas da rede por meio da sua derivada parcial relativa a cada parâmetro, corrigindo dessa forma os pesos da rede. O intervalo de tempo no qual são expostos todos os exemplos disponíveis para a rede neural e há a atualização dos pesos é conhecido como época, podendo ser necessário várias épocas para a convergência da rede (Silva et al., 2010). Por fim, vale também ressaltar que, estruturalmente, as redes neurais possuem por unidade mais básica os neurônios, como pode ser visto na figura 3. Os neurônios são estruturas que possuem um conjunto de entradas ponderadas pelos chamados pesos sinápticos e que, após uma combinação linear dos valores de entrada, geram um valor que será avaliado por um limiar de ativação. Caso o limiar seja atingido, o neurônio é considerado como relevante e a diferença entre o limiar

de ativação e a combinação linear das entradas é passado por uma função que tem por objetivo limitar os valores de saída do neurônio entre certo intervalo, chamada de função de ativação (Silva et al., 2010).

Figura 3: Neurônio Artificial




Fonte: Rauber (2005)

De forma geral, as redes neurais artificiais possuem uma arquitetura e neurônios e funções de ativação arranjados em uma determinada disposição. A forma de divisão mais básica de uma rede neural é dada pela camada de entrada, onde são recebidos os dados, as camadas escondidas, que são responsáveis por extrair as características relevantes para a rede e, por fim, a camada de saída (Silva et al., 2010). Olhando pela ótica da divisão básica, as redes neurais convolucionais podem ser perfeitamente descritas dessa forma. Nelas há a camada de entrada para receber os dados, seguida por um conjunto de camadas de convolução que é responsável pela extração das características relevantes para determinada tarefa, culminando nas camadas totalmente conectadas, ou camadas densas, que são responsáveis por estabelecer qual característica pertence a qual classe na rede neural.

### 2.2.1 Composição

As redes neurais convolucionais são geralmente utilizadas para a solução de problemas de Visão Computacional, sobretudo classificações de imagens. Ela é um tipo de rede neural que, como brevemente dito anteriormente, apresenta os seguintes componentes: camadas de convolução, camadas de *pooling*, funções de ativação e camadas densas (Ponti, 2017).

Antes de serem explicados os conceitos por trás de cada camada das redes neurais convolucionais, há a necessidade de introduzir o conceito da operação de convolução. A convolução é um operador linear que, a partir de duas funções, gera a soma do produto das funções dadas através da superposição delas em função do deslocamento entre elas. De forma matemática, sejam duas funções $f_1 e f_2 : R \rightarrow R$. Define-se a convolução entre

elas, denotada pelo operador $*$, a partir de um deslocamento $\tau$:

$$(f_1 * f_2)(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau \tag{2.1}$$

ou, de forma contínua, para o tratamento de sinais discretos, tem-se que:

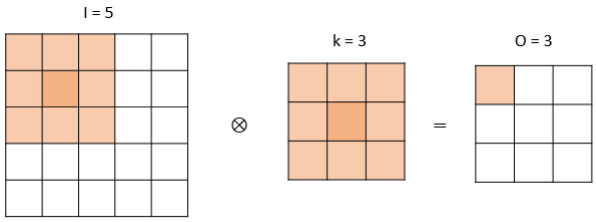
$$(f_1 * f_2)(n) = \sum_{\tau=-\infty} f_1(\tau) f_2(n - \tau) \tag{2.2}$$

A partir desse conceito, pode ser definida a operação de convolução em uma imagem. Sendo uma imagem dada por $I(x, y)$ e a matriz bidimensional que filtrará a imagem dada por $K$ de tamanho $mxn$, tem-se que o processo de convolução é dado por:

$$K(x, y) * I(x, y) = \sum_m \sum_n K(m, n) I(x - m, y - n) \tag{2.3}$$

Tendo ciência dos conceitos sobre a convolução, é possível compreender as camadas de uma rede neural convolucional. As camadas de convolução são compostas de filtros que serão aplicados na matriz de entrada. Cada filtro é uma matriz $n$ x $n$ de pesos $\omega_i$. Cada peso é uma parâmetro a ser aprendido pela rede. Cada filtro aplicado na matriz de entrada faz uma combinação linear dos valores em uma vizinhança na matriz igual ao tamanho do filtro (Ponti, 2017). A figura 4 ilustra o processo de convolução de forma simples.

Figura 4: Exemplo do processo de convolução



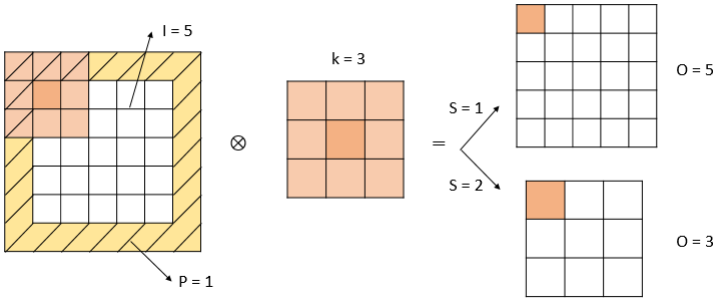

Fonte: Autor

Vale ressaltar que o processo de convolução pode ser afetado pela presença ou não de preenchimento (do inglês, *padding*) na matriz de entrada, ou de passo (do inglês, *stride*) diferente de um. *Padding* é uma técnica que adiciona valores (normalmente zero) a margem da matriz de forma que seu tamanho aumente. Dessa forma, há a capacidade de controlar o tamanho da matriz de saída. Já *stride* é o nome dado ao espaçamento que o seu centro do filtro se move na matriz de entrada. Na figura 4, o *stride* é igual a um, ou seja, o centro do filtro move um pixel de cada vez para formar a matriz de saída. Com essas variáveis adicionadas, é possível calcular o tamanho da matriz de saída em uma camada de convolução a partir da seguinte equação:

$$O = \frac{I + 2P - K}{S} + 1 \tag{2.4}$$

em que O representa o tamanho da matriz de saída, I o tamanho da matriz de entrada, P o tamanho do preenchimento e S o valor do passo do filtro. A figura 5 demonstra o processo de convolução ao serem variados os parâmetros de *stride* e *padding*.

Figura 5: Exemplo do processo de convolução mostrando as variáveis *padding* e *stride*




Fonte: Autor

As camadas de *pooling*, também chamadas de *downsample*, são geralmente aplicadas após algumas camadas convolucionais como objetivo de reduzir espacialmente as dimensões do vetor que está sendo processado pela rede e, consequentemente, diminuir o custo computacional do processamento (Ponti, 2017). A camada de *pooling* mais utilizada é a chamada de *max-pooling* que gera um processo de amostragem pegando o maior valor dentre um conjunto de pixels determinado por um filtro, como demonstra a figura 6.

Figura 6: Exemplo do processo de *max-pooling*

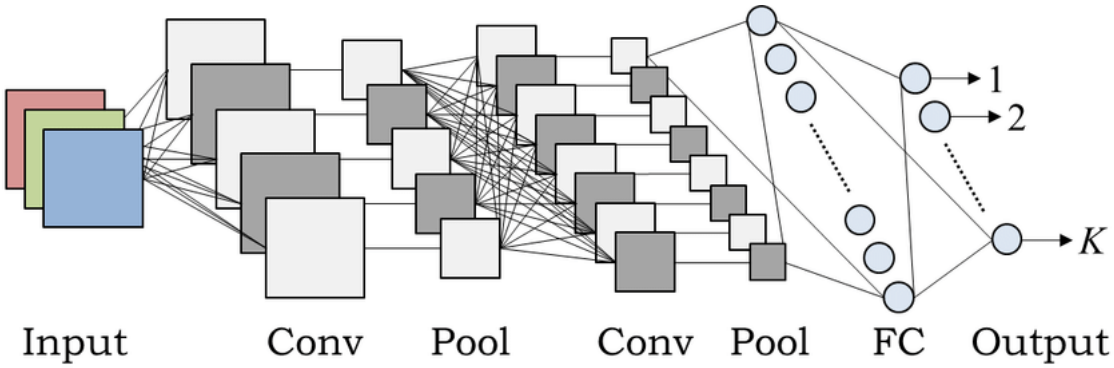

Fonte: Autor

As funções de ativação de um neurônio definem como serão as saídas dos dados. Elas introduzem propriedades não lineares na rede, apresentando por principal objetivo converter um sinal de entrada de um neurônio para um sinal de saída que será utilizado como entrada para o próximo neurônio (Walia, 2017). Existem diversas funções de ativação, dentre as quais pode-se tomar como exemplo algumas funções como a sigmoide, a softmax, a tangente hiperbólica ou a ReLU (Rectified Linear Units).

Por fim, as camadas densas, também chamadas de camadas totalmente conectadas (*fully conected layers*), são camadas que recebem um vetor como entrada e produzem um

escalar como saída, diferentemente das camadas convolucionais que recebem uma matriz na entrada e produzem uma matriz na saída. Para que haja a transição entre uma camada convolucional para uma densa, é feito uma remodelagem da matriz de forma a torna-la um único vetor. Tal tipo de camada é, normalmente, a última camada de uma rede neural convolucional, contendo as probabilidades de classificação(Ponti, 2017) Um exemplo da arquitetura completa de uma rede neural convolucional pode ser visto na figura 7.

Figura 7: Exemplo da Arquitetura de uma Rede Neural Convolucional




Fonte: Hidaka (2017)

### 2.2.2 *Overfitting*

O aprendizado de máquina supervisionado pode ser observado como a tentativa de escolher a melhor função $f$, tal que $y_i \approx f(x_i)$ para a maioria dos valores $i$ em um conjunto de dados $(x_i, y_i)$. Entretanto, na busca pela melhor função, muitas vezes o modelo convolucional pode gerar um $f$ que se adéqua perfeitamente ao modelo mas não funcionará bem em dados novos. Esse problema é conhecido como sobre-ajuste ou *overfitting*. Com a finalidade de minimizar tal problema, existem diversas técnicas que podem ser utilizadas, dentre as quais podem ser citadas as seguintes principais: validação cruzada, parada antecipada e regularização. Cada uma delas será brevemente explicada a seguir.

#### 2.2.2.1 Validação Cruzada

A validação cruzada corresponde a um método para avaliar a capacidade de generalização de um modelo utilizando partição do conjunto de dados. As principais formas de se realizar a partição do conjunto de dados são os métodos *holdout*, *k-fold* e *leave-one-out*.

O método *holdout* consiste em dividir o conjunto de dados totais em duas partes mutuamente exclusivas em qualquer proporção. Comumente essa divisão utiliza 2/3 dos dados para treinamento e 1/3 para testes (chamado de *holdout set*). Considerando que a acurácia de um modelo aumenta com o aumento da quantidade de dados disponível

para treinamento, tem-se um dilema na divisão dos dados. Quanto maior a quantidade de dados no conjunto de teste, maior é o enviesamento do estimador, podendo levar ao caso de sub estimação (*underfitting*). Por outro lado, quanto menor a quantidade de dados no conjunto de teste, maior é o intervalo de confiança da acurácia do modelo. Portanto, essa abordagem normalmente é utilizada quando está disponível uma grande quantidade de dados, sendo, mesmo assim, considerado um método ineficiente ao não utilizar uma parte dos dados para o treinamento do modelo (Kohavi et al., 1995).

