UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA

Aplicação de Redes Convolucionais Profundas para Detecção de Massas em Mamografias

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Eletrônica, Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de bacharel no Curso de Engenharia Eletrônica.

Roberto Augusto Philippi Martins

Orientador: Prof. Danilo Silva, Ph.D.

Florianópolis, 05 de Julho de 2019.

ROBERTO AUGUSTO PHILIPPI MARTINS

APLICAÇÃO DE REDES CONVOLUCIONAIS PROFUNDAS PARA DETECÇÃO DE MASSAS EM MAMOGRAFIAS

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Eletrônica, Departamento de Engenharia Elétrica e Eletrônica da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de bacharel no Curso de Engenharia Eletrônica.
Orientador: Prof. Danilo Silva, Ph.D. .

FLORIANÓPOLIS
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Martins, Roberto Augusto Philippi
Aplicação de Redes Convolucionais Profundas para Detecção de Massas em Mamografias / Roberto Augusto Philippi Martins ; orientador, Danilo Silva, 2019.
100 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Eletrônica, Florianópolis, 2019.

Inclui referências.

1. Engenharia Eletrônica. 2. Redes convolucionais profundas. 3. Diagnostico assistido por computador. 4. Câncer de mama. I. Silva, Danilo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Eletrônica. III. Título.

Roberto Augusto Philippi Martins

APLICAÇÃO DE REDES CONVOLUCIONAIS PROFUNDAS PARA DETECÇÃO DE MASSAS EM MAMOGRAFIAS

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do título de Bacharel em Engenharia Eletrônica e aprovado em sua forma final pelo Curso de Graduação em Engenharia Eletrônica.

Florianópolis, 05 de Julho de 2019.

Prof. Jefferson Luiz Brum Marques, Ph.D.
Coórdenador do Curso

Banca examinadora:

Prof. Danilo Silva, Ph.D.
Universidade Federal de Santa Catarina

Prof. Jefferson Luiz Brum Marques, Ph.D.
Universidade Federal de Santa Catarina

Tiago Stein D'Agostini, MSc.,UFSC
Pixeon Comércio e Desenvolvimento de Software LTDA

Agradecimentos

Agradeço em primeiro lugar a minha família, e em especial a minha mãe, pelo apoio oferecido durante todo esse tempo. Obrigado pelo carinho e os ensinamentos que me fizeram ser quem eu sou hoje. Obrigado por sempre acreditar em mim e pelo suporte que me permitiu chegar até aqui.

Aos amigos e colegas da faculdade, que me ajudaram a manter a sanidade durante esses anos de graduação.

Ao professor Danilo e aos colegas da empresa Pixeon, pela oportunidade e apoio no desenvolvimento desse projeto.

E por fim à própria universidade, por possibilitar essa incrível oportunidade de aprendizado.

++?????++ Out of Cheese Error. Redo From Start.
(Terry Pratchett, Interesting Times, 1994)

RESUMO

Câncer de mama é um dos tipos de câncer mais comum entre mulheres no mundo. A alta taxa de incidência e mortalidade da doença exige o desenvolvimento de técnicas que permitam a detecção precoce e precisa de anomalias. Sendo mamografia o tipo de exame de imagem mais comum para avaliação médica do câncer de mama, este trabalho propõe uma ferramenta de auxílio ao diagnóstico médico, visando acelerar o trabalho do profissional e permitir uma avaliação mais precisa. Observa-se que redes convolucionais são o estado da arte em tarefas de visão computacional, e portanto este trabalho emprega redes convolucionais profundas para a detecção de massas, utilizando especificamente detectores convolucionais de um estágio (RetinaNet). Avaliando a ferramenta desenvolvida em um conjunto de teste, obtém-se uma sensitividade de 85% com 1,64 falsos positivos por imagem no banco de dados CBIS-DDSM (*Curated Breast Imaging Subset of DDSM*).

Palavras-chave: Câncer de Mama. Redes Neurais Convolucionais Profundas. Diagnóstico Assistido por Computador. Detecção de Objetos. Visão Computacional.

ABSTRACT

Breast cancer is one of the most frequently diagnosed forms of cancer among women. The high incidence and mortality rates related to this pathology encourages the development of diagnostic technologies that allow early and accurate detection of anomalies. As mammograms are the most commonly used type screening test for breast cancer evaluation, this research proposes the development of a tool to aid medical diagnosis, aiming to accelerate the work of medical professionals and allow more accurate assessments. Convolutional networks obtain state of the art performance on computer vision tasks; thus, this project applies deep convolutional networks to the detection of abnormal breast tissue growths (masses), using a one-stage convolutional detector (RetinaNet). The evaluation of the developed tool on a test group showed a sensitivity of 85%, with 1.64 false positives per image on the CBIS-DDSM database (Curated Breast Imaging Subset of DDSM).

Keywords: Breast cancer. Deep Convolucional Neural Networks. Computer Assisted Diagnosis. Object Detection. Computer Vision.

Lista de Figuras

Lista de Tabelas

Sumário

CAPÍTULO 1

Introdução

O câncer de mama é uma doença causada por mutações genéticas que levam ao crescimento anormal e descoordenado de células do tecido mamário. Apesar de acometer homens e mulheres, é 100 vezes mais comum em pacientes femininos, sendo a forma de câncer mais frequente neste público [11].

No Brasil o câncer de mama representa 29% dos novos diagnósticos oncológicos registrados em mulheres anualmente, com estimados 59.700 novos casos para 2019, representando uma incidência de 51,29 casos/100mil mulheres [12]. Em 2015 a taxa de mortalidade pós-diagnóstico no Brasil foi estimada em 13,68 óbitos/100 mil mulheres, sendo maior nas regiões Sul e Sudeste (15,26 e 14,56/100 mil mulheres) [13]. Tanto a incidência quanto a mortalidade aumentam com a idade de diagnóstico inicial, sendo a taxa de mortalidade 10 vezes maior em pacientes com mais de 60 anos [12].

O diagnóstico precoce e redução de riscos são as principais abordagens utilizadas na redução da mortalidade pelo câncer de mama, com redução de óbitos estimada em 28-95% para pacientes com hábitos de vida saudáveis e que realizam exames anuais [12].

O desenvolvimento de tumores se dá pelo crescimento descoorde-

nado de células, causado por mutações no material genético. Enquanto tecidos saudáveis possuem um ciclo de vida normal, células cancerígenas caracteristicamente dividem-se de forma descontrolada, gerando uma massa celular tumoral [14]. Tumores podem ser benignos ou malignos, onde tumores benignos possuem divisão celular lenta e não são invasivos e tumores malignos possuem crescimento acelerado e são capazes de realizar metástase (formação de nova lesão tumoral secundária) [15].

O diagnóstico do câncer de mama se dá pela realização do autoexame, por mamografias de rotina e biópsias de massas tumorais. A introdução do autoexame e mamografias em programas da saúde da mulher se deu mutualmente na década de 80, aumentando drasticamente o diagnóstico desta doença [14]. Entretanto, ainda hoje são discutidos os impactos de triagens por mamografia no prognostico e mortalidade do câncer de mama [14, 16, 17]. Enquanto a IARC (*International Agency for Research on Cancer*) advoca uma queda de 23-40% nas taxas de mortalidade em participantes de programas de mamografia, outros autores acreditam que esta diminuição é decorrente de outros fatores (realização de autoexames, melhora de terapias sistêmicas, melhora na tecnologia de diagnóstico por imagem) [14, 17]. Apesar de um consenso ainda não ter sido alcançado, é unânime entre profissionais da saúde que são vitais para o prognóstico positivo do câncer de mama que a) haja o diagnóstico precoce e preciso; b) seja determinado o estágio tumoral [14].

Programas de auxílio ao diagnóstico médico (CAD) aplicados em exames médicos tem como objetivo melhorar a qualidade do diagnóstico. Essas ferramentas tem mostrado resultados promissores em diversas áreas [18, 19, 20], e pesquisas intensivas tem sido realizadas na área de mamografias [21].

Para melhorar o diagnóstico de exames de mamografia, é essencial buscar uma ferramenta que permita a detecção precoce e precisa de anomalias. O principal objetivo é diminuir a taxa de falso negativos, evitando que lesões passem despercebidas pelo médico. Esse tipo de ferramenta pode realizar a análise prévia de imagens, apontando regiões com possíveis anomalias para uma seguinte avaliação profissional, ou uma segunda opinião, confirmando o diagnóstico feito pelo médico.

1.1 Objetivos

1.1.1 Objetivos Gerais

Desenvolver uma ferramenta de auxílio ao diagnósticos de câncer de mama utilizando redes convolucionais profundas.

1.1.2 Objetivos Específicos

- Estudar as principais características do câncer de mama e as propriedades mais relevantes dos exames de mamografia.
- Aplicar ferramentas modernas de redes convolucionais para avaliação de anomalias em exames de mamografia.
- Comparar os resultados obtidos com outros métodos da literatura.

CAPÍTULO 2

Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica sobre visão computacional e redes neurais, desenvolvendo os conceitos básicos utilizados no trabalho.

2.1 Visão computacional

A área de visão computacional (*computer vision*) tem como objetivo gerar algoritmos capazes de processar, analisar e identificar imagens de maneira similar ao processo realizado pelo ser humano.

O trabalho de visão computacional vai além do processamento de imagens, visando não só modificar as características da imagem, mas entender o seu conteúdo. As tarefas realizadas envolvem a interpretação do sinal, adquirindo e processando informações relevantes, produzindo então uma caracterização de alto nível da imagem original.

As principais tarefas de interesse realizadas na área de visão computacional podem ser divididas em:

- Classificação: categorizar a imagem dentro de um conjunto de possíveis classes conhecidas.

- Detecção de objetos: encontrar instâncias de objetos em uma imagem. Além da classificação dos objetos, é necessário também localizá-los dentro da imagem.

- Segmentação semântica: classificar individualmente os pixels de uma imagem dentre o conjunto de possíveis classes conhecidas.

- Segmentação semântica de instâncias: segmentação semântica que separa cada ocorrência de objetos dentro de uma classe. Cada instância de um classe é segmentada independentemente.

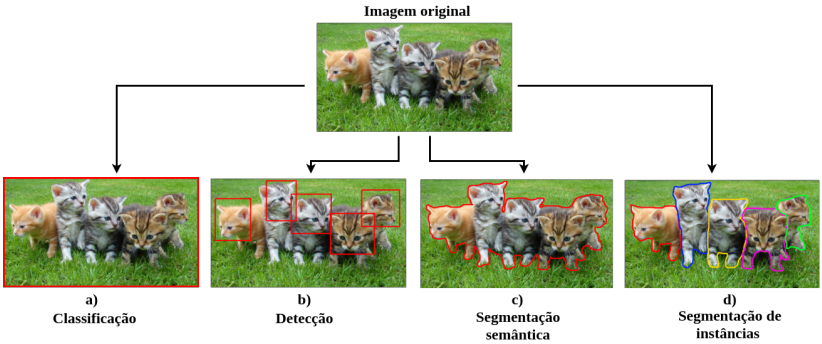

Figura 2.1: Tipos básicos de tarefas realizadas na área de visão computacional. **a)** Classificação da imagem como um todo dentro de um conjunto de classes. **b)** Detecção de objetos entre diferentes classes. Cada objeto é individualmente localizado de maneira aproximada por uma região retangular da imagem. **c)** Segmentação de todos os objetos de uma classe, feito pixel a pixel para a imagem inteira. **d)** Segmentação dos objetos de maneira individual, de forma que a segmentação separada as diversas ocorrências de objetos.

Essas técnicas podem ser aplicadas em diversas áreas, gerando um sistema capaz de interpretar o mundo real e que permite a interação entre computadores e o ambiente. Algumas das áreas de aplicação de visão computacional são:

- Automação (*machine vision*, controle industrial, inspeção de produtos e mercadorias).

- Navegação (carros autônomos, prevenção de acidentes).

- Identificação de pessoas (controle de tráfego, vigilância).
- Leitura automática de textos e placas.
- Diagnóstico médico (detecção de anomalias, classificação de exames, triagem automatizada).

Cada uma dessas tarefas consiste em interpretar a informação contida na imagem original de maneira útil. Isso requer que a variável original (a imagem) seja condensada em uma variável de mais alto nível. Essa nova variável será um vetor com dimensão muito menor que a imagem, mas que é representativo das suas principais características.

Essa condensação da informação pode ser realizada a partir do uso de descritores, que são funções projetadas especificamente para obter variáveis que caracterizam a imagem original.

2.2 Descritores

Uma das maneiras de implementar métodos de visão computacional é pelo uso de descritores. Essa técnica permite gerar um conjunto de variáveis que descreve um ou mais aspectos da imagem original de maneira simplificada.

O uso de descritores foi bastante comum nas últimas décadas, devido principalmente a eficiência computacional dos algoritmos. O projeto de descritores gerou resultados muito interessantes na área de visão computacional, como a detecção de faces [22] e classificação de objetos [23].

Descritores são algoritmos de processamento de imagem que retiram de uma imagem informações como cor, textura, formato, momento de cores, histograma, etc. O sucesso dessa técnica foi impulsionado pelo desenvolvimento de descritores mais eficientes, como HOG (*Histogram of Oriented Gradients*) [24], SIFT (*Scale-Invariant Feature Transform*) [25] e ORB (*Oriented FAST and Rotated BRIEF*) [26], que permitem obter de forma eficiente informações mais complexas sobre a imagem.

Com a obtenção dos descritores adequados é possível realizar a tomada de decisões sobre a imagem (classificação, detecção, segmentação, registro, etc).