O método *k-fold* é responsável por reduzir significativamente a variância do erro de generalização de uma rede neural, se comparado ao método *holdout*, mesmo para pequenos valores k (Kale et al., 2011). O procedimento da técnica inicia particionando o conjunto de dados em k partes mutuamente exclusivas e de aproximadamente mesmo tamanho. Utiliza-se então uma parte para teste e as demais para treino, repetidas vezes, variando sempre qual pasta será utilizada para teste. Por fim, o método *leave-one-out* é uma variação do método *k-fold* no qual k assume o valor da quantidade total de dados (Kohavi et al., 1995). Essa técnica possui um alto custo computacional e temporal, sendo utilizada apenas quando o conjunto de dados disponível é pequeno.

#### 2.2.2.2 Regularização - *Dropout*

A regularização é o nome dado a um conjunto de técnicas que tem por objetivo diminuir a complexidade de um modelo. A técnica que será utilizada para atingir tal objetivo depende do local onde será aplicada (redes neurais, modelos de regressão, árvores de decisão, entre outros). Como o presente trabalho foca seus estudos em redes neurais convolucionais, será apresentado brevemente o conceito de *dropout*.

A técnica de *dropout* é uma técnica que tem por objetivo de reduzir o *overfitting* presente em uma rede através de um método que combina diversas redes, extraindo a média da saída delas.

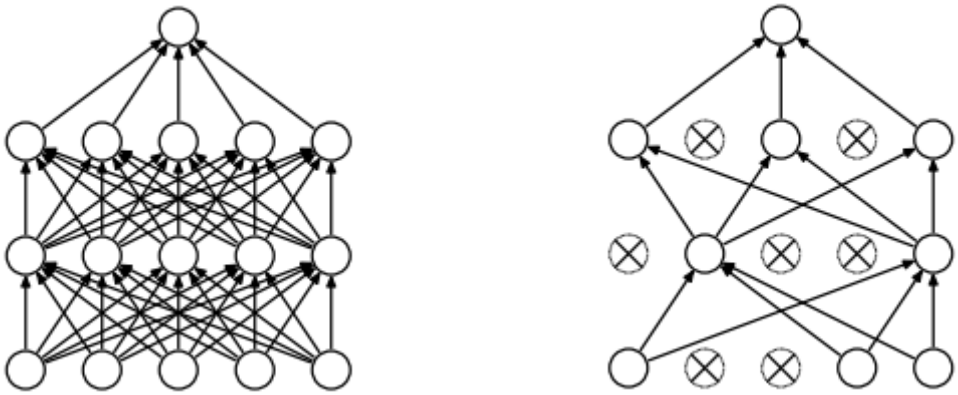
A ideia de combinar diversas redes neurais muito grandes em uma só, até então, era considerada extremamente custosa. Não apenas por que seriam necessários treinar diversas redes neurais com arquiteturas diferentes, como também seria necessário uma enorme quantidade de dados disponível para que cada uma delas fosse treinada em diferentes grupos de dados.

Entretanto, a ideia trazida pela técnica de *dropout* é a de, como indicado pelo nome, não utilizar unidades na rede neural de forma temporária, como exemplifica a figura 8. Para cada unidade presente na rede neural, atribui-se uma probabilidade $p$, que geralmente é escolhida como 0.5, e randomicamente as unidades são retiradas. Dessa forma, durante a fase de treinamento, são geradas redes menores randomicamente. Para uma rede com $n$ unidades, existem $2^n$ redes diferentes que podem ser treinadas compartilhando os pesos. Durante a fase de testes, no entanto, apenas uma rede neural retornará as saídas sem

utilizar *dropout*. Para que seja possível combinar todo o aprendizado em apenas uma rede, as unidades que possuíam probabilidade $p$ durante a etapa de treinamento multiplicam seus pesos por $p$ na etapa de teste. Dessa forma, as saídas esperadas das unidades que utilizaram *dropout* serão as mesmas durante a fase de testes, já sem a utilização da técnica (Srivastava et al., 2014).

Figura 8: *Dropout*

(a) Rede Neural Padrão (b) Rede Neural com Dropout



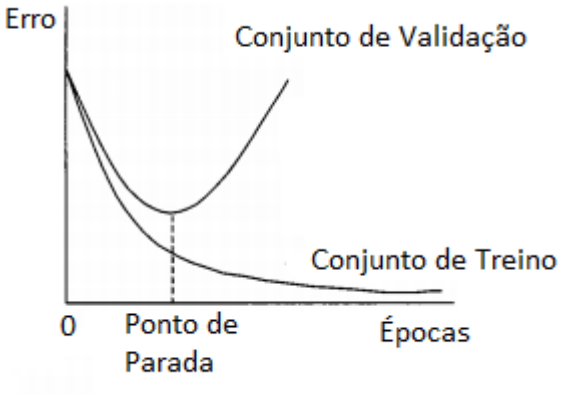
Fonte: Srivastava et al. (2014)

#### 2.2.2.3 Parada Antecipada

Durante a etapa de treinamento de uma rede neural chegará um momento em que o modelo deixará de generalizar e começará a aprender apenas ruído do conjunto de dados de treino, perdendo assim sua capacidade de generalização. Esse erro, como já abordado em seções anteriores, é o chamado erro de *overfitting*. O grande desafio, portanto, é treinar uma rede neural tempo suficiente para que ela consiga generalizar a ideia por trás do conjunto de dados, mas não treinar por tanto tempo permitindo que ela sofra o *overfitting*.

Uma forma de controlar o treinamento de um modelo é através do monitoramento do erro do conjunto de validação. Normalmente, antes de uma rede perder sua capacidade de generalização, a perda do conjunto de validação decresce antes de começar a subir (Bishop, 2006), como exemplifica a figura 9. Logo, parar o treinamento no ponto de menor erro da validação é desejado. Esse procedimento é chamado de parada antecipada (do inglês *early stopping*). Essa técnica é pode ser compreendida como um tipo implícito de regularização (Zhang et al., 2016).

Figura 9: Parada Antecipada




Fonte: Adaptado de Gençay and Qi (2001)

### 2.2.3 Arquiteturas

A arquitetura de uma rede neural refere-se à disposição dos neurônios, um em relação ao outro. Normalmente, utiliza-se a palavra arquitetura também para se referir a topologia de uma rede neural, que consiste nas diferentes composições estruturais possíveis de serem realizadas com diferentes quantidades de neurônios nas camadas de entrada, intermediária e de saída da rede.

Com o passar dos anos, no estudo das redes neurais, algumas arquiteturas se tornaram mais conhecidas, como por exemplo as redes *FeedForward* de Camada Simples como a rede perceptron, as redes *FeedForward* de Camadas Múltiplas como a rede perceptron multicamadas ou as redes realimentadas como a rede Hopfield. No estudo das redes neurais profundas, algumas arquiteturas também se tornaram conhecidas com o avançar das pesquisas. Alguns exemplos de arquiteturas profundas serão brevemente mostrados a seguir.

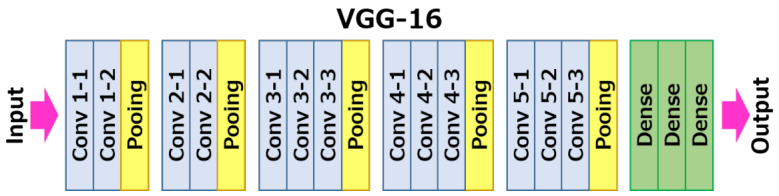
#### 2.2.3.1 VGG Net

Essa configuração de rede surgiu e se tornou importante na classificação de imagens através do desafio ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*), competição anual iniciada em 2010 na qual algoritmos competem pelo melhor desempenho na classificação e detecção de objetos e cenas (Russakovsky et al., 2015).

O modelo VGG competiu no desafio ILSVRC em 2014 e ficou conhecida por sua simplicidade. No entanto, tal configuração tem por problema seu tamanho, que no modelo pré treinado chega a mais de 500MB. Sua premissa básica é a de utilizar filtros de convolução de tamanho 3x3, em contrapartida aos filtros da rede AlexNet de tamanho 11x11, mantendo mais informações nas primeiras camadas. Outro efeito da diminuição do

tamanho dos filtros de convolução é o do aumento no número de camadas convolucionais, aumentando a profundidade da rede para 16 a 19 camadas (Simonyan and Zisserman, 2014). A arquitetura da rede VGG16 pode ser vista na figura 10.

Figura 10: Arquitetura da rede VGG 16

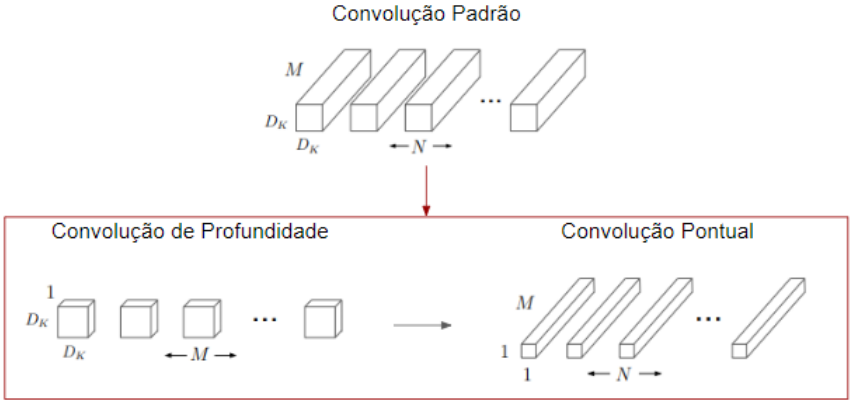



Fonte: Hassan (2018)

#### 2.2.3.2 MobileNet

O modelo *MobileNet* foi desenvolvido para ser uma rede neural pequena, rápida e, como sugere o nome, com capacidade de ser integrada em dispositivos móveis como em aplicativos de celulares. O modelo da rede é baseado em convoluções separáveis em profundidade (*Depthwise Separable Convolution*) para diminuir a complexidade. Em uma arquitetura de rede neural, uma camada de convolução padrão filtra e combina as entradas em apenas uma etapa de forma a gerar saídas. No entanto, a *MobileNet* utiliza uma convolução de forma fatorizada, sendo seus fatores uma convolução de profundidade seguida de uma convolução 1x1 chamada de convolução pontual. A camada de convolução de profundidade filtra as entradas e a camada de convolução pontual combina as entradas. A figura 11 ilustra esse processo.

Figura 11: Convoluções Separáveis em Profundidade

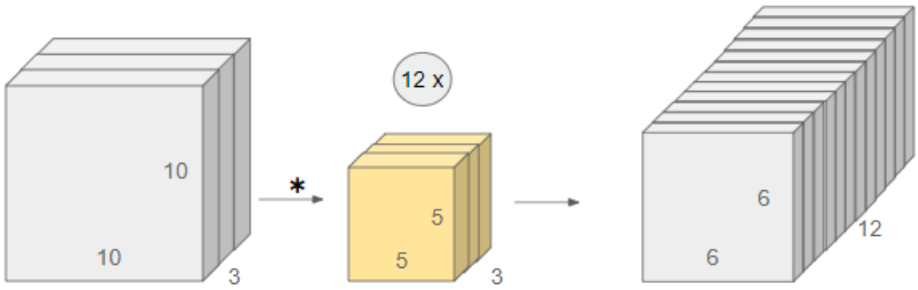



Fonte: Adaptado de Howard et al. (2017)

Para que possa ser compreendido a redução do esforço computacional, observe um exemplo simples. Suponha uma imagem RGB de tamanho 10x10x3, através da qual

haverá a convolução com 12 filtros 5x5x3 sem preenchimento e com passo igual a um. De acordo com o que foi explicado anteriormente sobre convolução, o tamanho da imagem de saída será de $10 - 5 + 1 = 6$, como mostrado pela equação 2.4. Como também descrito na seção sobre redes neurais convolucionais, o processo de convolução de imagens pode ser compreendido como a movimentação do filtro sobre a imagem para gerar o valor de cada pixel de saída. Logo, 12 filtros de tamanho 5x5x3 se moverão 6x6 vezes, gerando um total de multiplicações em uma convolução padrão de 12x5x5x3x6x6 = 32400 para gerar uma saída 6x6x12, como ilustra a figura 12.

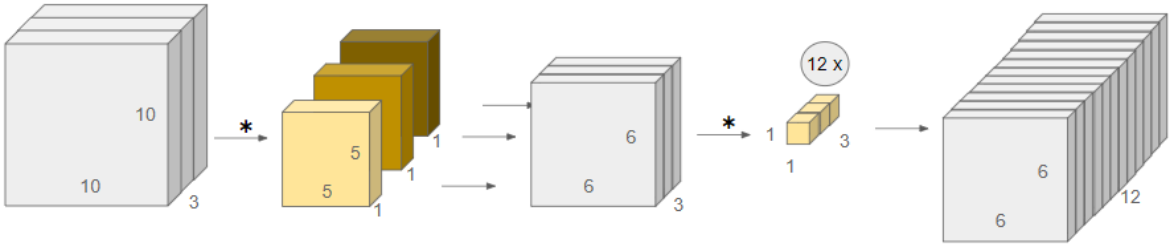
Figura 12: Convolução padrão exemplo




Fonte: Autor

Já para o processo de convolução separável em profundidade da rede *MobileNet*, primeiramente faz-se a convolução sem profundidade, ou seja, 3 filtros 5x5x1 se movem 6x6 vezes através de cada canal, gerando 3x5x5x6x6 = 2700 multiplicações e uma saída de 6x6x3. Então na convolução pontual, 12 filtros 1x1x3 se movem 6x6 vezes na matriz de saída 6x6x3 da convolução sem profundidade, gerando 12x1x1x3x6x6 = 1.296 multiplicações para, por fim, gerar a mesma saída 6x6x12, como ilustra a figura 13. No total, para o processo de profundidade separável são feitas 2700+1296 = 3.996 multiplicações, o que representa cerca de aproximadamente 12% do número de multiplicações em uma convolução normal, demonstrando como as arquiteturas MobileNet são capazes de reduzir imensamente o processamento de uma rede neural.

Figura 13: Processo de convolução separável em profundidade exemplo




Fonte: Autor

#### 2.2.3.3 ResNet

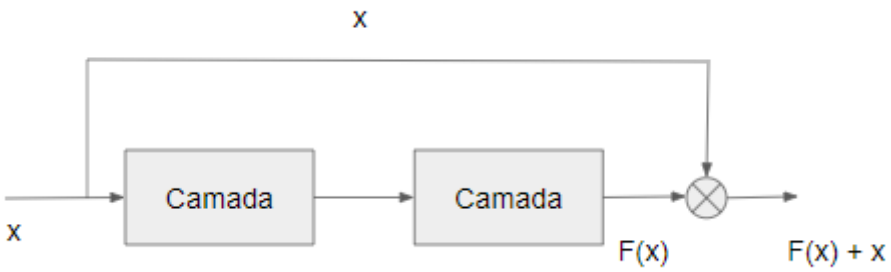
Com o avançar das pesquisas em redes neurais, estudos foram feitos focados em entender mais a respeito da relação entre a profundidade da rede e o seu resultado. Tais estudos, no entanto, observaram que quanto mais profunda a rede, surgiam os seguintes problemas: primeiramente a acurácia se tornava saturada e depois se degradava (He et al., 2016). A acurácia saturar é intuitivamente compreendido como um ponto em que o poder de modelagem do problema atingiu seu ápice no tratamento com o dado fornecido. Quanto a queda da acurácia da rede após esse ponto, primeiramente entendia-se que era devido ao problema de *overfitting*. Porém, foi observado que, de forma contraintuitiva, ao continuar adicionando camadas na rede, o erro durante o treinamento foi aumentando, descartando a possibilidade do problema se caracterizar como *overfitting*.

Para resolver esse problema, He et al. (2016) desenvolve a ideia de que, ao invés de esperar que as camadas empilhadas se moldem ao problema, talvez fosse mais eficaz que as camadas se ajustem ao resíduo.

Uma vez que as redes Neurais são bons estimadores de função, elas deveriam facilmente aprender a função identidade, na qual a saída de uma função se tornaria a própria entrada, ou seja, $f(x) = x$. Seguindo a mesma lógica, caso seja ligado diretamente a entrada da primeira camada de um modelo para a saída da última camada, a rede deveria ser capaz de predizer qualquer função que ela estivesse aprendendo juntamente com a entrada somada a ela, ou seja, $f(x) + x = h(x)$. De forma intuitiva, espera-se que a rede seja capaz de predizer f(x) = 0 facilmente.

Seguindo essa lógica, criou-se o conceito de pular camadas demonstrado na figura 14, com o objetivo de que camadas superiores aprendam a função identidade e tenham sua performance pelo menos igual as camadas inferiores. Isso fez com que o problema do gradiente fosse mitigado, gerando assim as redes residuais.

Figura 14: Pulo de Conexões




Fonte: Autor

## 2.3 Transferência de Aprendizado (*Transfer Learning*)

Como visto no capítulo anterior, transferência de aprendizado corresponde a se aproveitar de uma tarefa previamente assimilada para, de alguma forma, facilitar o aprendizado de outra tarefa. Para que seja feita uma definição mais formal sobre o tema, no entanto, é necessário serem definidos os conceitos de domínio e tarefa. De acordo com Pan and Yang (2010), um domínio apresenta dois componentes: um espaço de característica e uma distribuição de probabilidade marginal. A notação utilizada será como abaixo:

$$\begin{aligned} &\chi \rightarrow \text{Espaço de Característica} \\ &P(X) \rightarrow \text{Distribuição de Probabilidade Marginal} \\ &X = \{x_1, x_2, ..., x_n\} \in \chi \\ &\text{Portanto,} \quad D = \{\chi, P(X)\} \end{aligned} \tag{2.5}$$

Uma forma de compreender a definição de domínio pode ser exemplificando um caso em que deseja-se classificar algo binariamente. Nesse caso, $\chi$ seria o espaço de todos os vetores de termos, $x$ é o n-ésimo termo de um vetor e X é uma amostra particular de aprendizado, um *batch*. Dada a definição de domínio, é possível definir tarefa. Similarmente, uma tarefa é composta de dois elementos: um espaço de rótulos e uma função objetiva de predição.

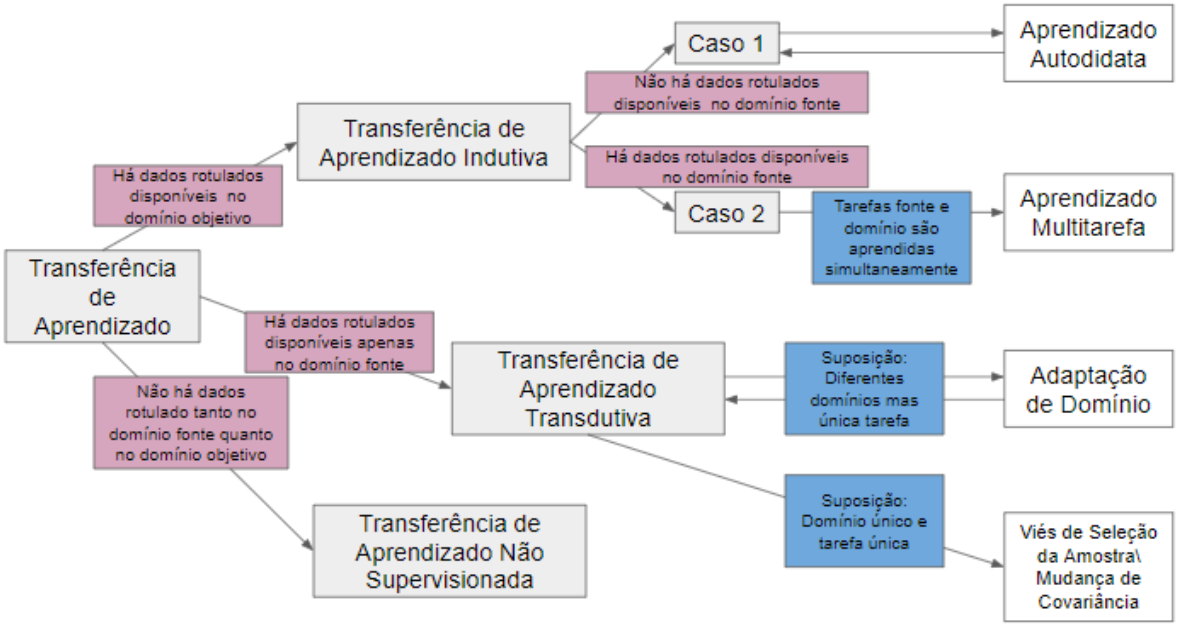
$$\begin{aligned} &\gamma \rightarrow \text{Espaço de Rótulos} \\ &f(.) \rightarrow \text{Função Objetiva de Predição} \\ &\text{Portanto,} \quad \tau = \{\gamma, f(.)\} \end{aligned} \tag{2.6}$$

A tarefa não é algo observável, mas sim algo que pode ser aprendido através dos dados de treinamento, que consistem em $\{x_i, y_i\}$, sendo $x_i \in \chi$ e $y_i \in \gamma$, ou seja, uma amostra e seu rótulo, que em um exemplo de classificação pode ser visto como "Verdadeiro"ou "Falso". A função $f(.)$ pode ser utilizada como modo de predizer o rótulo correspondente a uma amostra, $f(x)$, com x sendo um novo dado de entrada.

Enfim é possível definir a transferência de aprendizado. Dado um domínio $D_s$ e uma tarefa de aprendizado $\tau_s$, bem como um domínio objetivo $D_t$ e uma tarefa de aprendizado $\tau_t$, a transferência de aprendizado tem por alvo melhorar o aprendizado da função de predição objetivo $f_t(.)$ em $D_t$ utilizando o conhecimento de $D_s$ e $T_s$, sendo $D_s \neq D_t$ ou $T_s \neq T_t$ (Lin and Jung, 2017).

Tendo ciência do conceito de transferência de aprendizado, é possível categorizar a técnica de acordo com a disponibilidade dos dados da rede, o domínio e as tarefas, como pode ser visto de acordo com a figura 15.

Figura 15: Diferentes tipos de Transferência de Aprendizado




Fonte: Adaptado de Pan and Yang (2010)

Os métodos de transferência de aprendizado podem, portanto, serem categorizados de acordo com o tipo de algoritmo envolvido:

- *Transferência de Aprendizado Indutiva:* Caso em que, apesar do domínio fonte ($D_S$) e o domínio objetivo ($D_T$) serem os mesmos, as tarefas fonte ($T_S$) e objetivo ($T_T$) são diferentes. Como o domínio é o mesmo, o algoritmo tenta utilizar *bias* indutivos do domínio fonte com o objetivo de melhorar a tarefa objetivo.