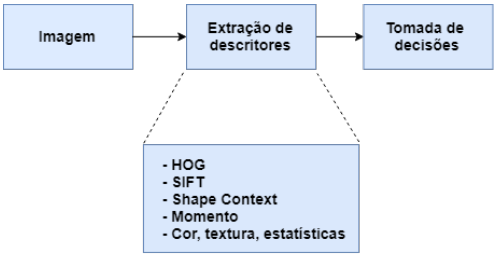

Figura 2.2: A interpretação da imagem (tomada de decisões) é feita com base nos descritores obtidos. Para que a interpretação seja feita corretamente, é necessário que os descritores utilizados reflitam as características de interesse da imagem. Múltiplos descritores podem ser utilizados, formando um vetor de variáveis que representa a imagem original.

Como o conjunto de informações mais relevantes de uma imagem é dependente da aplicação, os descritores precisam ser escolhidos ou adaptados especificamente para cada tarefa. Em geral, descritores são definidos a partir das características da imagem original e heurísticas do desenvolvedor. Dessa forma, a principal dificuldade na utilização de descritores está na complexidade do desenvolvimento de descritores específicos para cada tarefa a ser realizada.

O problema relacionado com o desenvolvimento e escolha de descritores específicos pode ser resolvido com o uso de uma ferramenta adaptável, capaz de aprender a função desejada a partir da experiência e análise de exemplos. Esse tipo de método pertence ao domínio de aprendizado de máquina (*machine learning*), de onde surgem as técnicas de redes neurais e redes convolucionais, capazes de solucionar diversos problemas na área de visão computacional.

2.3 Redes Neurais

Redes neurais artificiais são modelos computacionais com uma inspiração biológica baseada no cérebro humano, capazes de realizar funções complexas a partir da combinação de uma grande quantidade de elementos básicos.

Essa estrutura é montada a partir da combinação de múltiplos nós, onde cada nó realiza uma função matemática simples. O resultado final é um modelo extremamente parametrizável, capaz de se moldar e

aprender a realizar tarefas específicas a partir do ajuste adequados dos parâmetros.

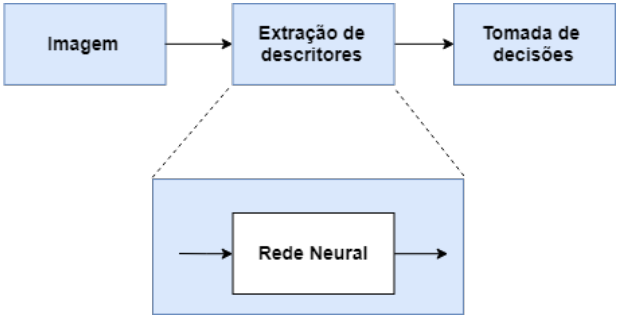

Figura 2.3: As redes neurais são capazes de substituir a função realizada pelos descritores, obtendo variáveis que caracterizam a imagem original. Os procedimentos seguintes, como classificação e detecção, podem ser mantidos iguais ou eventualmente absorvidos para dentro da própria rede neural.

A operação realizada por cada nó de uma rede neural é basicamente a junção de uma combinação linear e uma função de ativação não linear. A simplicidade de um nó lhe permite apenas realizar funções de baixa complexidade, mas um conjunto suficientemente grande de nós é capaz de aproximar qualquer função matemática [27, 28].

A Figura 2.4 é uma representação esquemática de um nó de uma rede neural. A entrada $\mathbf{x}$ passa por uma combinação linear definida pelo vetor de pesos $\mathbf{W}$, produzindo o escalar a_j^k. A saída final z_j^k do nó é o valor a_j^k após passar por uma função de ativação $g(a)$.

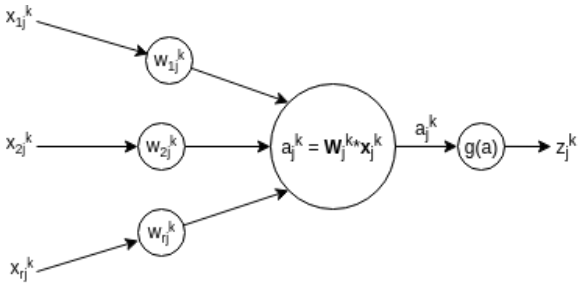

Figura 2.4: Unidade básica de uma rede neural. Representação do nó j, pertencente à camada k e com um vetor de entrada com r componentes. Nota-se que é comum a existência de uma variável de *bias* na combinação linear, de forma que $a_j^k = W_j^k x_j^k + b_j^k$, porém isso pode ser simplificado supondo que um dos componentes do vetor de entrada $\mathbf{x}$ é constante.

A função de ativação $g(a)$ é responsável por adicionar um componente não linear ao comportamento do nó. Sem essa função de ativação, todas os nós realizariam apenas funções lineares que não são capazes de produzir aproximações mais complexas.

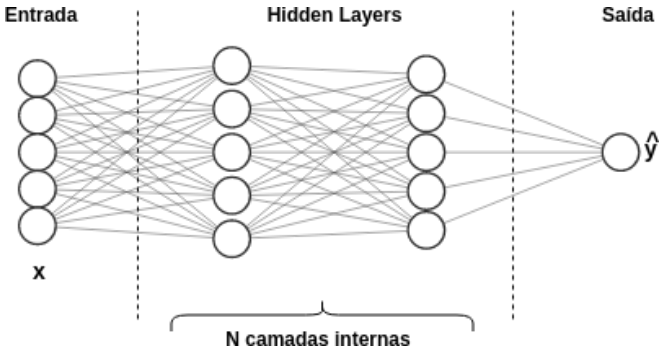

Figura 2.5: Exemplo de uma rede neural básica (*Fully Connected Neural Network*) com as camadas de entrada, saída e internas. A saída y é o resultado da rede, obtido pela composição de operações de cada nó em relação ao vetor de entrada $\mathbf{x}$.

A Figura 2.5 mostra um exemplo de rede neural, uma estrutura conhecida como Rede Neural Completamente Conectada (*Fully Connected Neural Network*). Essa rede é montada a partir da combinação de nós em paralelo (várias unidades em uma mesma camada) e em série

(várias camadas operando em sequência). As características estruturais da rede, como como número de nós, número de camadas, funções de ativação, etc., são conhecidas como hiper-parâmetros da rede.

Independente da estrutura específica, o objetivo é que a função $\hat{f}(\mathbf{x})$ realizada pela rede neural aproxime uma função qualquer $f(\mathbf{x})$ com o menor erro possível.

A qualidade da aproximação realizada pela rede neural depende do ajuste dos parâmetros internos de cada nó. O principal método de ajuste desses parâmetros é formado por um processo de otimização iterativo, onde os pesos são modificados a partir da análise de múltiplos pontos de referência da função $f(x)$ original. A partir da observação de cada uma dessas amostras, a rede é otimizada de forma a minimizar a diferença entre os valores $\hat{y}$ e y, utilizando algoritmos como gradiente descendente e *backpropagation*.

2.4 Gradiente descendente e backpropagation

Backpropagation é o principal método utilizado para treinar redes neurais. Esse algoritmo tem como objetivo otimizar os pesos inicias de uma rede neural, minimizando uma função de perda que caracteriza a qualidade das predições atuais da rede.

Essa técnica é uma maneira eficiente de implementar a otimização por gradiente descendente em redes neurais com múltiplas camadas [29], utilizando a regra da cadeia para propagar o valor do gradiente para todos os elementos da rede. Sem o uso de *backpropagation*, o treinamento de redes neurais com múltiplas camadas se torna computacionalmente custoso demais, impedindo a construção de estruturas maiores e portanto limitando a sua capacidade de representar funções de maior complexidade.

2.4.1 Gradiente Descendente

Supondo uma rede neural com pesos $\mathbf{W}$, obtemos a saída $\hat{y}$ a partir da propagação do sinal de entrada $\mathbf{x}$ pela rede. A qualidade da previsão é medida a partir da função de erro $L(y, \hat{y})$, que mede uma distância característica (como erro médio quadrático, entropia cruzada, etc.) entre o valor esperado y e o previsto $\hat{y}$.

A otimização dos pesos pode ser feita iterativamente utilizando o método do gradiente descendente:

$$w_n^{i+1} = w_n^i - \eta \frac{\partial L(y, \hat{y})}{\partial w_n}, \forall w_n \in \mathbf{W} \tag{2.1}$$

onde η é a taxa de aprendizado que regula a velocidade do aprendizado, e i é a iteração atual do algoritmo. Todos os pesos são otimizados de forma a minimizar o erro $L(y, \hat{y})$, utilizando o valor do gradiente $\frac{\partial L(y,\hat{y})}{\partial w_n}$ para caminhar em direção ao ponto ótimo no espaço.

A principal dificuldade em utilizar esse algoritmo é no cálculo da derivada parcial do erro em relação a cada peso da rede. Backpropagation surge como um método eficiente para calcular os gradientes necessários dentro de uma rede neural com múltiplas camadas.

2.4.2 Backpropagation

O algoritmo *backpropagation* é o método utilizado para propagar eficientemente o valor de $\frac{\partial L(y,\hat{y})}{\partial w_k}$ para as camadas internas da rede neural [30]. Esse método permite calcular o gradiente da camada L a partir do gradiente calculado para a camada $L + 1$, utilizando as características estruturais da rede e a regra da cadeia para diferenciação. O gradiente é, portanto, propagado sequencialmente das camadas mais próximas a saída em direção à entrada.

O funcionamento desse algoritmo depende fortemente da estrutura da rede neural utilizada, de maneira que mudanças na arquitetura e funções de ativação alteram a eficiência do processo de otimização. Além de ser uma forma de implementar o gradiente descendente em redes neurais, *backpropagation* pode ser utilizado em conjunto com outros métodos, como normalização, momento, aprendizado em *batches*, etc.

As equações do *backpropagation* obtém uma forma bastante simplificada e recursiva, o que permite o uso de diferentes métodos para otimizar a implementação computacional do algoritmo [29]. O principal fator para o treinamento de alto desempenho de redes neurais é pela utilização de placas de processamento gráfico (GPUs), que permitem resolver tanto as equações diretas (*feedforward*) quanto as equações do *backpropagation* de maneira muito mais eficiente.

2.5 Redes Neurais Convolucionais

Uma das principais desvantagens de redes neurais completamente conectadas é a dificuldade de se trabalhar com sinais com coerência, como sinais de áudio (coerência sequencial, no tempo) e imagens (coerência espacial, em 2D). Esse tipo de rede neural não leva em consideração a relação entre as partes do sinal de entrada, perdendo uma informação muito preciosa sobre as variáveis.

Uma rede neural convolucional (CNN) é um tipo de rede que permite processar imagens e ainda assim manter a coerência espacial do sinal. Isso é realizado utilizando filtros convolucionais dentro das suas unidades básicas, substituindo a combinação linear convencional por uma operação mais apropriada para uso em imagens.

2.5.1 Filtro Convolucional

Cada camada de uma rede convolucional é composta por um conjunto de filtros convolucionais, onde cada filtro opera sobre a saída da camada anterior.

De forma similar às redes neurais completamente conectadas, cada filtro de uma rede convolucional realiza uma função matemática simples, e o conjunto total de filtros resulta em uma função de maior complexidade.

Convolução é uma operação feita entre dois sinais, a entrada (uma imagem ou a ativação de uma camada convolucional anterior) e um *kernel*, que define a função por qual a entrada será processada. A operação realizada entre uma entrada I e um *kernel* K é:

$$A(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.2}$$

ou, abrindo mão da propriedade de comutatividade, podemos utilizar a função de correlação cruzada [29], que é muito utilizada em diversas bibliotecas computacionais de redes convolucionais,

$$A(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2.3}$$

Essa distinção não é muito importante, já que a única modificação

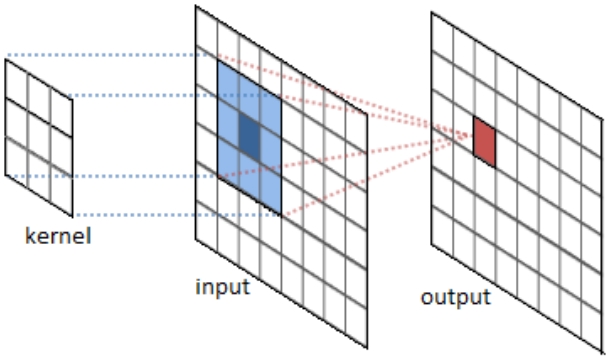

Figura 2.6: Visualização da operação de convolução [1] sobre uma imagem (*input*). Cada ponto da ativação resultante (*output*) corresponde à aplicação do *kernel* sobre uma região diferente da imagem. A aplicação do filtro por toda a imagem resulta em um mapa de ativações

é a inversão ou não do *kernel* antes da sua aplicação à imagem. Como os pesos dos filtros não são parâmetros fixos, e sim valores que serão aprendidos durante o treinamento, essa distinção será irrelevante após a otimização da rede.

A operação de convolução pode visualizada na Figura 2.6.

Três propriedades importantes distinguem as redes convolucionais das redes neurais tradicionais [29] e permitem trabalhar com imagens de maneira mais eficiente:

- Interações esparsas: diferente das redes neurais tradicionais, as redes convolucionais trabalham com conexões esparsas entre a entrada e saída de uma camada. Uma unidade na saída de uma camada convolucional interage apenas com uma região de tamanho $W \times H$ (tamanho do *kernel* utilizado) da imagem original, conhecido como campo receptivo. Essa característica permite um processamento mais eficiente dos dados e facilita a obtenção de informações locais (em torno de uma região da imagem).