- *Transferência de Aprendizado Transdutiva:* Caso em que, de forma contrária ao item anterior, as tarefas fonte ($T_S$) e objetivo ($T_T$) possuem alguma similaridade, mas os domínios fonte ($D_S$) e objetivo ($D_T$) são diferentes. Nessa configuração, o domínio fonte possui dados rotulados enquanto o domínio objetivo não possui nenhum.

- *Transferência de Aprendizado Não Supervisionada:* Esse caso possui configuração similar ao indutivo, possuindo domínio fonte ($D_S$) e objetivo ($D_T$) similares, porém com as tarefas fonte ($T_S$) e objetivo ($T_T$) diferentes. Nesse caso, no entanto, não existem dados rotulados em nenhum dos domínios e o foco é na tarefa não supervisionada no domínio objetivo.

Do ponto de vista prático para imagens, normalmente são utilizadas duas técnicas quando se deseja utilizar transferência de aprendizado: Extração de Características e *Fine Tuning*.

*Extração de Características em Modelos Pré Treinados*

Os modelos extratores de característica utilizam redes neurais profundas pré treinadas com seus pesos congelados. A saída de tais redes são então conectadas em outros modelos que, geralmente, são modelos compostos de camadas densas (*fully connected models*) ou modelos clássicos como máquinas de vetores de suporte (*suport vector machine* - SVM) ou arvores aleatórias (*random forest* - RF).

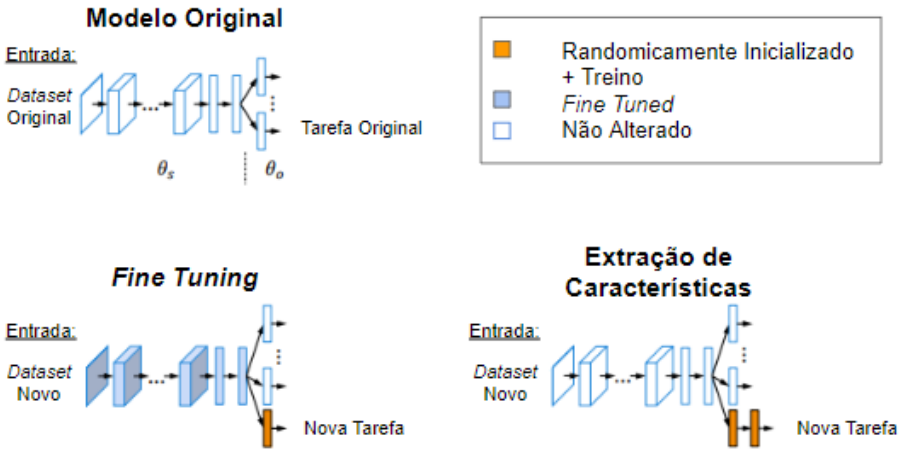
Tais modelos não modificam a rede neural original, permitindo com que as novas tarefas a serem aprendidas se beneficiem das características aprendidas durante as tarefas anteriores. O que caracteriza o uso dessa técnica é que, independentemente de qual parte for mantida da rede neural profunda pré treinada, ela não será modificada. Já as partes que serão acrescidas, sejam elas apenas as camadas densas ou até mesmo camadas convolucionais, serão treinadas com seus pesos inicializados randomicamente.

*Fine Tuning em Modelos Pré Treinados*

Diferentemente do extrator de características, os modelos que utilizam *fine tuning*, como implícito pelo próprio nome, realizam um ajuste fino dos parâmetros existentes na rede convolucional profunda pré treinada. Normalmente, a camada de saída da rede é inicializada com pesos randômicos para aprender a nova tarefa e uma de taxa de aprendizado pequena para minimizar a perda. Dependendo dos hiper-parâmetros utilizados, a rede treinada com *fine tuning* pode ter uma performance melhor não só apenas do que um extrator de características como também de uma rede treinada especificamente para a mesma tarefa.

O que caracteriza a utilização dessa técnica é o fato de que os pesos anteriormente utilizados serão alterados de modo a adaptarem melhor a nova tarefa a rede. A diferença entre extração de características e *fine tuning* pode ser resumida como na imagem 16.

Figura 16: Diferença entre *Fine Tuning* e Extração de Característica




Fonte: Adaptado de Li and Hoiem (2017)

# 3 MATERIAIS E MÉTODOS

## 3.1 Hardware e Configurações

O Computador utilizado para a execução de todos os códigos feitos durante os testes possui um processador "Ryzen 5 1600", 16GB de memória RAM, placa de vídeo "NVIDIA GeForce RTX 2060"e sistema operacional "Windows 10". Todos os códigos feitos utilizaram a placa de vídeo para que pudessem ser executados, se valendo das configurações do Cuda 10.1, Cudnn 7.6.4. O principal fator limitante na execução do trabalho foi a baixa quantidade de memória RAM, que fez com que a geração das imagens através do *data augmentation*, bem como a subsequente geração dos arquivos HDF5 a serem carregados nas redes neurais, tivesse que ser feita por partes, demorando mais do que o esperado.

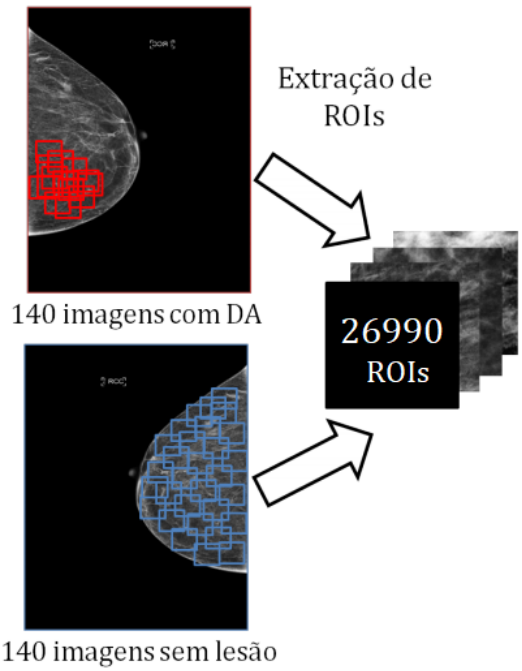
## 3.2 Banco de Dados

O presente trabalho contou com exames clínicos de mamografia digital (FFDM) anônimos obtidos em colaboração com médicos radiologistas do Instituto de Radiologia do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo (InRad/HCFMUSP), utilizando o mamógrafo Selenia Dimensions da Hologic. Para a utilização de tais imagens, foi obtido aprovação do Comitê de Ética em Pesquisa da Faculdade de Medicina da USP e da Plataforma Brasil (Parecer nº 1.581.220) para a aquisição de imagens mamográficas clínicas e dos laudos correspondentes.

O banco de dados utilizado para os testes feitos foi uma adaptação do banco gerado no trabalho feito por Costa (2019). Segue uma breve explicação de como o banco de dados foi gerado originalmente pelo autor supracitado: primeiramente foram selecionados 280 imagens, com cortes craniocaudal ou mediolateral-oblíquo, dentre as quais 140 apresentavam um laudo de distorção arquitetural e as outras 140 com a ausência desta lesão. Cada uma das imagens dos exames possui dimensão de 4096x3328 ou 3328x2560 *pixels* de acordo com o tamanho estabelecido pelo sistema de medição. Vale ressaltar também que cada tamanho de pixel é de aproximadamente $70\mu m^2$ e quantização em 12 bits. Em seguida, Costa decidiu por extrair dos exames as regiões de interesse (ROIs) de cada uma das classes. Para a classe das imagens com DA, definiu-se que os recortes deveriam conter a coordenada central da lesão. Já para a classe das imagens sem distorção arquitetural mamária, os recortes foram feitos em diversos locais diferentes. Utilizando os conceitos de *data augmentation*, foram geradas cerca de 100 recortes por exame utilizando varredura em janela sobre a imagem com passo de 25 pixels, correspondendo ao aumento de amostras por translação. Os recortes tiveram o tamanho fixo de 256x256 *pixels* por motivo de ser uma dimensão média das lesões marcadas. O total de imagens geradas pelo autor foi de

26990. A figura 17 ilustra esse processo de geração dos recortes.

Figura 17: Composição do banco de dados original




Fonte: Adaptado de Costa (2019)

O banco de dados utilizado no presente trabalho utilizou por base o banco de dados citado anteriormente. Foram feitos testes com as arquiteturas utilizando duas abordagens distintas. A primeira abordagem foi a de utilizar o banco de dados contendo as 26990 imagens de forma inalterada e a segunda foi a de utilizar *data augmentation* para generalizar as predições feitas, aumentando a quantidade de imagens consideravelmente. Para cada imagem presente no *dataset* original, foram geradas outras 9 imagens utilizando as seguintes transformações: espelhamento vertical, rotação nos ângulos 90º, 180º e 270º e adição de ruído gaussiano com média 0 e variância 0.04 nas imagens anteriores. Vale ressaltar que, como foram utilizadas redes pré-treinadas no banco de dados *ImageNet*, o tamanho padrão das imagens deveria ser 224x224x3. Portanto, foram feitos recortes centralizados em todas as imagens que originalmente possuíam tamanho 256x256 para que pudessem ser utilizadas. Foram também copiadas 3 vezes a mesma imagem para que pudessem ser simulados os canais RGB exigidos pela entrada da rede. O total de imagens do conjunto com *data augmentation* foi de 269900. Tendo em vista que as imagens do banco de dados original estavam separadas em 10 pastas diferentes, foi utilizado o método *k-fold*. Nas seções de Resultados e Conclusão, o conjunto de imagens inalterado será referido como DA ou conjunto de dados original, e o conjunto de imagens com a utilização de *data augmentation* será referido como DA+ ou conjunto de dados aumentado.

## 3.3 Método Proposto

### 3.3.1 Configuração das Redes Neurais

Com relação ao processo de configuração e treinamento das CNN, segue-se como foi estabelecido cada passo.

#### 3.3.1.1 Escolha das Arquiteturas

Com objetivo de fazer um trabalho comparativo entre as arquiteturas, foi feita uma avaliação dentre as redes pré treinadas disponíveis no site do *keras*. As redes disponíveis se encontram na tabela 1.

Tabela 1: Arquiteturas Pré Treinadas no Banco de Dados Imagenet

| Arquiteturas | Tamanho de Entrada | Parâmetros | Profundidade |
|---|---|---|---|
| Xception | 299x299 | 22.910.480 | 126 |
| VGG16 | 224x224 | 138.357.544 | 23 |
| VGG19 | 224x224 | 143.667.240 | 26 |
| ResNet50 | 224x224 | 25,636,712 | - |
| ResNet101 | 224x224 | 44,707,176 | - |
| ResNet152 | 224x224 | 60,419,944 | - |
| Inception | 299x299 | 23,851,784 | 159 |
| InceptionResNet | 299x299 | 55,873,736 | 572 |
| MobileNet | 224x224 | 4,253,864 | 88 |
| DenseNet121 | 224x224 | 8,062,504 | 121 |
| DenseNet169 | 224x224 | 14,307,880 | 169 |
| DenseNet201 | 224x224 | 20,242,984 | 201 |
| NASNetLarge | 224x224 | 88,949,818 | - |
| NASNetMobile | 331x331 | 5,326,716 | - |

As imagens do banco de dados das ROIs centralizadas possuem o tamanho de 255x255. Portanto, foram excluídas as possibilidades de arquitetura que necessitam de tamanho maior do que esse. Dentre as escolhas de tamanho 224x224, foram selecionadas as redes VGG16, Mobilenet e ResNet50. A justificativa para essas escolhas se baseia na diferença grande que existe entre cada uma das redes, como apresentado na seção de desenvolvimento teórico.

#### 3.3.1.2 Quantidade de Camadas Densas

Não existe nenhum método para determinar a quantidade de camadas densas que uma determinada rede deve possuir. Normalmente, observa-se a quantidade de dados para treinamento, bem como a complexidade da informação que se deseja classificar. Para aplicações em geral, comunidades de cientistas de dados online como *Kaggle* sugerem o

uso de 2 camadas. Nesse trabalho, no entanto, optou-se por testar a utilização de uma até quatro camadas densas.

#### 3.3.1.3 Função de Ativação

Para função de ativação da ultima camada densa foi escolhida a função *softmax*, tendo em vista que o resultado almejado para a saída da rede foi a probabilidade de cada imagem pertencer a determinada classe (imagem com ou sem distorção arquitetural), ao invés da predição de qual classe pertence a imagem, para que as análises de desempenho utilizando as curvas ROC pudessem ser feitas.

#### 3.3.1.4 Função de Custo

A função de custo escolhida para os testes foi a função *categorical cross-entropy*. Para o entendimento dessa escolha há a necessidade de apresentar o conceito da função *cross-entropy* ou $CE$. A função $CE$ é definida pela seguinte equação:

$$CE = -\sum_{i}^{C} t_i \log\left(s_i\right), \tag{3.1}$$

em que $s_i$ e $t_i$ são o valor probabilístico dado pela CNN e a classificação real, respectivamente, para cada dado $i$ pertencente a $C$. No caso da rede proposta, $C = 2$ tornando a função um classificador binário. Para classificadores, normalmente as funções de ativação da ultima camada densa são ou a função *softmax* ou a função *sigmoid* cujas definições podem ser vistas a seguir:

$$F\left(X_i\right) = \frac{\operatorname{Exp}\left(X_i\right) \quad i = 0, 1, 2, \ldots k}{\sum_{j=0}^{k} \operatorname{Exp}\left(X_j\right)} \quad \text{Função Softmax} \tag{3.2}$$

$$F\left(X_i\right) = \frac{1}{1 + \operatorname{Exp}\left(-X_i\right)} \quad \text{Função Sigmoid} \tag{3.3}$$

Para quaisquer valores de quantidade de classe $C$, a vantagem da utilização da função *softmax* é que sua saída terá um valor limitado entre 0 e 1. Ou seja, a soma de todas as probabilidades das $C$ classes será igual a 1. Essa propriedade não necessariamente é observada na função *sigmoid*, apesar de seu valor de saída estar entre 0 e 1.

Como consequência, a função *softmax* sempre é utilizada para problemas com múltiplas classes exclusivas, enquanto a função *sigmoid* é utilizada em exemplos com múltiplos rótulos, no qual um rótulo pode pertencer a mais de uma classe. Em termos práticos, a função *softmax* gera a probabilidade das classes enquanto a função *sigmoid* gera algo similar a uma distribuição marginal de probabilidades.

A soma de uma função *softmax* com uma função *cross-entropy* gera a função denominada de *categorical cross-entropy* que foi a função escolhida para a realização dos testes.

#### 3.3.1.5 Algoritmo de Otimização

O algoritmo de otimização escolhido para treinar a rede foi o Adam. Esse algoritmo é conhecido por ser um estimador adaptativo de momento, sendo utilizado como opção ao *gradient descent*. Sua vantagem vem de sua eficiência computacional ao utilizar pouca memória, sendo recomendado para situações em que há muito dado a ser processado, sendo essas as razões que o levaram a ser escolhido para este trabalho.

A seguir tem-se uma exposição sobre o funcionamento do algoritmo conforme proposto por Kingma and Ba (2014). O objetivo da operação é o de minimizar o valor esperado $E[f(\theta)]$ de uma função escalar estocástica *f(θ)* diferenciável em $\theta$. De acordo com Kingma and Ba (2014), a estocasticidade da função pode estar associada ao fato de que o modelo pode estar avaliando pacotes de amostras aleatórias ou pode surgir como ruído inerente à função.

O algoritmo Adam atualiza as médias móveis do gradiente da função ($m_t$) bem como do gradiente quadrado ($v_t$), sendo tais variáveis avaliadas como estimativas para o 1º e 2º momentos do gradiente (Kingma and Ba, 2014).

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1)\, g_t \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2)\, g_t^2 \end{aligned} \tag{3.4}$$

Os parâmetros $\beta_1, \beta_2 \in [0, 1)$ tem por valores padrão 0.9 e 0.999 respectivamente e, juntamente com a taxa de aprendizado $lr = 0.001$, não costumam ser modificados por serem considerados robustos para a maioria das aplicações (Goodfellow, 2016). Considerando as estimativas como verdade, tem-se que a seguinte expressão também deve ser verdade:

$$\begin{aligned} E\left[m_t\right] &= E\left[g_t\right] \\ E\left[v_t\right] &= E\left[g_t^2\right] \end{aligned} \tag{3.5}$$

Tal propriedade no entanto não é verificada uma vez que as médias móveis são inicializadas com valores 0, levando as estimativas de momento a tenderem para 0. Para corrigir isso, Kingma and Ba (2014) propõe uma correção na inicialização dos momentos, de tal forma que:

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1-\beta_1} \\ \hat{v}_t &= \frac{v_t}{1-\beta_2} \end{aligned} \tag{3.6}$$

Resolvido o problema dos momentos tendendo a 0, tem-se que a atualização dos pesos é dada por:

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.7}$$

sendo $w$ os pesos e $\eta$ o valor de passo (*step size*).

## 3.4 Critério de Avaliação

Para avaliar o trabalho serão utilizados a curva característica de operação do receptor (ROC) e sua área (AUC), bem como o teste pareado de Wilcoxon para determinar se existe diferença significativa entre os modelos propostos. Um breve detalhamento sobre tais técnicas será feito abaixo.

### 3.4.1 Curva Característica de Operação do Receptor (ROC)

Para que a curva ROC possa ser compreendida, primeiramente serão apresentados os conceitos de sensibilidade e especificidade utilizando uma abordagem relacionada a exames médicos. Considere a relação existente entre a doença em um paciente com os resultados diagnosticados em um exame, como mostrado na tabela 2.

Tabela 2: Relação entre Doença e Teste

| | | Doença | |
|---|---|---|---|
| | | Positivo | Negativo |
| Teste | Positivo | Verdadeiro Positivo (VP) | Falso Positivo (FP) |
| | Negativo | Falso Negativo (FN) | Verdadeiro Negativo (VN) |

Fonte: Kawamura (2002)

A sensibilidade (s) é definida como a probabilidade de um indivíduo avaliado e doente de ter seu teste avaliado como positivo para a doença. Portanto, numericamente ela é dada pela quantidade de pessoas doentes e com teste positivo dividida pelo número total de indivíduos doentes em um grupo.

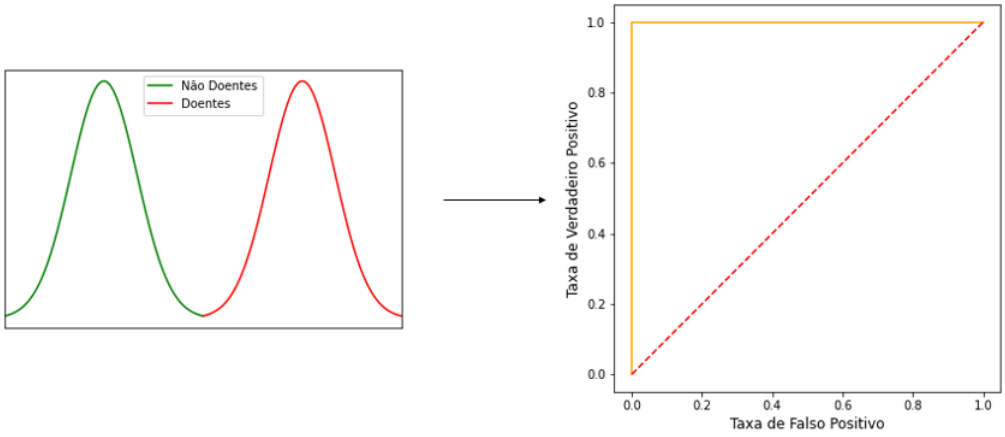
$$s = \frac{VP}{VP + FN} \tag{3.8}$$

A especificidade (e), ao contrário, é definida como a probabilidade de um indivíduo avaliado e saudável de ter seu teste avaliado como negativo para a doença. Numericamente ela é dada pela quantidade de pessoas saudáveis e com teste negativo dividida pelo número total de indivíduos saudáveis em uma grupo.

$$e = \frac{VN}{VN + FP} \tag{3.9}$$

A curva ROC mostra o quão bem um modelo pode distinguir duas classes. Para construi-la, traça-se um diagrama que representa a taxa de verdadeiros positivos, ou sensibilidade (s), em função da taxa de falsos positivos (1 - e). As figuras a seguir mostram as situações em que um modelo pode diferenciar dois grupos.

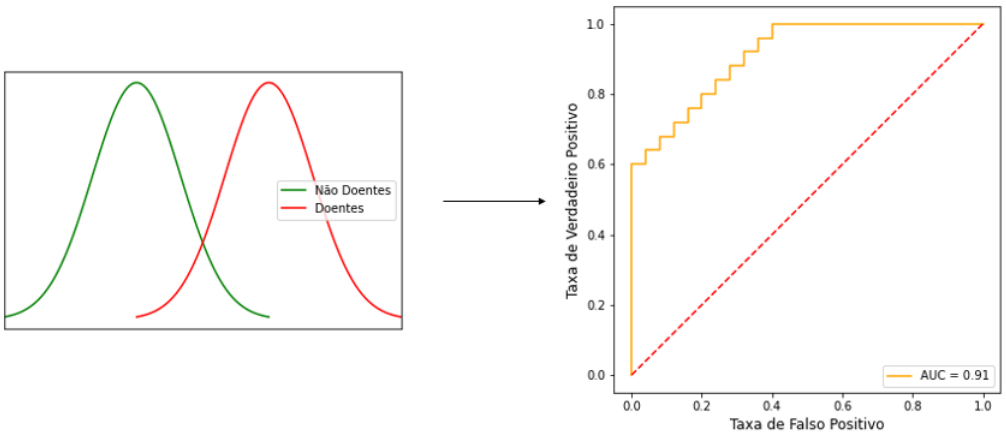
Figura 18: Curva ROC para um modelo perfeito




Fonte: Autor

Quando, idealmente, um modelo é capaz de diferenciar completamente os casos em que um paciente está ou não está com determinada doença, ele gera uma curva ROC como mostra a figura 18. Entretanto, para conjuntos de dados complexos como os exames de mamografia, a diferenciação dos dados não é obtida facilmente e, portanto, a curva ROC raramente será perfeita. Para os casos mais comuns, sempre haverá uma parcela do conjunto de dados observados na qual o modelo não saberá perfeitamente classificar. A curva ROC gerada nesses casos apresenta um formato que se aproxima de uma parábola que não encosta no canto superior esquerdo do gráfico, como mostra a figura 19.