- Compartilhamento de parâmetros: durante a convolução, o mesmo *kernel* é aplicado sobre todo o sinal de entrada de maneira sequen-

cial, de forma que os parâmetros da função são utilizados múltiplas vezes. Isso é eficiente para o processamento de imagens, já que um padrão de interesse pode ocorrer em qualquer posição da imagem de maneira independente, e portanto a reutilização do *kernel* permite a otimização de uma única função. Por exemplo, a detecção de uma textura não depende da sua posição na imagem, e sim das características locais, de forma que o mesmo filtro pode ser aplicado em todo o sinal.

- Invariância sob translação: a operação de convolução é invariante à translação, formando uma ativação que é capaz de mapear a ocorrência de uma característica por toda extensão do sinal de entrada. Isso é muito interessante para detecção de objetos, já que a translação do objeto na imagem original resultará numa translação equivalente das ativações. Deve-se notar que redes convolucionais não são invariantes a rotação, mudanças de escala e outras transformações geométricas, de forma que diferentes técnicas devem ser utilizadas para se resolver esses problemas.

2.5.2 Rede Convolucional

Uma rede convolucional básica é estruturada da mesma maneira que uma rede neural comum, a partir da concatenação de filtros em uma série de camadas. Cada camada contém um número conhecido de filtros, onde cada um deles opera sobre a saída da camada anterior.

O treinamento de uma rede convolucional também é feito utilizando *backpropagation*. O princípio de uso do algoritmo é o mesmo, com objetivo de encontrar o valor de $\frac{\partial L(\hat{y},y)}{\partial w_{ij}^k}$ para todos os pesos da rede. A principal diferença está no cálculo de δ^k em função de δ^{k+1}, pois aqui o cálculo de a_j^k é feito a partir da operação de convolução.

Uma rede convolucional é definida por diversos hiper-parâmetros, que descrevem o formato geral da sua construção. Por exemplo, o número de camadas, número de filtros por camada e número total de parâmetros são hiper-parâmetros básicos, geralmente utilizados como um critério de comparação entre estruturas diferentes.

Existem diversos casos de estudo de redes convolucionais que se tornaram base para o desenvolvimento de estruturas futuras. Essas redes são reconhecidas pela apresentação de técnicas inovadoras que

avançaram as pesquisas na área. Alguns exemplos notáveis são LeNet [2] (Figura 2.7), AlexNet [31], VGGNet [3] (Figura 2.8), GoogLeNet [32], ResNet [4] e outros.

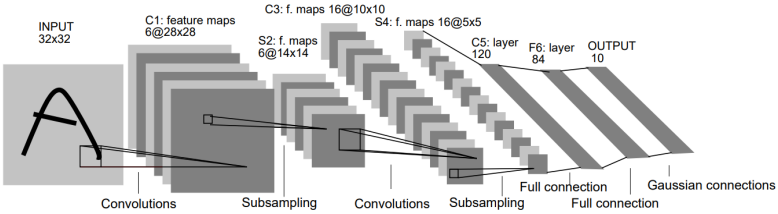

Figura 2.7: Uma das primeiras aplicações bem sucedidas de Redes Convolucionais, feita em 1998 [2]. A rede LeNet foi desenvolvida para realizar reconhecimento de carácteres utilizando filtros convolucionais, mostrando uma alternativa bem sucedida ao uso de descritores.

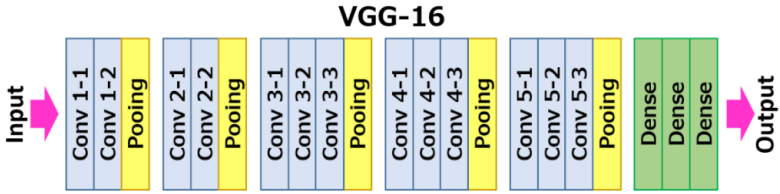

Figura 2.8: A rede VGG16 [3] foi apresentada na competição ILSVRC2014, pelo grupo *Visual Geometry Group* da Universidade de Oxford. Essa rede se tornou a base para o desenvolvimento de diversas outras estruturas, popularizando o uso de múltiplos filtros 3x3 em sequência no lugar de filtros de largura superior, permitindo o desenvolvimento de redes mais profundas com menos parâmetros.

2.6 Redes Residuais

Grande parte do estudo na área de redes neurais está no desenvolvimento de estruturas mais eficientes. O publicação de redes convolucionais como LeNet [2], AlexNet [31] e VGGNet [3] correspondem ao desenvolvimento de novas técnicas que permitem a geração de redes mais complexas e com mais camadas.

A rede ResNet [4] é uma estrutura muito poderosa, que alcança

alguns dos melhores resultados na área de redes convolucionais. Com a publicação dessa nova técnica permitiu-se a construção e otimização de redes muito mais profundas, alcançando desempenho superior às construções disponíveis até então.

Uma das motivações para o desenvolvimento de redes residuais está na questão do desvanecimento do gradiente.

2.6.1 Desvanecimento do Gradiente

Uma das dificuldades de se desenvolver redes mais profundas está no efeito de desvanecimento do gradiente (*vanishing gradient*) [29]. Esse é um fenômeno que ocorre no treinamento das redes durante o uso do algoritmo de *backpropagation*, e impede a propagação adequada do erro para as camadas internas da rede.

Funções de ativação como sigmoide e tangente hiperbólica são utilizadas devido ao efeito de saturação da saída, o que mantém os valores produzidos pelas unidades da rede controlados e impede a explosão dos valores internos da rede. Por outro lado, a derivada dessas funções é limitada (sempre menor que um valor conhecido) e converge para zero quando a entrada se distancia de zero (Figura 2.9).

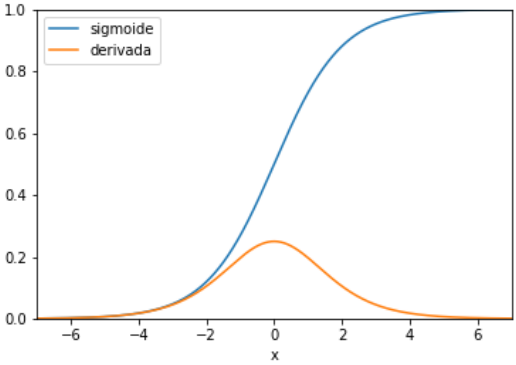

Figura 2.9: Função de ativação sigmoide e a sua derivada. Observa-se que a função sigmoide satura para valores de $\mid x \mid$ grandes, de forma que a sua derivada se aproxima de zero.

Como visto anteriormente, o algoritmo de *backpropagation* realiza a

propagação do gradiente pelas camadas da rede neural. Esse processo é sequencial, e cada etapa exige a multiplicação do resultado anterior pela derivada da função de ativação utilizada entre as camadas. Dessa forma, o uso de funções de ativação como a sigmoide pode gerar valores gradativamente menores do gradiente dentro da rede, de forma que eventualmente o valor recebido pelas primeiras camadas da rede é insuficiente para realizar qualquer otimização significativa na rede, efetivamente interrompendo o treinamento.

Existem diferentes formas de mitigar este fenômeno, que envolvem principalmente mudanças nas funções de ativação e na estrutura da rede. Alguns deles são:

- Utilização de funções de ativação diferentes. A função de ativação ReLU [33, 34], por exemplo, mantém um valor de derivada constante para entradas positivas, evitando o problema de saturação do gradiente para um intervalo de entrada maior.

- Normalização de dados [34]. Podemos normalizar as ativações de cada camada da rede neural, colocando os sinais em um intervalo mais próximo de zero, e portanto dentro de uma janela onde a função de ativação mantém uma derivada maior.

- Uso de funções de perda auxiliares [35], colocadas em camadas intermediárias da rede, de forma a incluir um termo de perda adicional para as primeiras camadas da rede.

- Utilização de blocos convolucionais que utilizam *skip connections* ou *shortcuts* entre as camadas da rede, criando novos caminhos para propagação do gradiente.

2.6.2 Blocos Residuais

Um bloco residual consiste na junção de um conjunto de camadas tradicionais adicionada de uma conexão de atalho em paralelo. A Figura 2.10 mostra um exemplo, onde a conexão de atalho é a função identidade.

O uso de blocos residuais gera um caminho direto entre $\mathbf{y}$ e $\mathbf{x}$, que permite sempre a existência de um gradiente significativo independentemente da função realizada pelas outras camadas. O uso dessa técnica

permite a construção de redes muito mais profundas sem perda de desempenho [4].

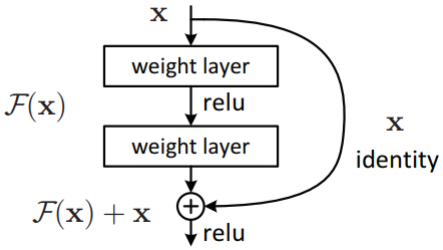

Figura 2.10: Bloco básico de uma rede residual [4]. Existem diversas variações dessa construção, modificando tanto a função realizada na conexão de atalho quanto nas camadas convolucionais tradicionais.

Efetivamente, a função realizada por esse bloco residual é:

$$\mathbf{x}_{i+1} = F(\mathbf{x}_i, \{W_i\}) + \mathbf{x}_i$$

onde $\mathbf{x}_i$ e $\mathbf{x}_{i+1}$ são a entrada e saída do bloco, respectivamente, e $F(\mathbf{x}, \{W\})$ é a função residual implementada pelo bloco.

Outra razão para o sucesso dos blocos residuais, e o motivo do seu nome, está no fato de que a função $F(\mathbf{x}, \{W\})$ precisa aprender apenas a diferença entre os valores x_i e x_{i+1}, e não construir completamente x_{i+1} a partir de x_i. Numa situação onde todos os pesos do bloco residual são zero, ainda será implementado a função identidade, mantendo a continuidade da rede.

Redes residuais podem ser construídas a partir da concatenação de blocos residuais em sequência. O uso de conexões residuais permitiu a construção de redes com centenas de camadas (ResNet101, ResNet152, ResNet1000) [4] sem perda de desempenho.

Estrutura	# Camadas	# Parâmetros	Erro%
FitNet	19	2.5M	8.39
Highway	19	2.3M	7.54
Highway	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43**
ResNet	1202	19.4M	7.93

Tabela 2.1: Comparação de desempenho entre redes ResNet de diferentes profundidades e outras estruturas (Highway, FitNet) [4] para o conjunto de dados CIFAR-10. Percebe-se que o uso de blocos residuais permite o treinamento correto de redes muito mais profundas, com ganho de desempenho. O aumento do erro entre as redes ResNet110 e ResNet1202 é atribuído ao *overfitting* da rede, dado que a capacidade da segunda rede é muito maior e o erro de treinamento de ambas reportado como sendo similar.

2.7 Overfitting, Underfitting e Augmentation

Dentro do contexto de aprendizado de máquina, deseja-se que o modelo produzido seja capaz de generalizar os conhecimentos obtidos no treinamento para novas amostras de dados [29]. Dessa definição surgem duas situações que definem o comportamento do modelo para novas amostras:

- *Overfitting*: situação onde o modelo treinado não aprende a estrutura interna do conjunto de dados, mas sim memoriza as características locais dos exemplos de treinamento. Dessa forma, a capacidade de generalização do modelo é baixa, e o desempenho do algoritmo para novas amostras é inferior ao desempenho obtido durante o treinamento.

- *Underfitting*: situação onde o modelo não alcança o desempenho desejável tanto durante o treinamento quanto em amostras novas. Isso acontece em casos onde a capacidade do modelo não é

suficiente para representar o conjunto de dados, ou que o modelo não foi suficientemente treinado.

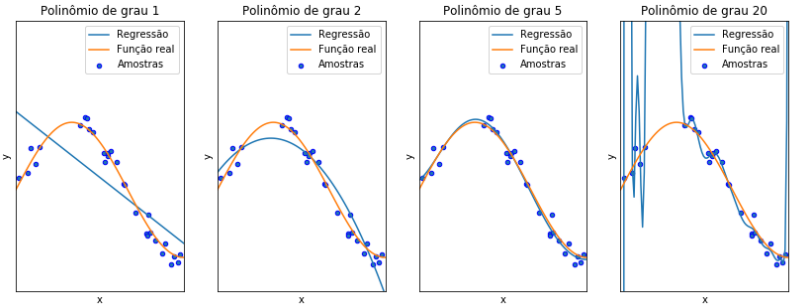

Figura 2.11: Regressão de um mesmo conjunto de dados, utilizando polinômios de diferentes graus. Vemos que um polinômio de grau baixo não tem capacidade de aproximar a função adequadamente, enquanto polinômios de grau muito alto tendem a divergir para valores intermediários às amostras de treinamento. O resultado ótimo pode ser obtido com o uso de uma solução intermediária.

O aparecimento de *overfitting* em redes neurais é bastante comum na prática, devido à alta parametrização dos modelos modernos e o número finito de amostras de treinamento. O resultado é o desenvolvimento de um modelo que realiza predições baseadas em características locais de cada amostra, basicamente memorizando o conjunto de dados, enquanto a capacidade de generalização é reduzida.

Uma maneira comum de amenizar este problema é pela utilização de *augmentation* em tempo de treinamento. Essa técnica visa estender artificialmente o conjunto de dados, criando novas amostras a partir de variações aleatórias de amostras originais.

O efeito disso é que o modelo não é treinado com a mesma imagem múltiplas vezes, mas sim com variações aleatórias distintas da mesma imagem. A tendência é que as características estruturais básicas da imagem continuem constantes (supondo que o processamento utilizado não comprometa o conteúdo da imagem), enquanto o ruído vira uma variável inconsistente para o modelo.

Essa técnica é especialmente poderosa para aplicações de classificação e detecção de imagens, devido à alta dimensionalidade das variáveis

e o número muitas vezes pequeno de amostras.