Figura 19: Curva ROC para um modelo que não sabe diferenciar perfeitamente todos os casos

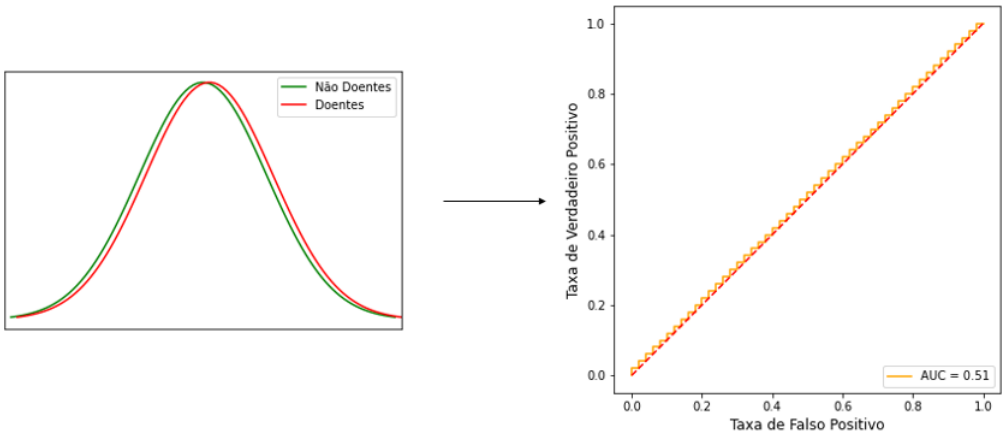



Fonte: Autor

Por fim, se um modelo treinado não for capaz de classificar um determinado grupo,

enxergando-o com iguais probabilidades de pertencer a ambas classes classificadas, sua curva ROC se apresenta como uma reta diagonal como mostra a figura 20, sendo esse o pior caso possível.

Figura 20: Curva ROC para um modelo que não sabe diferenciar nenhum caso




Fonte: Autor

Vale ressaltar que existem os casos em que o modelo pode aprender a classificar os dados de forma errada. Nesses casos, a curva gerada pelo modelo vai se assemelhando a uma parábola com a concavidade apontada para a diagonal esquerda superior. Quando o modelo classifica 100% dos dados de forma equivocada, trocando todos os rótulos, há a geração de uma curva similar a curva mostrada na figura 18, porém encostando no canto inferior direito.

### 3.4.2 Teste de Wilcoxon Pareado

O teste de Wilcoxon pareado consiste em um teste estatístico de hipótese não paramétrico. Esse teste é utilizado quando as amostras a serem utilizadas não seguem uma distribuição normal e serve para testar se existe diferenças significativas entre as condições em que as amostras se encontram. As hipóteses a serem testadas são as seguintes:

- *Hipótese Nula ($H_0$)*: A diferença entre os pares segue uma distribuição ao redor do valor 0.
- *Hipótese Alternativa ($H_1$)*: A diferença entre os pares não segue uma distribuição simétrica ao redor do valor 0.

Se o valor p calculado ao fim do teste for menor do que o valor testado, que é geralmente 0.05, pode-se rejeitar a hipótese nula. Para que esse teste possa ser realizado, as observações devem ser randômicas e vir de uma mesma população.

## 3.5 Método Final

Como já havia sido dito na seção anterior, os modelos foram treinados em duas situações: a primeira com o banco de dados original, que aparecerá nos resultados como DA, e a segunda com o banco de dados aumentado, que aparecerá nos resultados como DA+. Devido ao fato das imagens estarem separadas em pastas, decidiu-se utilizar a validação cruzada *k-fold* para $k = 6$. Essa quantidade de variações foi escolhida por ser a mínima necessária para a realização do teste estatístico não paramétrico de Wilcoxon pareado, com a finalidade de verificar se os modelos treinados gerariam resultados com diferença estatística significativa.

Utilizando as configurações de rede neural descritas anteriormente, foram treinadas as arquiteturas VGG16, ResNet50 e MobileNet. As arquiteturas foram treinadas com as seguintes configurações:

- Modelo A com apenas 1 camada densa de tamanho 2
- Modelo B com 2 camadas densas de tamanho 2 e 64
- Modelo C com 3 camadas densas de tamanho 2, 64 e 128
- Modelo D com 4 camadas densas de tamanho 2, 64, 128 e 256

Esse treinamento foi feito com o intuito de verificar se, ao aplicar a técnica de extração de características, a quantidade de camadas densas altera significantemente o resultado. Vale ressaltar que todos os modelos treinados neste trabalho tiveram a perda da validação monitorada com paciência de 3 épocas antes de uma possível parada antecipada. Por conta disso, quase todos os modelos não prosseguiram o treinamento além de cinco épocas.

# 4 RESULTADOS

A primeira tabela disponível em cada uma das seções mostra uma comparação entre os modelos de mesma configuração que foram treinados em condições diferentes, ou seja, há uma comparação entre os modelos semelhantes treinados no conjunto de dados DA e no conjunto de dados DA+. Essa comparação ocorre mostrando tanto os resultados médios de acurácia quanto os de AUC. A coluna que apresenta o valor p indica se existe ou não alguma diferença estatística significativa entre os modelos treinados com DA ou DA+. Esse valor p, como já citado na seção anterior, foi calculado através do teste de Wilcoxon pareado não paramétrico com nível de significância de 0.05. Ou seja, como também explicado na seção anterior, caso o valor p calculado possua valor inferior a 0.05, tem-se que a hipótese nula é rejeitada e, portanto, os modelos possuem resultados cujas diferenças que não são simétricas ao redor de 0. Caso, de forma contrária, o valor p calculado seja superior a 0.05, nada se pode inferir sobre a hipótese nula e, por sua vez, não se pode afirmar que os modelos produzem resultados que sejam de fato diferentes.

As demais tabelas, em cada uma das seções, farão um comparativo entre os modelos com diferentes configurações que foram treinados em condições iguais, ou seja, há uma comparação entre os modelos com diferentes quantidades de camadas treinados em iguais conjuntos de DA ou DA+. O objetivo é unicamente observar se os resultados produzidos foram estatisticamente diferentes, verificando se a quantidade de camadas densas influencia no resultado de uma rede.

## 4.1 MobileNet

Tabela 3: Valores médios de Acurácia e AUC para as diferentes configurações e conjuntos de dados de treinamento da arquitetura MobileNet

| | MobileNet Acurácia e Desvio Padrão | | | MobileNet AUC e Desvio Padrão | | |
|---|---|---|---|---|---|---|
| Configs | DA | DA+ | p | DA | DA + | p |
| Modelo A | 55,68% ± 2,86% | 52,70% ± 2,33% | 0,07 | 0,60 ± 0,05 | 0,63 ± 0,08 | 0,60 |
| Modelo B | 58,07% ± 2,47% | 58,05% ± 8,33% | 0,75 | 0,66 ± 0,06 | 0,71 ± 0,07 | **0,03** |
| Modelo C | 57,85% ± 3,87% | 52,77% ± 2,32% | 0,07 | 0,63 ± 0,05 | 0,66 ± 0,07 | 0,34 |
| Modelo D | 57,13% ± 3,65% | 53,71% ± 4,08% | 0,12 | 0,64 ± 0,07 | 0,64 ± 0,03 | 0,92 |

Tabela 4: Valor-p calculado entre os pares de Acurácia da arquitetura MobileNet para os modelos treinados no banco de dados original (DA)

| Acurácia | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,34 |
| Modelo A e Modelo C | 0,34 |
| Modelo A e Modelo D | 0,92 |
| Modelo B e Modelo C | 0,92 |
| Modelo B e Modelo D | 0,60 |
| Modelo C e Modelo D | 0,75 |

Tabela 5: Valor-p calculado entre os pares de AUC da arquitetura MobileNet para os modelos treinados no banco de dados original (DA)

| AUC | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,25 |
| Modelo A e Modelo C | 0,60 |
| Modelo A e Modelo D | 0,55 |
| Modelo B e Modelo C | 0,24 |
| Modelo B e Modelo D | 0,60 |
| Modelo C e Modelo D | 0,50 |

Tabela 6: Valor-p calculado entre os pares de Acurácia da arquitetura MobileNet para os modelos treinados no banco de dados aumentado (DA+)

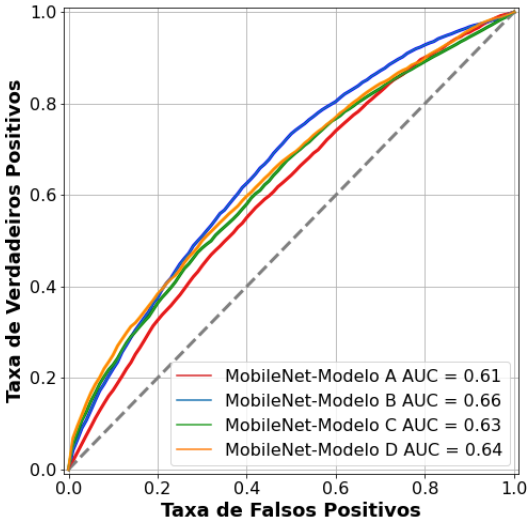
| Acurácia | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | **0,05** |
| Modelo A e Modelo C | 0,92 |
| Modelo A e Modelo D | 0,92 |
| Modelo B e Modelo C | **0,05** |
| Modelo B e Modelo D | **0,03** |
| Modelo C e Modelo D | 0,34 |

Tabela 7: Valor-p calculado entre os pares de AUC da arquitetura MobileNet para os modelos treinados no banco de dados aumentado (DA+)

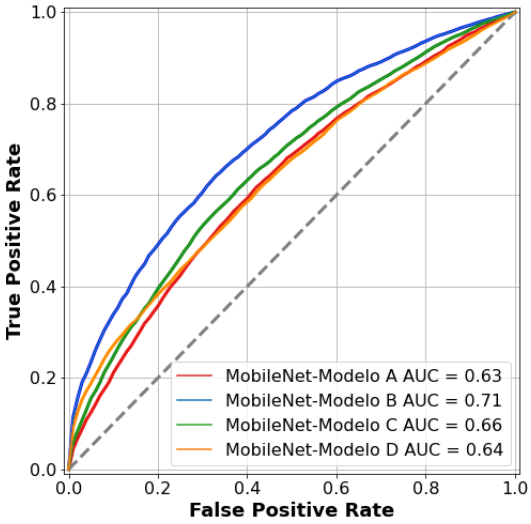
| AUC | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,11 |
| Modelo A e Modelo C | 0,35 |
| Modelo A e Modelo D | 0,92 |
| Modelo B e Modelo C | **0,04** |
| Modelo B e Modelo D | **0,03** |
| Modelo C e Modelo D | 0,60 |

Figura 21: Curva ROC MobileNet

(a) ROC MobileNet DA




(b) ROC MobileNet DA+




Ao observar os dados da tabela 3 é possível observar que a acurácia da rede teve seu melhor desempenho médio com duas camadas densas, tanto para o caso em que a rede neural foi treinada com o banco de dados original (DA) quanto para o caso em que a rede neural foi treinada com o banco de dados aumentado (DA+). Dentre os valores-p calculados para a tabela 3, é possível observar que, para o caso em que a rede neural possuía 2 camadas densas, a AUC apresentou em seu teste de Wilcoxon $valorp = 0,03 < 0,05$. Ou seja, nesse caso foi possível rejeitar a hipótese nula, comprovando que existe diferença estatisticamente significativa entre os resultados obtidos através dos modelos treinados com o conjunto de dados original e com o conjunto de dados aumentado.