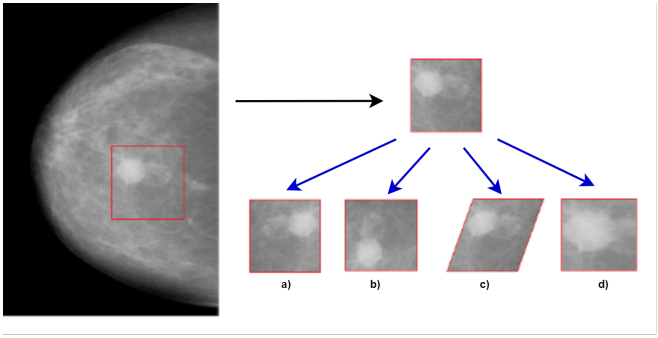

Figura 2.12: Exemplos de *augmentation* sobre uma região da imagem. **a)** e **b)** consistem em um giro horizontal e na rotação da imagem, respectivamente, **c)** consiste na distorção (*skew*) e **d)** representa mudança de escala (*zoom*) da imagem. Muitos outros tipos de *augmentation* existem, não limitados apenas a transformações afins, como por exemplo adição de ruído branco, alteração do contraste e histograma, etc.

2.8 Feature Pyramid Network

Um dos desafios presentes na área de detecção de objetos é o reconhecimento de objetos em escalas diferentes. Um mesmo objeto pode se apresentar em tamanhos diferentes, dependendo da relação entre o objeto e a câmera, o que dificulta o desenvolvimento de algoritmos capazes de detectar os objetos em qualquer situação.

Esse problema é ainda mais acentuado em situações onde trabalha-se com objetos de diferentes classes, e classes diferentes contém naturalmente tamanhos diferentes. A resolução resultante em uma rede convolucional pode ser adequada pra avaliar algumas classes de objetos, mas inadequada para outras.

É interessante, portanto, desenvolver técnicas de detecção que permitam trabalhar com imagens e objetos sujeitos a variações de escalas.

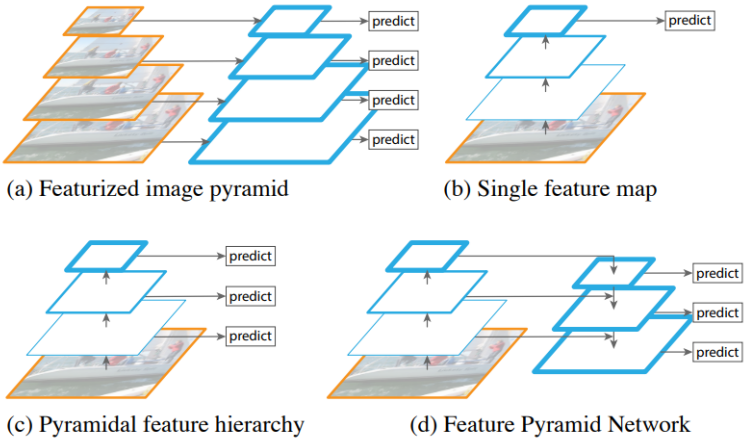

Figura 2.13: **a)** Utilizando uma pirâmide de imagens, é possível avaliar a imagem original em escalas diferentes, mantendo a informação em todos os níveis a um custo computacional alto. **b)** O uso de apenas a última camada convolucional permite um processamento muito mais rápido, porém com um alcance de escalas limitado. **c)** O uso das camadas intermediárias produzidas pela rede convolucional permite se trabalhar com múltiplas escalas sem custo adicional, porém a informação contida em cada nível é diferente, alterando a capacidade de detecção para cada escala avaliada. **d)** A proposta de *Feature Pyramid Network* (FPN) permite uma estrutura com desempenho equivalente à a), porém com complexidade computacional inferior. As camadas produzidas permitem trabalhar com diferentes escalas, ainda assim mantendo ativações com informações de alto nível, devido às conexões laterais e a estrutura *top-down* utilizada.

Featurized Image Pyramid [36] (Figura 2.13(a)) é a base das soluções atuais, sendo muito utilizada tanto com detectores baseados em descritores quanto redes convolucionais. Esse método consiste na realização de predições para diferentes escalas da imagem original, onde cada nível da pirâmide é analisado independentemente dos outros. Isso garante que cada nível da pirâmide mantenha um alto nível de informação, porém em escalas diferentes. A principal desvantagem desse método é a necessidade de realizar todo o processamento para cada nível da pirâmide, aumentando significativamente o uso de memória e tempo de processamento.

Single Feature Map (Figura 2.13(b)) é o método básico utilizado

em redes convolucionais, onde a detecção é feita apenas na última camada de ativação da rede convolucional. Utilizado originalmente em redes como [37, 6], esse método obteve resultados interessantes, mas a inclusão de técnicas multi-escala resulta em estruturas mais precisas.

Pyramidal Feature Hierarchy promete resolver o problema de multi-escala e custo computacional, utilizando camadas já naturalmente produzidas pela rede convolucional para predição. Utilizado por redes como SSD [38], essa técnica falha principalmente na detecção de objetos pequenos, já que as camadas com resolução maior não são processadas o suficiente para possibilitar a detecção de objetos.

Por fim, temos a estrutura *Feature Pyramid Network* (FPN) [36] que permite produzir camadas com escalas diferentes, onde todas contém informação de alto nível capazes de possibilitar a detecção de objetos.

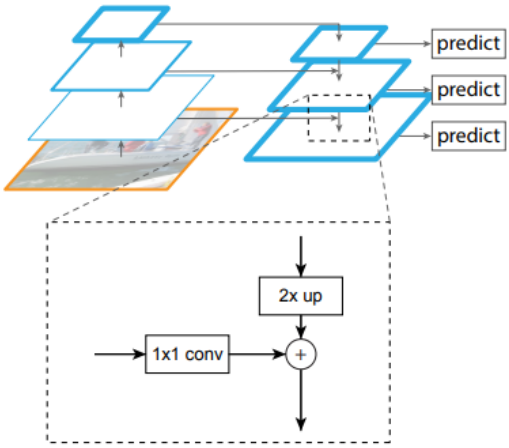

Figura 2.14: *Feature Pyramid Network*. As camadas multi-escala são produzidas a partir de uma estrutura *top-down* e pelo uso de conexões laterais. A construção de camadas de alta resolução a partir de camadas de baixa resolução (utilizando 2x *upscaling*) permite a propagação de informação de alto nível por toda pirâmide, enquanto as conexões laterais (convoluções 1x1) ajudam a reconstruir as relações espaciais da imagem original.

A estrutura FPN permite criar as camadas multi-escala a partir de uma rede básica, chamada de *backbone*, com um processamento extra mínimo. A Figura 2.15 mostra um exemplo de construção feito

a partir de um *backbone* residual (ResNet), resultando em 4 camadas $\{P2, P3, P4, P5\}$ diferentes.

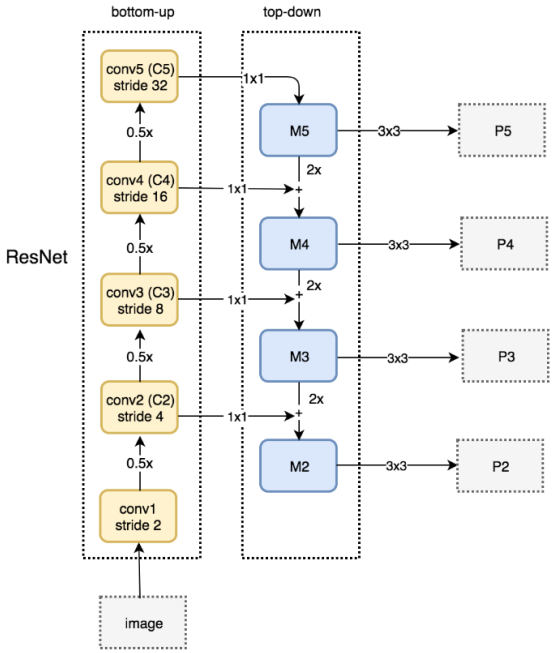

Figura 2.15: *Feature Pyramid Network* construído a partir de um *backbone* ResNet [5]. A estrutura *bottom-up* se refere à rede convolucional, onde são intercaladas camadas convolucionais (conv) e camadas de *pooling* (0.5x). A construção *top-down* constrói a pirâmide de ativações, utilizando as conexões laterais (convoluções 1x1 para modificar o número de canais da ativação) e *upsampling* (para reverter o processo de *pooling*) e por fim a soma elemento-a-elemento dos dois ramos. A saída P_n das camadas é produzida por uma última convolução dos valores obtidos.

Observa-se que os níveis P_n são criados a partir das camadas de ativação do *backbone* original. Cada nível é produzido de maneira sequencial, formando saídas com resoluções crescentes e nível de informação semântica elevado. Dessa forma, a detecção de objetos pode ser feita em todas os níveis, possibilitando a detecção de objetos em múltiplas escalas de maneira eficiente e precisa.

2.9 Âncoras

Algoritmos de detecção de objetos consistem na avaliação de múltiplas regiões de uma imagem, determinando se dada região contém ou não os objetos de interesse. O uso de âncoras permite realizar essa tarefa de maneira rápida e precisa para um grande número de regiões.

Âncoras são um conjunto de *bounding boxes* com altura e largura pré-definidas, alocadas de forma regular sobre a imagem original. A detecção de objetos é feita a partir da classificação e ajuste fino de cada âncora, indicando que a região da imagem original correspondente à esta âncora contém o objeto de interesse.

O processo de detecção é feito gerando-se uma relação entre cada âncora na imagem original e a saída da rede. Cada âncora definida sobre a imagem original é considerada uma região de referência, e o objetivo da rede é classificar (definir se a âncora corresponde à um objeto) e ajustar (corrigir a posição da âncora em relação ao objeto).

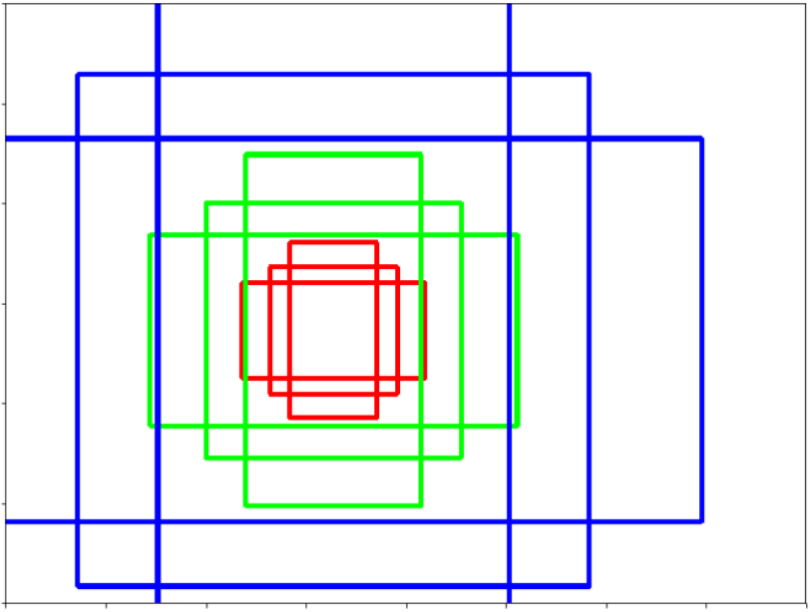

Figura 2.16: Exemplos de âncoras posicionadas em torno do centro da imagem. Aqui é apresentado um conjunto de 9 âncoras centradas num mesmo ponto, com tamanhos e formatos diferentes. Esse conjunto é repetido por toda a imagem, de forma que o conjunto de âncoras resultante cobre a imagem inteira.

As predições feitas pela rede são realizadas em relação às âncoras dispostas, como na Figura 2.16. Para cada âncora dentro de um conjunto de N possíveis, são previstos os seguintes valores:

- Classificação: classifica-se cada âncora entre as K possíveis classes, resultando em um vetor com $N \times K$ probabilidades.

- Regressão: são refinados a posição e tamanho de cada âncora, visando ajustar o posicionamento delas em relação ao objeto verdadeiro na imagem. O resultado é um *offset* para cada uma das quatro coordenadas da âncora, formando um vetor com $N \times 4$ elementos.

A classificação e regressão das âncoras é feito com base nas ativações da rede convolucional [36, 6], como mostra a Figura 2.17. A camada

de ativação mostrada é o resultado de uma única rede convolucional, mas o mesmo princípio pode ser aplicado para múltiplas saídas, como no caso de uma estrutura FPN.

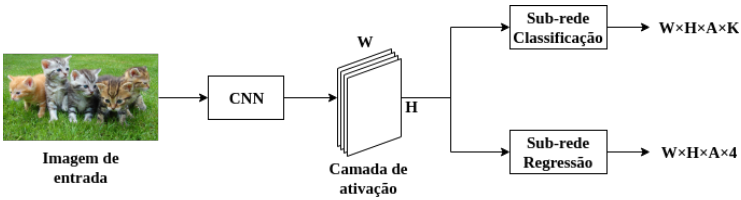

Figura 2.17: Classificação e regressão de âncoras correspondentes à uma imagem de entrada. As sub-redes de classificação e regressão avaliam as âncoras na imagem original a partir de uma camada de ativação da rede convolucional principal, resultando em $W \times H \times A \times (K+4)$ valores previstos.

A classificação e regressão são realizadas com o uso de duas sub-redes que avaliam a camada de ativação. A implementação dessas sub-redes é geralmente convolucional, com um número muito menor de camadas do que a rede principal.

Supondo uma camada de ativação com resolução $W \times H$ (e um número qualquer de canais), serão avaliadas $W \times H$ regiões na imagem original, cada uma com A âncoras correspondentes. Desta forma, os resultados das sub-redes de classificação e regressão são vetores de tamanho $W \times H \times A \times K$ e $W \times H \times A \times 4$ respectivamente.