Outra observação é a de que a acurácia dos modelos treinados com o conjunto de dados original teve sua média superior as acurácias dos modelos treinados com o conjunto de dados aumentado. Isso pode ser um indicativo de que a arquitetura MobileNet, na presente configuração dos hiper-parâmetros, não foi capaz de generalizar suas predições e, por sua vez, pode haver regredido no aprendizado. Entretanto, os valores p calculados entre os modelos não permitem indicar que os resultados obtidos possuem diferença estatisticamente significativa.

Vale ressaltar também que, ao observar as tabelas 6 e 7, o Modelo B, treinado com 2 camadas densas, teve seu resultado comprovadamente diferente de praticamente todas as arquiteturas, não rejeitando a hipótese nula apenas ao comparar os valores de AUC com o modelo de uma camada densa.

Por fim, ao observar as curvas ROC mostradas na figura 21, é possível observar que com o aumento da sensibilidade dos modelos, houve a rápida diminuição da especificidade, ou seja, houveram taxas grandes de falsos positivos classificados. Para o caso dos modelos treinados no conjunto de dados aumentado, as redes treinadas com 2 camadas densas

tiveram sua performance visivelmente melhores que os demais casos, apesar das taxas de especificidade ainda diminuírem rapidamente com o aumento da sensibilidade.

## 4.2 VGG16

Tabela 8: Valores médios de Acurácia e AUC para as diferentes configurações e conjuntos de dados de treinamento da arquitetura VGG16

| | VGG16 Acurácia e Desvio Padrão | | | VGG16 AUC e Desvio Padrão | | |
|---|---|---|---|---|---|---|
| Configs | DA | DA+ | p | DA | DA + | p |
| Modelo A | 72,06% ± 4,08% | 74,58% ± 3,83% | 0,34 | 0,81 ± 0,06 | 0,83 ± 0,03 | 0,60 |
| Modelo B | 73,60% ± 2,04% | 75,18% ± 2,40% | 0,07 | 0,84 ± 0,03 | 0,82 ± 0,04 | **0,05** |
| Modelo C | 74,04% ± 2,21% | 74,19% ± 1,67% | 0,75 | 0,79 ± 0,03 | 0,79 ± 0,03 | 0,75 |
| Modelo D | 73,02% ± 2,86% | 74,72% ± 2,34% | 0,07 | 0,80 ± 0,04 | 0,80 ± 0,03 | 0,34 |

Tabela 9: Valor p calculado entre os pares de Acurácia da arquitetura VGG16 para os modelos treinados no banco de dados original (DA)

| Acurácia | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,92 |
| Modelo A e Modelo C | 0,46 |
| Modelo A e Modelo D | 0,75 |
| Modelo B e Modelo C | 0,25 |
| Modelo B e Modelo D | 0,92 |
| Modelo C e Modelo D | 0,17 |

Tabela 10: Valor p calculado entre os pares de AUC da arquitetura VGG16 para os modelos treinados no banco de dados original (DA)

| AUC | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,07 |
| Modelo A e Modelo C | 0,35 |
| Modelo A e Modelo D | 0,75 |
| Modelo B e Modelo C | **0,03** |
| Modelo B e Modelo D | **0,03** |
| Modelo C e Modelo D | 0,46 |

Tabela 11: Valor p calculado entre os pares de Acurácia da arquitetura VGG16 para os modelos treinados no banco de dados aumentado (DA+)

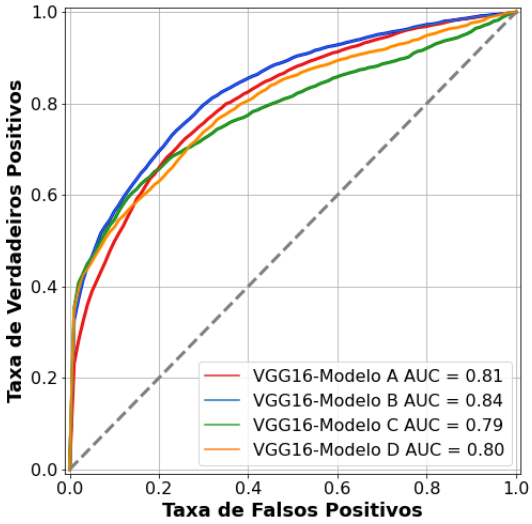
| Acurácia | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,60 |
| Modelo A e Modelo C | 0,92 |
| Modelo A e Modelo D | 0,60 |
| Modelo B e Modelo C | 0,53 |
| Modelo B e Modelo D | 0,60 |
| Modelo C e Modelo D | 0,92 |

Tabela 12: Valor p calculado entre os pares de AUC da arquitetura VGG16 para os modelos treinados no banco de dados aumentado (DA+)

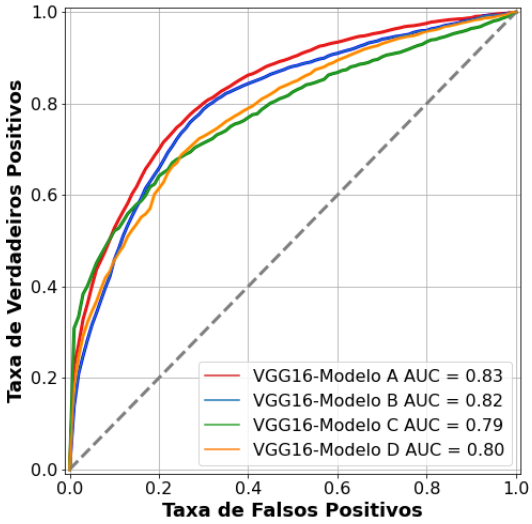
| AUC | |
|---|---|
| Configurações | Valor p |
| Modelo A e Modelo B | 0,25 |
| Modelo A e Modelo C | 0,07 |
| Modelo A e Modelo D | 0,07 |
| Modelo B e Modelo C | 0,17 |
| Modelo B e Modelo D | 0,12 |
| Modelo C e Modelo D | 0,34 |

Figura 22: Curva ROC VGG16

(a) ROC VGG16 DA




(b) ROC VGG16 DA+




A arquitetura VGG16 produziu resultados muito similares, conforme indicado pela tabela 8. Apesar de ligeiramente superiores, as acurácias dos modelos treinados com o conjunto de dados aumentado não possuem diferença estatisticamente significativa das acurácias dos modelos treinados com o conjunto de dados original. O mesmo pode ser dito

dos valores de AUC, apesar de, para o caso com duas camadas densas, o valor-p inferior a 0.05 indicou haver diferença entre os modelos treinados com conjuntos de dados diferentes.

No que diz respeito a comparação direta entre a quantidade de camadas, as tabelas 11 e 12 mostram que não houveram diferenças estatisticamente significativas entre os modelos treinados no mesmo conjunto de dados (DA+). As tabelas 9 e 10 mostram que houveram dois casos em que se rejeitou a hipótese dos modelos gerarem resultados similares, sendo ambos envolvendo o Modelo B com duas camadas densas da tabela, indicando que os resultados foram diferentes dos casos com três e quatro camadas densas para AUC.

Observando as curvas ROC mostradas na figura 23, é possível observar que, tanto para os modelos treinados com o banco de dados aumentado quanto para os modelos treinados com o banco de dados original, a medida que o número de camadas aumenta, tem-se uma diferenciação nas curvas. Para o mesmo valor de sensibilidade, a especificidade diminui, ou seja, a medida que o número de camadas aumentou, a quantidade de exames classificados erroneamente como positivos também aumentaram. É possível observar que nos modelos treinados no conjunto de dados aumentado, essa diferença foi ligeiramente maior entre os modelos A e B, com uma e duas camadas densas respectivamente, para os modelos C e D, com 3 e 4 camadas densas respectivamente.

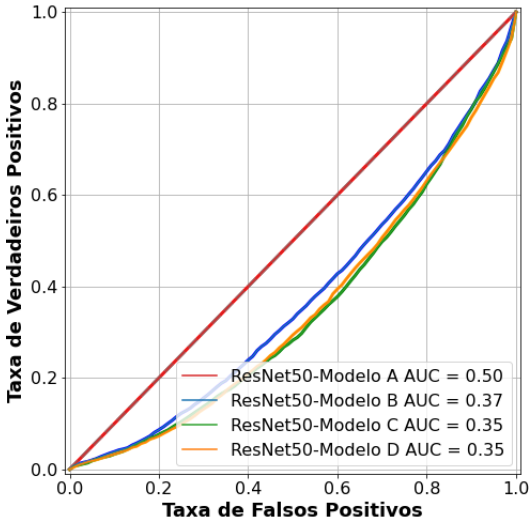
## 4.3 ResNet50

A Arquitetura ResNet50 não foi capaz de aprender a tarefa da forma esperada, mesmo no conjunto de dados com *data augmentation*, uma vez que a rede diferenciou as distorções arquiteturais de forma contrária. As imagens abaixo mostram que as curvas ROC foram geradas no canto inferior direito, indicando que os modelos treinados tiveram um alto número de falsos positivos classificados. A acurácia de todas as redes neurais treinadas permaneceu em 50%, indicando a incapacidade de aprender a tarefa.

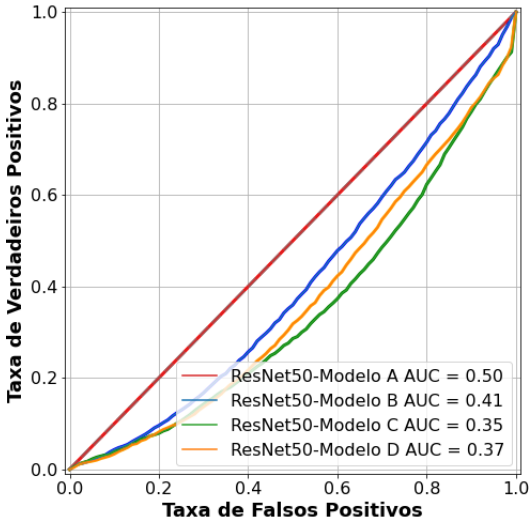
Caso a rede possuísse uma acurácia com valor inferior a 50%, seria possível afirmar que os resultados obtidos foram diferenciados, apesar de estarem com sua classificação errada, indicando um possível erro na alimentação da rede. Entretanto, como a acurácia de todas as redes é fixa em 50%, nada se pode afirmar uma vez que os resultados são aleatórios.

Figura 23: Curva ROC ResNet50

(a) ROC ResNet50 DA




(b) ROC ResNet50 DA+



## 4.4 Comparações

Essa seção tem por objetivo comparar os resultados obtidos por meio de *transfer learning* com os resultados da rede neural treinada especificamente para detectar distorção arquitetural mamária feita por Costa (2019).

Em seu trabalho, Costa (2019) fez o treinamento de seu modelo utilizando o mesmo conjunto de dados original (DA) utilizado no presente trabalho, bem como um treinamento com um conjunto de dados aumentado (DA+). Sobre a rede neural convolucional implementada, a quantidade de camadas de convolução alternadas com as camadas de *max pooling* foi determinada pelo autor de maneira a gerar mapas de característica na saída que possuíssem dimensão 4x4. Dessa forma, o mapa de característica resultante possui dimensão inferior à dos filtros utilizados, que por sua vez possuem tamanho 5x5. Nas etapas de convolução foi utilizado o *padding* com espelhamento para que a dimensão fosse mantida e *stride* com valor unitário. Já as camadas de *max pooling* possuem dimensão 2x2 com *stride* de 2 *pixels*, para que não ocorresse a repetição dos valores analisados e nem a sobreposição da janela. Por fim, foram adicionadas 2 camadas densas possuindo 1024 e 2 neurônios respectivamente. Foi utilizado a técnica de *dropout* com taxa de 0.5 e o algoritmo de otimização escolhido foi o Adam. A figura 24 mostra o modelo especificamente treinado com imagens de distorção arquitetural mamográfica proposto pelo autor, onde a sigla TC referencia a camada de rede totalmente conectada ou densa.