Percebe-se que o número total de âncoras na imagem pode ser diferente para aplicações distintas. Os principais parâmetros que modificam a quantidade de âncoras são referentes à resolução da camada de ativação ($W \times H$), que regula o número de regiões avaliadas, e A, que define o número de âncoras por região.

CAPÍTULO 3

RetinaNet

Nesse capítulo será apresentado o conceito de redes de detecção de um e dois estágios. Veremos as características desses dois métodos, e enfim será apresentada a estrutura de detecção RetinaNet, que será utilizada nas simulações desse trabalho.

Utiliza-se a implementação em Python/Keras da estrutura RetinaNet disponível em `https://github.com/fizyr/keras-retinanet` [39] como referência para o desenvolvimento deste trabalho.

3.1 Redes de dois e um estágio

Detectores de dois estágios foram durante muito tempo as estruturas com melhor desempenho em termos de acurácia disponível na literatura. Essas redes utilizam dois estágios para a detecção, primeiro buscando as regiões de interesse (*candidate proposal*), e em seguida refinando as propostas a partir de uma rede de classificação.

Muito trabalho foi realizado nessas estruturas, tanto em questão de desempenho quando em complexidade computacional. Os trabalhos em Fast-RCNN [37] e Faster-RCNN [6] resolveram muitos dos desafios da estrutura original R-CNN [40] pelo uso de redes convolucionais para

realizar a seleção de regiões de interesse, aumentando drasticamente a velocidade de processamento da estrutura.

Detectores de um estágio vem em uma vertente paralela a esse desenvolvimento, sugerindo formas de realizar o processamento em uma única etapa. Redes como YOLO [41] e SSD [38] apresentam resultados promissores, com complexidade computacional muito menor, porém com uma taxa de erros maior que as contrapartidas de dois estágios. Devido à essas características, as estruturas de um estágio ficavam reservadas principalmente para problemas em tempo real onde velocidade de avaliação é um quesito de maior importância [42].

Uma das principais dificuldades observadas por redes de um estágio como SSD está em trabalhar com a detecção densa de objetos. Como todo o processamento será feito em uma etapa, todas as possibilidades de objetos (com tamanhos, localização e proporções diferentes) devem ser avaliados simultaneamente. Isso pode resultar em até centenas de milhares de regiões diferentes, e trabalhar com todas essas regiões ao mesmo tempo sem o uso de um método de *candidate proposal* se torna complicado devido ao efeito de desbalanceamento de classes.

Desbalanceamento de classes é um problema que aparece quando o número de objetos de uma classe é muito maior do que os das outras. Essa situação resulta em um desempenho insatisfatório quando se considera a acurácia média ao longo de todas as classes, pois nesse caso o treinamento da rede tende se tornar enviesado em prol das classes mais numerosas.

No caso de detectores de um estágio, isso ocorre porque o número de objetos na imagem é muito menor do que o número de predições realizadas. Dessa forma, a grande maioria do treinamento corresponde ao *background* e não aos objetos de interesse. Esse problema não aparece em redes de dois estágios, devido ao uso de técnicas de *candidate proposal*, que filtram as regiões de interesse para conjuntos equilibrados de classes.

3.2 RetinaNet

RetinaNet é uma rede de detecção de um estágio, que busca alcançar o desempenho das redes de dois estágios e ainda assim manter a complexidade computacional reduzida.

	Backbone	**AP**
Two-stage methods		
Faster R-CNN+++ [43]	ResNet-101-C4	34.9
Faster R-CNN w FPN [36]	ResNet-101-FPN	36.2
Faster R-CNN by G-RMI [44]	Inception-ResNet-v2	34.7
Faster R-CNN w TDM [45]	Inception-ResNet-v2-TDM	36.8
One-stage methods		
YOLOv2 [41]	DarkNet-19	21.6
SSD513 [38]	ResNet-101-SSD	31.2
DSSD513 [46]	ResNet-101-DSSD	33.2
RetinaNet [7]	**ResNet-101-FPN**	**39.1**
RetinaNet [7]	**ResNeXt-101-FPN**	**40.8**

Tabela 3.1: Tabela de comparação de desempenho para diferentes estruturas de detecção [7]. Avaliando-se o banco de dados COCO [8], com a métrica de precisão média (AP) das predições feitas por um único modelo para cada estrutura. Observa-se que o desempenho da estrutura RetinaNet é inclusive superior ao encontrado com outras redes de dois estágios.

Essa estrutura compartilha diversas características com outras redes de um estágio (âncoras, *Pyramid Feature Network*), porém implementa uma função de perda inovadora capaz resolver o problema de desbalanceamento de classes.

O desempenho da estrutura RetinaNet chega a ser superior ao encontrado em outras redes de detecção de dois estágios, como mostra a Tabela 3.1. Já o tempo de execução, como mostra a Figura 3.1, é menor que o de outras redes de um estágio para valores de precisão similares.

A principal contribuição da rede RetinaNet é no desenvolvimento da função de Perda Focal [7]. Essa função tem como objetivo diminuir o peso do desbalanço entre as âncoras *foreground* e *background* no treinamento da rede, introduzindo um fator exponencial que diminui a perda de âncoras já bem classificadas.

Para uma âncora com classificação y e predição p, perda Focal é

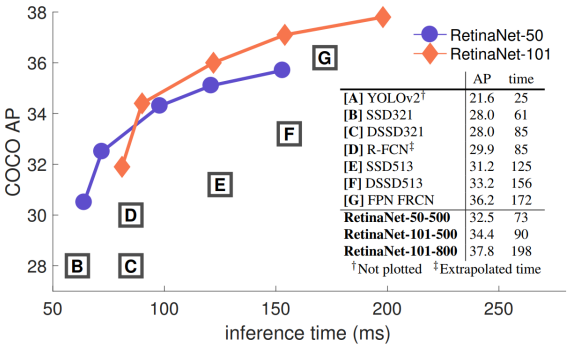

Figura 3.1: Tempo de execução (*inference time*) versus precisão (AP) no banco de dados COCO. A rede RetinaNet consegue um desempenho superior à todas as outras estruturas, inclusive a rede de dois estágios Faster R-CNN [6]. Os pontos da curva são gerados a partir da avaliação das imagens em resoluções diferentes (cinco escalas entre 400 e 800 píxeis).

definida como:

$$p_t = \begin{cases} p, & se\ y{=}1 \\ 1-p, & se\ y{=}0 \end{cases} \qquad (3.1)$$

$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t)$$

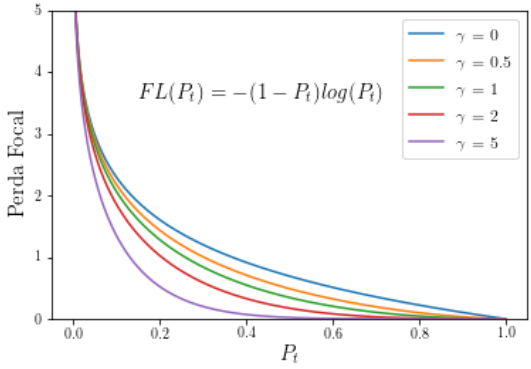

Figura 3.2: Perda Focal para diferentes valores de γ. Nota-se que para o caso $\gamma = 0$, obtemos a função de perda entropia cruzada. A principal observação aqui é o fato de que para a perda observada para amostras bem classificadas ($P \to 1$) é significativamente menor do que para a tradicional entropia cruzada.

A perda Focal é parametrizada por um fator γ, como mostra a Figura 3.2, que define a magnitude do amortecimento da função.

Por exemplo, suponha uma aplicação que contém 10.000 âncoras, onde 99% delas correspondem ao background e 1% ao foreground. Se classificarmos todas as âncoras igualmente, com probabilidade $P = 0.1$ (considerando 0 como background), estaremos acertando a predição para todo o *background* e errando todas as predições dos objetos de interesse. Entretanto, como a distribuição de classes é extremamente desbalanceada o erro médio resultante é baixo, apesar de que o resultado é inútil para um detector.

O resultado disso (Tabela 3.2) é que o treinamento de uma rede de um estágio com o uso de perda entropia cruzada fica completamente controlado pelas âncoras que correspondem ao *background*, enquanto as âncoras de interesse se tornam pouco relevantes para o treinamento.

Por outro lado, com o uso da perda Focal, a classificação das âncoras de interesse continua sendo influencial para o treinamento da rede, já que a perda referente às âncoras *background* trivialmente classificadas se tornam menos relevantes.

	CE	FL
Background, Nb=9900	1042.9	10.4
Foreground, Nf=100	230.2	186.5
Total	1273.1	196.9

Tabela 3.2: Comparação dos valores de perda para as funções Entropia Cruzada (CE) e Perda Focal (FL) com $\gamma = 2$. Considerando um total de $N = 10.000$ âncoras, vemos que para o caso CE a perda total é dominada pelos termos relacionados às âncoras background, mesmo que essas já estejam bem classificadas. Esse problema é solucionado com o uso de FL.

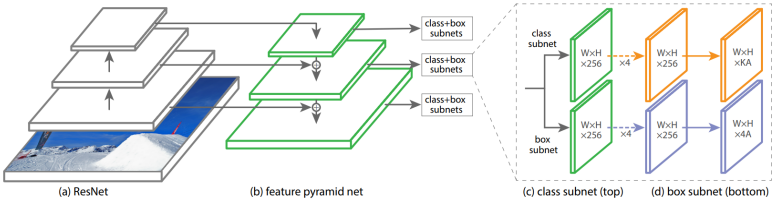

Figura 3.3: Estrutura geral da rede RetinaNet. Consiste de **a)** um backbone convolucional primário, **b)** uma construção Pyramid Feature Network para análise multi-escala, **c)** uma sub-rede para a classificação e **d)** uma para refinamento de cada âncora.

3.3 Estrutura RetinaNet

A rede RetinaNet é construída como mostra a Figura 3.3. É composta basicamente por 4 componentes fundamentais, descritos abaixo:

Backbone: a rede convolucional feedforward responsável por processar a imagem original. Aprende a retirar os descritores necessários para descrever a imagem. Pode ser realizado com diferentes tipos de estruturas, como redes residuais [4], densas [47] e estruturas como InceptionNet [48] e EfficientNet [49], etc.

Feature Pyramid Network (FPN): componente responsável por aumentar a capacidade da rede de trabalhar com objetos em escalas diferentes. Na implementação original [7], são utilizados 5 níveis de resolução, identificados como $\{P3, P4, P5, P6, P7\}$, onde cada nível recebe âncoras com resoluções básicas $\{32^2, 64^2, 128^2, 256^2, 512^2\}$, respectivamente. Por exemplo, o nível P3, que trabalha com a menor resolução,

recebe as âncoras com tamanho $32 * 32$ e todas as suas variações.

Subrede de classificação: utiliza as ativações de cada nível do FPN para classificar cada uma das âncoras correspondentes entre as possíveis classes. A mesma sub-rede de classificação é utilizada para os diferentes níveis da FPN, onde cada nível corresponde à um conjunto de âncoras com tamanho específico.

Subrede de refinamento: realiza uma função similar à sub-rede de classificação, porém o resultado é a regressão de um vetor de 4 elementos. Cada uma das âncoras recebe 4 coordenadas de refinamento, que predizem o *offset* entre essa âncora e o *bounding box* verdadeiro. Nota-se que esse refinamento é independente das classes, ou seja, é realizado de maneira global, e a distinção entre os tipos de objetos é feito apenas pela rede de classificação.

As sub-redes de classificação e regressão são convolucionais, compostas geralmente por convoluções com filtros 3x3 comuns. Isso resulta em uma rede completamente convolucional e treinável com o algoritmo backpropagation.

A perda utilizada para a sub-rede de classificação é a perda Focal, descrita anteriormente.

A perda utilizada para a sub-rede de regressão é a perda L_1 suave (Eq. 3.2) aplicada para cada coordenada, porém outras perdas poderiam ser utilizadas,

$$L_1^{smooth} = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & |x| \geq 1 \end{cases} \tag{3.2}$$

CAPÍTULO 4

Métodos na Literatura

O uso de técnicas de aprendizado de máquina para análise de imagens médicas tem crescido rapidamente nos últimos anos. Aplicações em diversas modalidades de imagens são pesquisadas constantemente, obtendo resultados promissores e facilitando o trabalho médico.

Por exemplo, os trabalhos realizados em [19] [20] desenvolvem o uso de técnicas de processamento natural de linguagem (para a geração do banco de dados) e de redes convolucional para a análise de imagens de Raio X torácicos. A rede é capaz de analisar as imagens para diversos tipos de anomalias diferentes, inclusive obtendo desempenho superior ao de especialistas em algumas tarefas.

As publicações de [50] e [18] utilizam técnicas de aprendizado de máquina para diagnóstico de casos de Alzheimer. Os resultados se mostram efetivos na detecção prévia da doença, com o predição feita corretamente até 6 anos antes do diagnóstico final do paciente [50].

O desenvolvimento de ferramentas de auxílio ao diagnóstico de imagens de mamografia é um trabalho que começou já na década de 1990 [21]. Existem diversos trabalhos que utilizam diferentes técnicas aplicadas a classificação e detecção de anomalias, porém o resultado apre-

sentado pelas técnicas tradicionais não se mostra estatisticamente relevante para a melhoria do diagnóstico médico final [51, 52].

Atualmente há muito interesse na aplicação de redes neurais na construção de ferramentas de auxílio ao diagnóstico. Com o crescente desenvolvimento na área de redes neurais, e em particular de redes convolucionais, surge uma nova técnica que pode ser utilizada para aprimorar os resultados obtidos.

As estruturas convolucionais modernas (como ResNet [4]) e detectores (Faster-RCNN [6], RetinaNet [7]) se mostram superiores às técnicas tradicionais em diversas áreas de aplicação, e portanto busca-se aplicar estas mesmas técnicas na análise de imagens médicas.

O trabalho de [53] utiliza métodos baseados em regras (*Rule Based Methods*) para detecção de massas em exames de mamografia, utilizando técnicas de processamento de imagem e uso de descritores. Esse trabalho se restringe a imagens craniocaudais do banco de dados DDSM [10].

A detecção de massas é feita em [54] utilizando descritores e o desenvolvimento de um novo tipo de filtro, chamado *Spiculated Lesion Filters* (SLF). O resultado desse trabalho é relacionado com a detecção de massas espiculadas (nódulos com formato irregular), e é avaliado em um conjunto de 21 imagens do banco de dados DDSM [10] e 50 imagens do banco de dados *mini-MIAS* [55].

Em [56] e [57] é realizado a detecção de massas pelo uso misto de descritores e redes neurais. Em [56], uma rede neural é utilizada para implementar o papel de classificador, analisando o resultado obtido a partir do uso de descritores na imagem original. Já em [57], a rede neural é utilizada para atribuir um nível de suspeita para cada pixel da imagem, com base numa etapa de pré processamento realizada.

O trabalho de Li Shen et al. [58] realiza a detecção de massas e calcificações utilizando uma rede puramente convolucional (treinamento "*end-to-end*"). A estrutura desenvolvida aplica o classificador de uma maneira similar a um filtro janela móvel (*sliding window*) por toda a imagem, produzindo um mapa de calor de classificações. A maior contribuição deste trabalho é o desenvolvimento de um método eficiente para o treinamento da rede. A inferência continua sendo feita na imagem completa, enquanto o treinamento é realizado separadamente, por partes. A rede final pode ser adaptada para um classificador global, que

produz uma predição única para a imagem (existência ou não de uma anomalia). Isso permite que a rede seja aperfeiçoada (*fine tuned*) para um novo banco de dados, com um treinamento semi-supervisionado (amostras que contém apenas classificação global). Isso é interessante para realizar a adaptação de domínio, necessário ao se trabalhar com exames de mamografia provenientes de aparelhos diferentes.

Hwenjin Jung et al. [9] realiza a detecção de massas em exames de mamografia utilizando um detector de um estágio puramente convolucional (RetinaNet [7]). Os resultados mostrados no trabalho são comparáveis ou superiores aos encontrados em outros trabalhos do estado da arte, alcançando alta precisão e sensibilidade nas detecções. O treinamento e avaliação desta rede é feita com os bancos de dados GURO e INbreast [59]. GURO é um conjunto criado internamente pelos desenvolvedores e não é disponibilizado publicamente, enquanto INbreast [59] é acessível via requisição, porém dispõe de uma licença de uso limitada. Desta forma, enquanto este trabalho [9] é a principal referência para o desenvolvimento desse projeto, as comparações de desempenho são feitas com relação à outras publicações que utilizam o mesmo conjunto de imagens (DDSM [10]) que este trabalho.

CAPÍTULO 5

Metodologia

Esse capítulo apresenta os principais métodos utilizados no projeto, bem como as soluções utilizadas para os problemas mais importantes.

5.1 Banco de dados

O conjunto de dados utilizado no projeto é conhecido como CBIS-DDSM (*Curated Breast Imaging Subset of Digital Database for Screening Mammography*) [60]. Este banco de dados dispõe de uma versão revisada dos dados disponíveis originalmente pelo projeto DDSM [10], contendo imagens anotadas de exames de mamografia.

No total, o banco de dados contém 1644 imagens, totalizando 2015 anomalias encontradas. As anomalias são diferenciadas entre massas e calcificações, sendo que as imagens de interesse nesse trabalho são as que contém massas.

Todas as imagens são disponibilizadas com segmentações das regiões de interesse para cada anormalidade encontrada, bem como anotações que caracterizam as regiões. As anotações disponíveis no conjunto de dados são:

- Tipo de anomalia: diferenciando os achados entre massas e calcificações.

- Patologia: diferenciação entre achados malignos e benignos.

- BI-RADS (Breast Imaging Reporting and Data System): padronização que permite avaliar as características de uma lesão em exames de mamografia. Todos os achados de um exame são classificados em uma faixa de valor, que corresponde à periculosidade achada. BI-RADS tem valores entre 0 e 6, de acordo com a tabela 5.1.

- Densidade: avaliação do profissional médico sobre a densidade do tecido presente na região de interesse.

- Sutileza: avaliação do profissional médico sobre a visibilidade da anomalia nesse exame, geralmente relacionado com a posição da lesão e densidade do tecido local.

Tabela 5.1: Descrição do padrão BI-RADS para anomalias em exames de mamografia

BIRADS	**Significado**	**Risco**
0	Exame limitado - avaliação incompleta	Não é possível estimar
1	Exame normal	Muito baixo
2	Alterações benignas	Muito baixo
3	Exame provavelmente benigno	2%
4	Lesão suspeita para câncer	20%
5	Lesão altamente suspeita para câncer	95%
6	Lesão já com diagnóstico de câncer	100%

O conjunto de dados é separado em três grupos, para treinamento, validação e teste (com 1138, 95 e 255 imagens respectivamente). Utiliza-se apenas imagens anotadas com a presença de massas.

5.2 Pré-processamento

Um dos desafios desse projeto é a complexidade computacional necessária para avaliar as imagens. A resolução das imagens é maior do que a maioria das aplicações de visão computacional, tipicamente em torno de 4000x3000 pixels, gerando um problema grave em questão de consumo memória e tempo de processamento.

Realiza-se *downsampling* como forma amenizar esse problema. A região de interesse (ROI) das imagens tem tamanho variável, mas pode chegar a espaços de 100x100 pixels. Empiricamente, percebe-se que a diminuição da resolução da imagem não compromete o desempenho da rede quando o fator de escala é mantido acima de 0.5.

Apesar da diminuição da resolução, ainda não é possível avaliar a imagem por inteiro com a rede convolucional utilizada. Dessa forma, é necessário dividir a imagem em regiões de menor tamanho, e avaliar cada parte independentemente.

Isso é feito dividindo-se a imagem em regiões de 512x512 pixels com uma sobreposição de 256 pixels entre cada região por eixo, como mostra a Figura 5.1. A sobreposição entre regiões da imagem é necessária para evitar problemas de detecção nas bordas das regiões, onde objetos podem ser segmentados pela divisão da imagem, não sendo corretamente detectados em nenhuma delas.

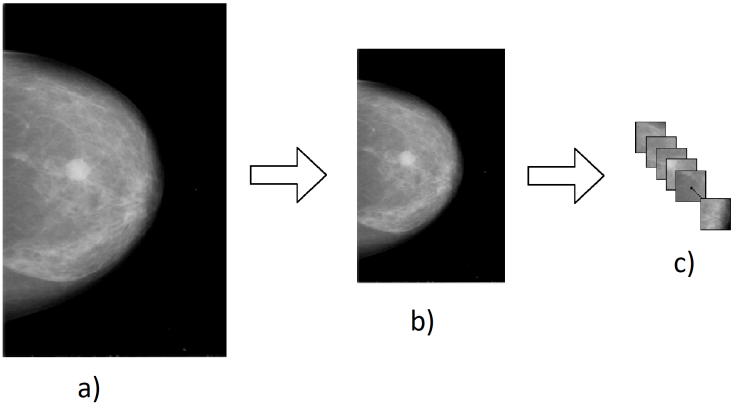

Figura 5.1: a) Imagem com a resolução original. b) Imagem com a resolução reduzida para 50% da resolução original. c) Divisão da imagem em regiões de 512x512 pixels, onde regiões vizinhas tem uma sobreposição de 256 pixels.

5.3 Detecção

A estrutura convolucional utilizada para realizar a detecção foi a rede RetinaNet. A implementação e os hiper-parâmetros utilizados seguem as indicações do artigo original, com certas adaptações feitas de acordo com o presente problema.

- *Backbone network*: foi utilizada uma rede residual para realizar o papel de *backbone* do detector. Foi escolhida especificamente a rede ResNet-50 [4], com fim de balancear o desempenho e complexidade computacional.

- *Transfer-learning*: foi utilizada a técnica de transferência de aprendizado, onde os parâmetros internos do *backbone* utilizado foram pré-treinados no bancos de dados ImageNet [61]. Esse método diminui o tempo de treinamento da rede, dado que os parâmetros iniciais formam um ponto de partida mais apropriado para a estrutura.

- Resolução: a resolução de entrada da rede foi definida em 512x512 pixels.

- Âncoras: são utilizadas âncoras com tamanhos $\{32^2, 64^2, 128^2, 256^2, 512^2\}$ (para os níveis P3 à P7, respectivamente), com escalas $\{2^0, 2^{\frac{1}{3}}, 2^{\frac{2}{3}}\}$ e *aspect ratios* $\{1:2, 1:1, 2:1\}$. Como resultado, temos 9 âncoras diferentes definidas em cada camada da FPN.

- *Focal Loss*: a função de perda de classificação foi parametrizada com $\gamma = 2$ e $\alpha = 0.25$.

Além das duas sub-redes normalmente presentes na rede RetinaNet (regressão e classificação), foi utilizada também uma terceira sub-rede, para a regressão de outras variáveis auxiliares. O objetivo desse procedimento é adicionar as informações de densidade e BI-RADS no processo de treinamento. Desta forma, todas as âncoras presentes na rede contém predições adicionais da densidade do tecido local e do valor BI-RADS da anomalia detectada.

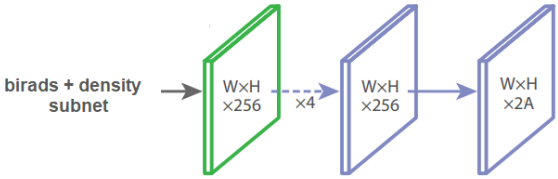

Figura 5.2: Sub-rede de regressão adicional, adicionada em paralelo às sub-redes de classificação e regressão de coordenadas. O objetivo é adicionar as informações de BI-RADS e de densidade de tecido no treinamento da rede. A saída dessa rede são duas variáveis contínuas, que correspondem a predição dos dois valores para cada âncora da imagem.

5.4 Pós-processamento

As detecções feitas pela rede RetinaNet tem como formato a classificação e ajuste de todas as âncoras definidas. É necessário, então, modificar estes resultados para um formato mais adequado para visualização.

Esse procedimento se inicia com duas etapas importantes, que simplificam e ajustam os resultados obtidos:

- A primeira etapa consiste na filtragem das predições com pontuação abaixo de um limiar pré determinado. Esse procedimento remove a grande maioria das predições, nos deixando com algumas centenas de resultados.
- A segunda etapa desse processo envolve traduzir as previsões feitas para as *bounding-boxes* correspondentes. Como toda regressão feita pela rede é realizada em relação à um conjunto de âncoras, os resultados devem ser traduzidos de volta para um sistema de referência absoluto.

Em seguida, é necessário agrupar as previsões feitas. Devido ao grande número de âncoras e regiões avaliadas, há um grande número de detecções feitas para uma mesma parte da imagem, o que exige o uso de um algoritmo de agrupamento nos resultados.

Primeiro recalcula-se a pontuação de cada uma das K âncoras de acordo com a equação abaixo,

$$P_i^{'} = \sum_{j=1}^{j=K} IoU(A_i, A_j) P_j \tag{5.1}$$

onde P_n corresponde à pontuação da âncora An, e IoU (também conhecido como índice de Jaccard) é uma métrica de similaridade entre detecções,

$$IoU(A, B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid} \tag{5.2}$$

Dessa forma, as detecções presentes em grandes grupos tem suas pontuações aumentadas, devido à sobreposição de múltiplas predições.

Em seguida é aplicado o algoritmo NMS (Nom-Maximum Supression) [62], que promove cada grupo de detecções para apenas os representantes com maiores pontuações. Esse método leva em consideração o valor de IoU entre cada par de predições, bem como a pontuação de cada *bounding box* produzida.

O algoritmo NMS prioriza predições com maior pontuação, e para cada par de predições A_j e A_k onde $IoU(A_j, A_k)$ é maior que um limiar, descarta-se a predição com menor score. Esse processo é feito iterativamente até que se obtenha apenas um (ou poucos) representantes para cada agrupamento de predições, como observado na Figura 5.3.

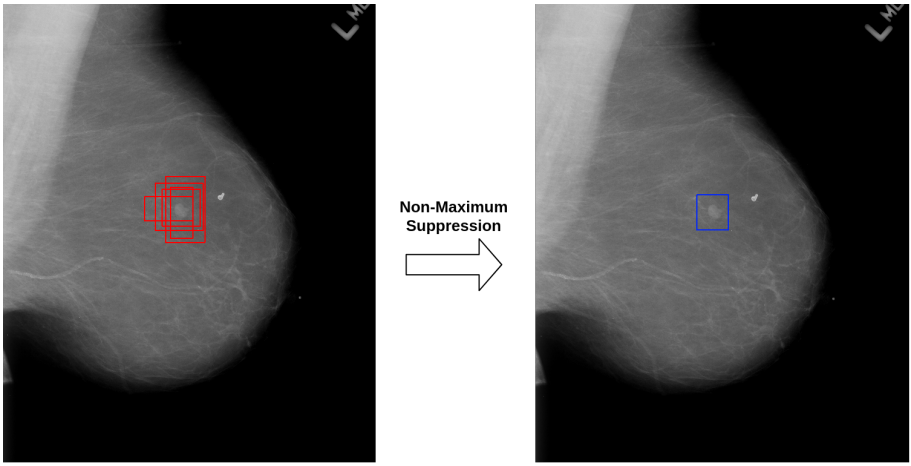

Figura 5.3: Exemplo de caso do algoritmo *Non-Maximum Suppression*, onde apenas uma predição representante do grupo é mantida. A supressão realizada por esse método pode ser ajustada, modificando o valor de limiar para agrupamento.