Figura 24: Modelo de rede neural convolucional proposto para detecção de distorção arquitetural mamográfica


Fonte: Costa (2019)

Para o conjunto de dados DA, o valor médio de sua acurácia e AUC foi de 68%±5, 2% e $0,76 \pm 0,064$ respectivamente. Os resultados para as redes MobileNet não foram capazes de atingir esses valores de acurácia e AUC, sendo inferiores em até 18% para os valores de acurácia e 20% para os valores de AUC. Já para a rede VGG16, os resultados obtidos foram superiores em até 4% para acurácia e 9% para AUC.

Para o conjunto de dados DA+, o valor médio de acurácia e AUC obtido por Costa (2019) foi de $71\% \pm 5,3\%$ e $0,78 \pm 0,056$ respectivamente. Os resultados da rede MobileNet

nessas condições foram inferiores em até 18% para acurácia e 8% para AUC. Já para a rede VGG16 utilizando o conjunto de dados aumentado, os resultados foram superiores em até 5% para a acurácia e 4% para AUC. A tabela 13 resume todas as informações supracitadas.

Tabela 13: Comparação entre os modelos com transferência de aprendizado e a rede específica

| Banco de Dados: | DA | | DA+ | |
|---|---|---|---|---|
| | Acurácia | AUC | Acurácia | AUC |
| Modelo Costa (2019) | 68% ±5,2% | 0,76 ±0,074 | 71% ±5,3% | 0,78 ±0,056 |
| MobileNet | 18% ↓ | 20% ↓ | 18% ↓ | 8% ↓ |
| VGG16 | 4% ↑ | 9% ↑ | 5% ↑ | 4% ↑ |

É possível observar que a rede VGG16 treinada previamente no conjunto de dados ImageNet teve sua performance superior a rede treinada exclusivamente para detectar distorções arquiteturais mamárias. Entretanto, algumas observações precisam ser feitas no tocante a comparação feita. Primeiramente, no trabalho de Costa (2019), o conjunto de dados aumentado foi feito de forma diferente do presente trabalho. Foram realizadas adições de ruído Poisson e foram geradas 15 imagens adicionais para cada imagem presente no conjunto de dados original, gerando um banco de dados 16 vezes maior. Isso contrasta com o presente trabalho que fez a utilização de ruído gaussiano com média 0 e variância 0.04, gerando 9 imagens adicionais para cada imagem do conjunto DA, totalizando em uma conjunto DA+ dez vezes maior.

Vale ainda ressaltar que Costa (2019) aplicou em seu trabalho uma validação cruzada *k-fold* para k = 10, enquanto no presente trabalho o valor de k foi de 6. Isso indica drásticas diferenças nos resultados. Uma vez que cada pasta das imagens possui recortes de 14 exames de mamografia digital laudados, as redes treinadas por Costa (2019) foram capazes de observar 140 diferentes exames com presença de distorção arquitetural mamária, enquanto que o presente trabalho, por utilizar k = 6, por limitações de tempo, obteve resultados nos quais as redes feitas foram capazes de observar apenas 84 casos.

# 5 CONCLUSÃO

A utilização da transferência de aprendizado para a detecção de distorção arquitetural, com base nas redes pré treinadas no conjunto de dados *ImageNet* da Universidade de Stanford, se manifestou de formas diferentes para diferentes arquiteturas. A rede ResNet50 não foi capaz de diferenciar as imagens, levando a uma porcentagem crescente de falsos positivos. Já as arquiteturas MobileNet e VGG16 foram capazes de produzir redes neurais com acurácias superiores a 50%.

No que diz respeito a quantidade de camadas densas, de acordo com comunidades profissionais de aprendizado de máquina online, como as da própria Google, aplicações em geral utilizam duas camadas e, portanto, esperava-se que os modelos com duas camadas mostrassem desempenho superior, mesmo que ligeiramente. De fato, para a arquitetura MobileNet, tanto para os modelos treinados com o conjunto de dados aumentado quanto para os modelos treinados com o conjunto de dados normal, os resultados de acurácia e AUC foram em média superiores para os modelos treinados com duas camadas densas, possuindo diferença estatisticamente significativa para os modelos treinados com o conjunto de dados aumentado. Para a arquitetura VGG16, os valores de acurácia e AUC para duas camadas densas também foram ligeiramente superiores para os modelos treinados no conjunto de dados original, apesar de não possuírem diferença estatisticamente significativa. O único caso em que o modelo com 2 camadas densas não superou os demais foi para a média das AUC no conjunto de dados aumentados, uma vez que o modelo com 1 camada densa teve uma performance melhor. Entretanto, não se pode rejeitar a hipótese nula de que os modelos possuem resultados similares.

Ao se comparar os resultados obtidos com as redes geradas com o uso de *transfer learning* e as redes treinadas exclusivamente para detectar distorções arquiteturais mamárias, foi possível observar que para a rede VGG16 os resultados foram superiores aos das redes exclusivas enquanto que os resultados da rede MobileNet foram inferiores. Como explicitado na seção anterior, os modelos do presente trabalho e os feitos exclusivamente para a detecção de distorção arquitetural mamária possuem diferenças de procedimento. Entretanto, a comparação entre eles não é inválida apesar das diferenças. O bom desempenho da arquitetura VGG16 em relação ao modelo feito especificamente para classificar distorções arquiteturais mamárias demonstra o grande potencial da técnica de *transfer learning*.

Foi possível observar que a técnica de extração de características pode ser utilizada para detecção de distorção arquitetural. O presente trabalho obteve os resultados com base em apenas uma configuração de hiper-parâmetros, por conta do alto custo temporal para treinamento de todos os casos. Entretanto, existe a possibilidade de melhorar os

resultados ao serem testados diferentes hiper-parâmetros para as redes. Mesmo a arquitetura ResNet50 não convergindo no presente trabalho, são necessários outros testes com diferentes funções de perda e otimizadores para afirmar que ela não é capaz de identificar distorções arquiteturais utilizando extração de características com o banco de dados ImageNet.

Por fim, sugere-se também para trabalhos futuros a observação do comportamento de extração de características em redes pré treinadas em imagens de exames médicos como, por exemplo, imagens de tomografia computadorizada. Existe a possibilidade de que as redes neurais sejam capazes de identificar distorções arquiteturais mais efetivamente do que quando treinada em um conjunto de imagens não relacionadas com a tarefa objetivo, como é o caso da ImageNet.

# REFERÊNCIAS


Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bordás, P., Jonsson, H., Nyström, L., Cajander, S., and Lenner, P. (2004). Early breast cancer deaths in women aged 40–74 years diagnosed during the first 5 years of organised mammography service screening in north sweden. *The Breast*, 13(4):276–283.

Costa, A. C. (2019). *Detecção de distorção arquitetural mamária em mamografia digital utilizando rede neural convolucional profunda*. PhD thesis, Universidade de São Paulo.

Fenton, J. J., Taplin, S. H., Carney, P. A., Abraham, L., Sickles, E. A., D'Orsi, C., Berns, E. A., Cutter, G., Hendrick, R. E., Barlow, W. E., et al. (2007). Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*, 356(14):1399–1409.

Gençay, R. and Qi, M. (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *IEEE Transactions on Neural Networks*, 12(4):726–734.

Goodfellow, Ian; Bengio, Y. e. C. A. (2016). Deep learning. <`http://www.deeplearningbook.org/` [Acesso em 10/06/2019].

Greenspan, H., Van Ginneken, B., and Summers, R. M. (2016). Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159.

Hassan, M. u. (2018). Vgg16 – convolutional network for classification and detection. <`https://neurohive.io/en/popular-networks/vgg16/` [Acesso em 20/06/2019].

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hidaka, Akinori e Kurita, T. (2017). Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks. volume 2017, pages 160–167.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Instituto Nacional de Câncer, I. (2020a). Câncer de mama. <`https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama` [Acesso em 10/04/2020].

Instituto Nacional de Câncer, I. (2020b). Estimativa 2020: incidência de câncer no brasil. <https://www.inca.gov.br/publicacoes/livros/estimativa-2020-incidencia-de-cancer-no-brasil [Acesso em 10/04/2020].

International Agency for Research on Cancer, I. (2018). Estimated number of deaths in 2018, worldwide, females, all ages. <https://gco.iarc.fr/today/online-analysis-pie [Acesso em 10/04/2020].

Kale, S., Kumar, R., and Vassilvitskii, S. (2011). Cross-validation and mean-square stability. In *In Proceedings of the Second Symposium on Innovations in Computer Science (ICS2011*. Citeseer.

Kawamura, T. (2002). Interpretação de um teste sob a visão epidemiológica: eficiência de um teste. *Arquivos Brasileiros de Cardiologia*, 79(4):437–441.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Li, Z. and Hoiem, D. (2017). Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Lin, Y.-P. and Jung, T.-P. (2017). Improving eeg-based emotion classification using conditional transfer learning. *Frontiers in human neuroscience*, 11:334.

Nemoto, M., Honmura, S., Shimizu, A., Furukawa, D., Kobatake, H., and Nawano, S. (2009). A pilot study of architectural distortion detection in mammograms based on characteristics of line shadows. *International journal of computer assisted radiology and surgery*, 4(1):27–36.

NIPS*95 Post-Conference Workshop (1995). Nips*95 post-conference workshop. <http://web.archive.org/web/19981207070548/http://www.cs.cmu.edu/afs/cs.cmu.edu/usr/caruana/pub/transfer.html#organizers> [Acesso em 10/06/2019].

of Radiology, A. C. et al. (2003). Illustrated breast imaging reporting and data system (bi-rads). *American College of Radiology*.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Ponti, M. A. e. a. (2017). Everything you wanted to know about deep learning for computer vision but were afraid to ask. *30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, pages 17–41.

Rangayyan, R. M., Banik, S., and Desautels, J. L. (2010). Computer-aided detection of architectural distortion in prior mammograms of interval cancer. *Journal of Digital Imaging*, 23(5):611–631.

Rauber, T. W. (2005). Redes neurais artificiais. *Universidade Federal do Espírito Santo*.

Rodriguez-Ruiz, A., Lång, K., Gubern-Merida, A., Broeders, M., Gennaro, G., Clauser, P., Helbich, T. H., Chevalier, M., Tan, T., Mertelmeier, T., et al. (2019). Stand-alone artificial intelligence for breast cancer detection in mammography: comparison with 101 radiologists. *JNCI: Journal of the National Cancer Institute*, 111(9):916–922.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Silva, I. d., Spatti, D., and Flauzino, R. (2010). Redes neurais artificiais, curso prático. *ArtLiber Ed*.

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

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Tosteson, A. N., Fryback, D. G., Hammond, C. S., Hanna, L. G., Grove, M. R., Brown, M., Wang, Q., Lindfors, K., and Pisano, E. D. (2014). Consequences of false-positive screening mammograms. *JAMA internal medicine*, 174(6):954–961.

Walia, A. S. (2017). Activation functions and it's types - which is better? <`https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f` [Acesso em 10/06/2019].

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.