CAPÍTULO 6

Simulações e Resultados

Neste capítulo são apresentados os resultados das simulações realizadas.

As predições são avaliadas de acordo com o valor de IoU observado (Equação 5.2). Uma predição é considerada correta (verdadeiro positivo) se o valor de IoU em relação à uma anomalia é maior que um determinado limiar, caso contrário a detecção é considerada incorreta (falso positivo). No caso de múltiplas detecções coincidindo com uma mesma anomalia, apenas uma delas é considerada correta, e as outras incorretas. O limiar escolhido é $IoU > 0.2$, valor normalmente utilizado em outros modelos de detecção de massas [9].

Exemplos de predições são mostrados no Apêndice A para diferentes imagens do conjunto de teste utilizado.

A curva FROC (*Free-Response Receiver Operating Characteristic*), na Figura 6.3, traça o desempenho da rede para diferentes limiares de avaliação, com os valores de taxa de verdadeiro positivo e número de falsos positivos por imagem para cada ponto da curva.

As curvas de aprendizado (Figura 6.1 e 6.2) descrevem o desenvolvimento da rede durante o treinamento, em relação à perda observada em cada sub-rede e à taxa de aprendizado.

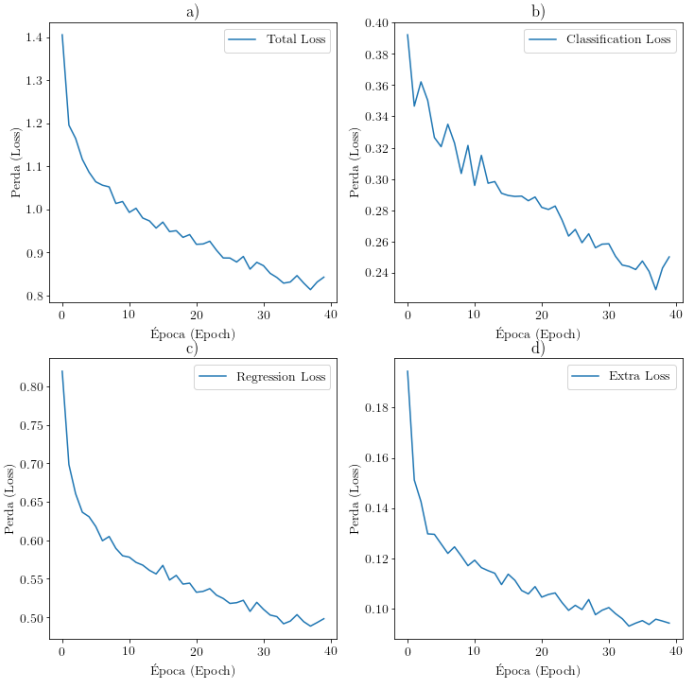

Figura 6.1: Curva de treinamento da rede convolucional, por um total de 40 épocas (epochs). **a)** apresenta a perda total da rede ao longo do tempo, enquanto **b), c) e d)** apresentam a perda de classificação, regressão e extra (BI-RADS e densidade), respectivamente.

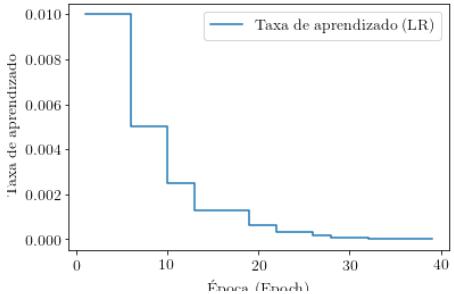

Figura 6.2: Desenvolvimento da taxa de aprendizado durante o treinamento da rede convolucional, durante as 40 épocas de treinamento.

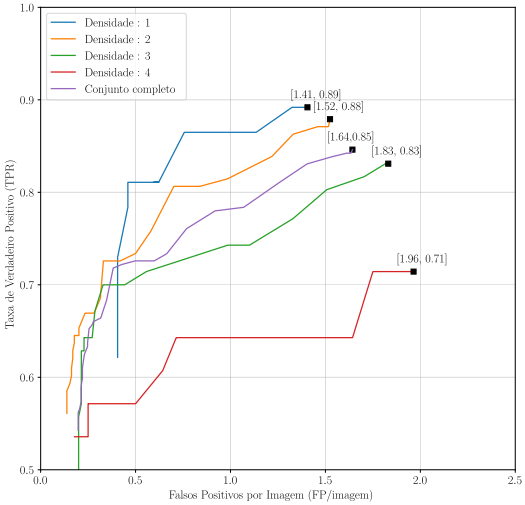

Figura 6.3: Curva de desempenho das predições para diferentes valores de limiares de detecção. A curva apresenta os pontos de TPR (taxa de verdadeiro positivo) em função do número de falsos positivos por imagem. Diferentes curvas são apresentadas, para subconjuntos de imagens com densidades específicas e para o conjunto total de imagens. Observa-se que o desempenho das predições caí de acordo com o aumento da densidade do tecido presente nas imagens.

	TPR@FPPI	Objetivo específico
Este trabalho	0.85@1.64	Detecção de massas
	0.81@1.25	
Sampat et al. [54]	0.88 @ 2.7	Detecção de massas espiculadas
	0.85 @ 1.5	
	0.80 @ 1.0	
Eltonsy et al. [63]	0.92 @ 5.4	Detecção de massas malignas
	0.88 @ 2.4	
	0.81 @ 0.6	
Eltonsy et al. [63]	0.61 @ 5.1	Detecção de massas benignas
	0.58 @ 2.8	
Beller et al. [64]	0.70 @ 8.0	Detecção de massas espiculadas
Campanini et al. [65]	0.80 @ 1.1	Detecção de massas
Dhungel et al. [66]	0.75@4.8	Detecção de massas
	0.70@4.0	

Tabela 6.1: Tabela de comparação de performance para diferentes métodos [9]. Todas as avaliações são referentes a imagens do banco de dados DDSM [10] ou variações deste, com objetivos específicos (tipos de massas detectadas) mostrados na tabela. Métrica utilizada TPR@FPPI (Taxa de verdadeiro positivo, para valores de falso positivos por imagem).

CAPÍTULO 7

Conclusão

De acordo com a literatura atual, sabe-se que as redes convolucionais são o estado da arte na área de detecção de objetos. Outros estudos mostram sucesso na aplicação de redes convolucionais na análise de diversos tipos de imagens médicas, inclusive em exames de mamografia.

Por tal motivo, utilizou-se redes convolucionais profundas para o desenvolvimento de uma ferramenta de auxílio ao diagnóstico de câncer de mama. Alta sensibilidade, poucos falsos positivos e baixo tempo de execução foram buscados utilizando redes de detecção de um estágio, que se mostram bem-sucedidas em diversas aplicações atuais. Especificamente, fez-se uso da estrutura RetinaNet.

A estrutura criada é capaz de detectar massas em mamografias, e os resultados obtidos se mostram comparáveis com outras publicações disponíveis na literatura.

Os resultados obtidos podem ser utilizados como base para o desenvolvimento de uma ferramenta de auxílio médico, permitindo um diagnóstico precoce e mais preciso do câncer de mama.

Trabalhos futuros

Apesar dos resultados obtidos, existem outros desafios presentes antes da implementação em mercado do algoritmo. Os principais fatores que devem ser avaliados são:

(*i*) Validação do algoritmo em imagens de diferentes fontes. Sabe-se que as imagens produzidas por aparelhos diferentes apresentam variações, de forma que pode ser necessário realizar alguma adaptação para que o algoritmo seja agnóstico a fonte da imagem.

(*ii*) Treinamento contínuo. Para que o algoritmo continue a ser aprimorado, é interessante realizar o treinamento da rede com novas imagens. O *feedback* dos usuários (profissionais médicos) é uma fonte de dados de grande valor, podendo ser utilizada para o melhoramento da rede convolucional.

(*iii*) Treinamento semi-supervisionado. A aplicação de técnicas de aprendizado semi-supervisionado permitiria o treinamento fino (*fine tuning*) da rede utilizando apenas imagens classificadas, sem necessidade de anotações médicas precisas, facilitando o processo de treinamento contínuo.

Referências bibliográficas

[1] Christopher Olah. Understanding convolutions. `https://colah.github.io/posts/2014-07-Understanding-Convolutions/`, acesso: 29-06-2019, 2014.

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, arXiv:abs/1409.1556, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[5] Jonathan Hui. Understanding feature pyramid networks for object detection (fpn). `https://medium.com/@jonathan_hui/understanding-feature-pyramid-networks-for-object-detec\tion-fpn-45b227b9106c`, acesso: 29-06-2019, 2018.

[6] R. Girshick S. Ren, K. He and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *In NIPS.*, 2015.

[7] Ross Girshick Kaiming He Piotr Dollár. Tsung-Yi Lin, Priya Goyal. Focal Loss for Dense Object Detection. *arXiv:1708.02002*, 2018.

[8] S. Belongie J. Hays P. Perona D. Ramanan P. Dollar T.-Y. Lin, M. Maire and C. L. Zitnick. Microsoft COCO: Common objects in context. *In ECCV.*, 2016.

[9] Inyeop Lee Minhwan Yoo Junhyun Lee Sooyoun Ham Okhee Woo Jaewoo Kang. Hwejin Jung, Bumsoo Kim. Detection of masses in mammograms using a one-stage object detector based on a deep convolutional neural network. *PLoS ONE 13(9): e0203355.*, 2018.

[10] Bowyer K. Kopans D. Moore R. Kegelmeyer W. P. Heath, M. The Digital Database for Screening Mammography. *Proceedings of the Fifth International Workshop on Digital Mammography 212–218.*, 2001.

[11] INCA. Instituto Nacional de Câncer. Câncer de mama. `https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama`, acesso: 29-06-2019, 2018.

[12] Giulianna Muneratto. O câncer de mama em números. `https://www.femama.org.br/2018/br/noticia/o-cancer-de-mama-em-numeros`, acesso: 29-06-2019, 2019.

[13] GBD 2015 Mortality and Causes of Death Collaborators. Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980-2015: a systematic analysis for the global burden of disease study 2015., 2017.

[14] Ros Mendoza L.H. Merino Bonilla J.A., Torres Tabanera M. Breast cancer in the 21st century: from early detection to new therapies. *Radiologia. 2017 Sep - Oct;59(5):368-379. doi: 10.1016/j.rx.2017.06.003.*, 2017.

[15] Breastcancer.org. What is breast cancer? `https://www.breastcancer.org/symptoms/understand_bc/what_is_bc`, acesso: 29-06-2019, 2018.

[16] American Cancer Society. What is breast cancer? `https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/`, acesso: 29-06-2019, 2018.

[17] Barry A. Miller, Eric J. Feuer, and Benjamin F. Hankey. Recent incidence trends for breast cancer in women and the relevanceof early detection: An update. *CA: A Cancer Journal for Clinicians*, 43(1):27–41, 1993.

[18] Goo-Rak Kwon. Debesh Jha. Diagnosis of alzheimer's disease using a machine learning technique. *Alzheimer's Dementia: The Journal of the Alzheimer's Association, Volume 13, Issue 7, P1538.*, 2017.

[19] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning, 2017.

[20] Michael Ko et al. Jeremy Irvin, Pranav Rajpurkar. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019.

[21] Matthias Elter and Alexander Horsch. Cadx of mammographic masses and clustered microcalcifications: A review. *Medical Physics*, 36(6Part1):2052–2068, 2009.

[22] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1. IEEE, 2001, pp. I–I.*

[23] P. Sabzmeydani and G. Mori. Detecting pedestrians by learning shapelet features. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[25] David G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision.*, 2:1150–1157, 1999.

[26] Kurt Konolige Gary R. Bradski Ethan Rublee, Vincent Rabaud. Orb: An efficient alternative to sift or surf. *in ICCV*, pages 2564–2571, 2011.

[27] Franco Scarselli and Ah Chung Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11(1):15 – 37, 1998.

[28] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[30] Christopher Williams et al. John McGonagle, George Shaikouski. Backpropagation. `https://brilliant.org/wiki/backpropagation/`, urldate = 29-06-2019, 2018.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[33] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. *Proceedings of ICML, volume 27*, pages 807–814, 06 2010.

[34] Abien Fred M. Agarap. Deep Learning using Rectified Linear Units (ReLU). *arXiv:1803.08375v2 [cs.ML]*, 2019.

[35] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[36] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[37] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[38] D. Erhan C. Szegedy W. Liu, D. Anguelov and S. Reed. SSD: Single shot multibox detector. *In ECCV.*, 2016.

[39] et al. Hans Gaiser; Maarten de Vries; Valeriu Lacatusu. Keras implementation of retinanet object detection. `https://github.com/fizyr/keras-retinanet`, acesso: 20-06-2019, version:fizyr/keras-retinanet 0.5.1, 2019.

[40] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[41] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[42] Petru Soviany and Radu Tudor Ionescu. Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction. *CoRR*, abs/1803.08707, 2018.

[43] S. Ren K. He, X. Zhang and J. Sun. Deep residual learning for image recognition. *In CVPR*, 2016.

[44] C. Sun M. Zhu A. Korattikara A. Fathi I. Fischer Z. Wojna Y. Song S. Guadarrama J. Huang, V. Rathod and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *In CVPR*, 2017.

[45] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *CoRR*, abs/1612.06851, 2016.

[46] A. Ranga A. Tyagi C.-Y. Fu, W. Liu and A. C. Berg. DSSD: Deconvolutional Single Shot Detector. *arXiv:1701.06659*, 2016.

[47] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[48] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[49] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[50] Yiming Ding, Jae Ho Sohn, and et al. Kawczynski. A deep learning model to predict a diagnosis of alzheimer disease by using 18f-fdg pet of the brain. *Radiology*, 290(2):456–464, 2019. PMID: 30398430.

[51] Joshua J. Fenton, Stephen H. Taplin, Patricia A. Carney, Linn Abraham, Edward A. Sickles, Carl D'Orsi, Eric A. Berns, Gary Cutter, R. Edward Hendrick, William E. Barlow, and Joann G. Elmore. Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*, 356(14):1399–1409, 2007. PMID: 17409321.

[52] Helga S. Marques R. Edward Hendrick Martin J. Yaffe Elodia B. Cole, Zheng Zhang and Etta D. Pisano. Impact of Computer-Aided Detection Systems on Radiologist Accuracy With Digital Mammography. *American Journal of Roentgenology 203:4, 909-916.*, 2014.

[53] Elmaghraby AS. Eltonsy NH, Tourassi GD. A Concentric Morphology Model for the Detection of Masses in Mammography. *IEEE Transactions on Medical Imaging, Vol. 26, No. 06.*, 2007.

[54] Mehul P. Sampat, Alan C. Bovik, Gary J. Whitman, and Mia K. Markey. A model-based framework for the detection of spiculated masses on mammographya). *Medical Physics*, 35(5):2110–2123, 2008.

[55] J. Suckling et al. The mammographic images analysis society digital mammogram database. *Exerpta Medica 1069, 375–378.*, 1994.

[56] R. Bellotti, F. De Carlo, S. Tangaro, G. Gargano, G. Maggipinto, M. Castellano, R. Massafra, D. Cascio, F. Fauci, R. Magro, G. Raso, A. Lauria, G. Forni, S. Bagnasco, P. Cerello, E. Zanon, S. C. Cheran, E. Lopez Torres, U. Bottigli, G. L. Masala, P. Oliva, A. Retico, M. E. Fantacci, R. Cataldo, I. De Mitri, and G. De Nunzio. A completely automated cad system for mass detection in a large mammographic database. *Medical Physics*, 33(8):3066–3075, 2006.

[57] Karssemeijer N. Hendriks J. H. C. L. Brake, G. M. te. An automatic method to discriminate malignant masses from normal tissue in digital mammograms. *Physics in Medicine and Biology, 45(10), 2843–2857.*, 2000.

[58] Li Shen. End-to-end training for whole image breast cancer diagnosis using an all convolutional design. *CoRR*, abs/1708.09427, 2017.

[59] Domingues I Cardoso A Cardoso MJ Cardoso JS. Moreira IC, Amaral I. . inbreast: toward a full-field digital mammographic database. *Academic radiology. 19(2):236–248.*, 2012.

[60] Assaf Hoogi Kanae Kawai Miyake Mia Gorovoy Daniel L. Rubin. Rebecca Sawyer Lee, Francisco Gimenez. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data volume 4, Article number: 170177.*, 2017.

[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[62] Younes Bensouda Mourri Andrew Ng, Kian Katanforoosh. Non-max suppression. `https://www.coursera.org/lecture/convolutional-neural-networks/non-max-suppression-dvrjH`, acesso: 29-06-2019.

[63] N. H. Eltonsy, G. D. Tourassi, and A. S. Elmaghraby. A concentric morphology model for the detection of masses in mammography. *IEEE Transactions on Medical Imaging*, 26(6):880–889, June 2007.

[64] Michael Beller, Rainer Stotzka, Tim Oliver Müller, and Hartmut Gemmeke. An example-based system to support the segmentation of stellate lesions. In Hans-Peter Meinzer, Heinz Handels, Alexander Horsch, and Thomas Tolxdorff, editors, *Bildverarbeitung für die Medizin 2005*, pages 475–479, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[65] Renato Campanini, Danilo Dongiovanni, Emiro Iampieri, Nico Lanconelli, Matteo Masotti, Giuseppe Palermo, Alessandro Riccardi, and Matteo Roffilli. A novel featureless approach to mass detection in digital mammograms based on support vector machines. *Physics in Medicine and Biology*, 49(6):961–975, feb 2004.

[66] Neeraj Dhungel, Gustavo Carneiro, and Andrew P. Bradley. Automated mass detection in mammograms using cascaded deep learning and random forests. *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2015.

APÊNDICE A

Resultados das simulações

As figuras abaixo mostram exemplos de predições obtidas com as simulações.

As imagens são retiradas do conjunto de teste, escolhidas de forma a demonstrar os resultados para imagens com diferentes características. São apresentados resultados corretos e incorretos, para uma demonstração mais completa das características e limitações do algorítimo.

Os valores de densidade e BI-RADS são referentes às anotações presentes no banco de dados, resultados da avaliação médica sobre as características da imagem e da lesão.

As marcações em azul demonstram as anotações médicas disponibilizadas no banco de dados. As marcações em vermelho representam as predições feitas pela rede, com a pontuação correspondente anotada abaixo.

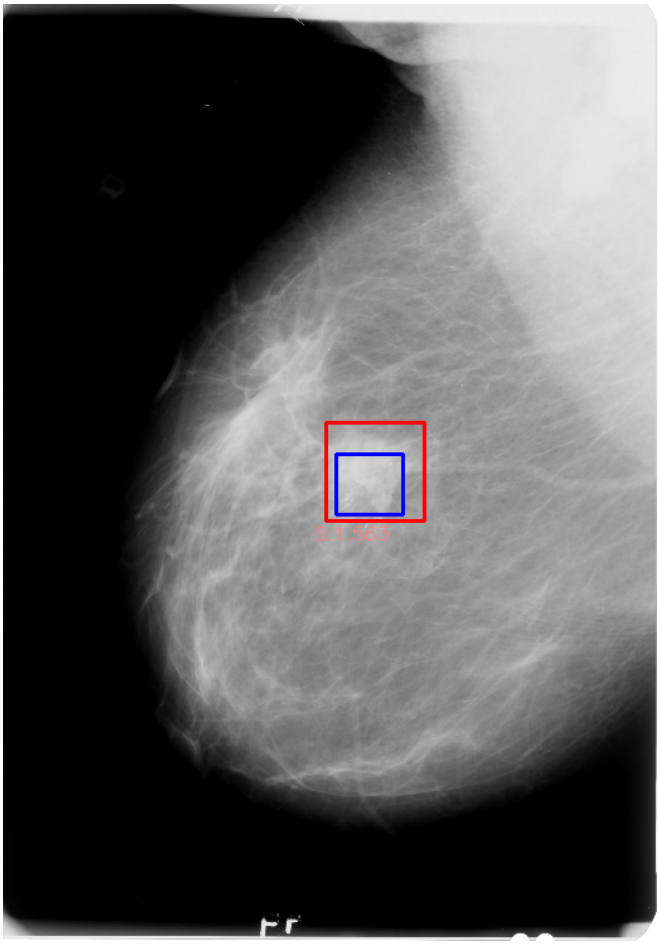

Figura A.1: Detecção em uma imagem com densidade nível 1, anomalia classificada com BI-RADS nível 4. Objeto detectado com $IoU = 0.417$ e $score = 1.563$.

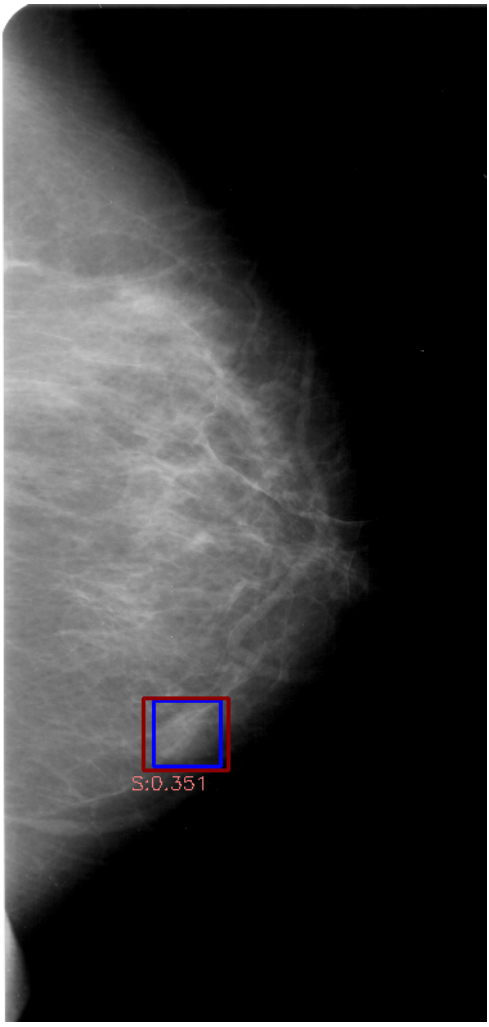

Figura A.2: Detecção em uma imagem com densidade nível 3, anomalia classificada com BI-RADS nível 4. O IoU entre a detecção e anotação é 0.727. O *score* da detecção é igual a 0.351.

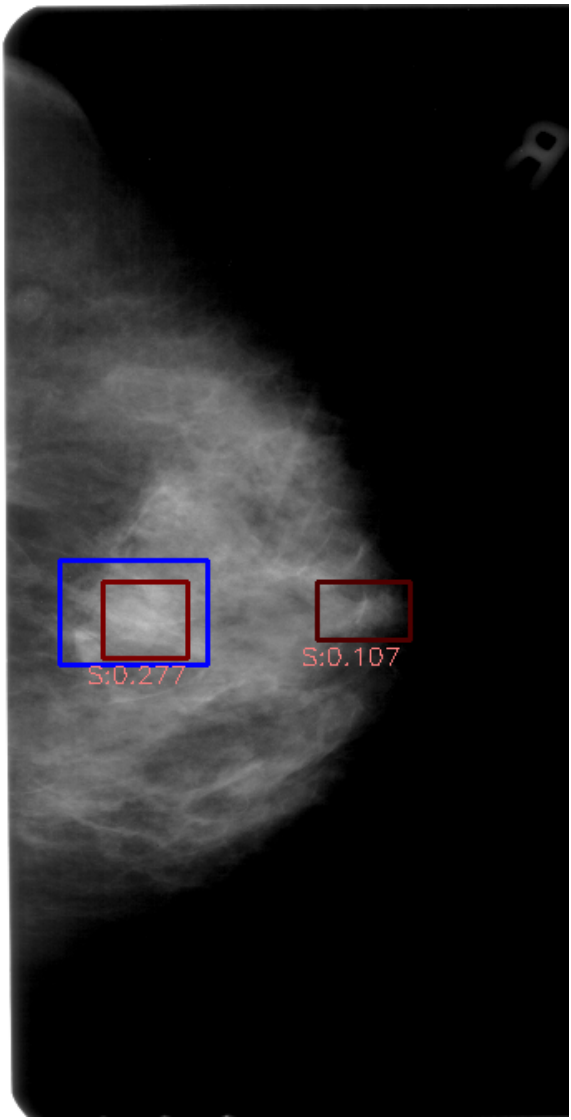

Figura A.3: Detecção em uma imagem com densidade nível 3, anomalia classificada com BI-RADS nível 4. O IoU entre a detecção correta e anotação é 0.417. O *score* da detecção correta é igual a 0.277. Observa-se um falso positivo, com *score* igual a 0.107.

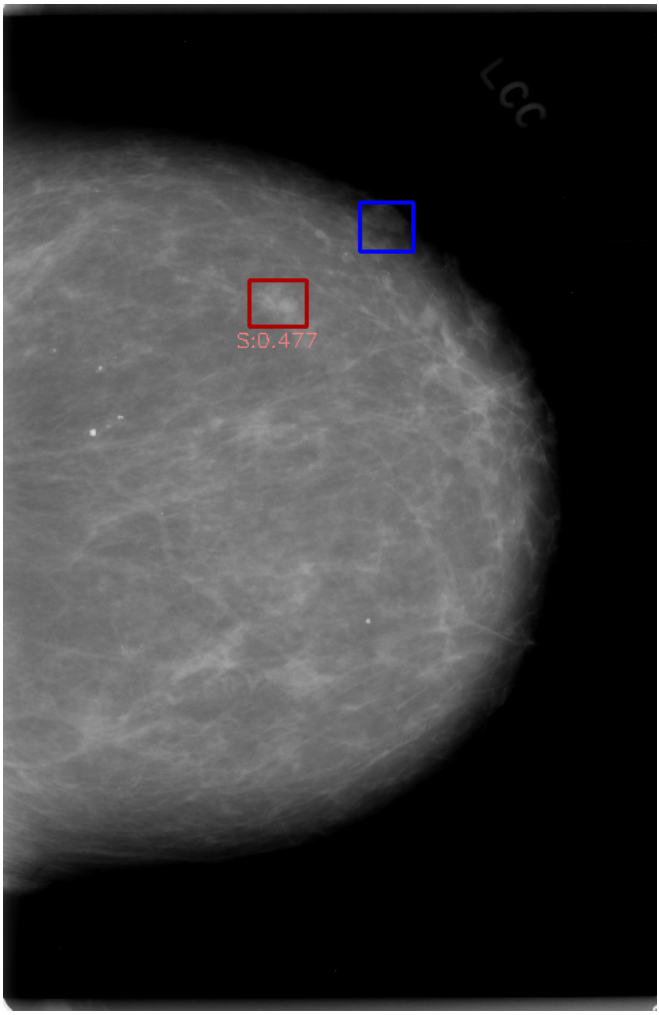

Figura A.4: Resultado em uma imagem com densidade nível 3, e anomalia classificada com BI-RADS nível 4. Observa-se que a anomalia não foi detectada, e um falso positivo com *score* igual a 0.477 é presente.

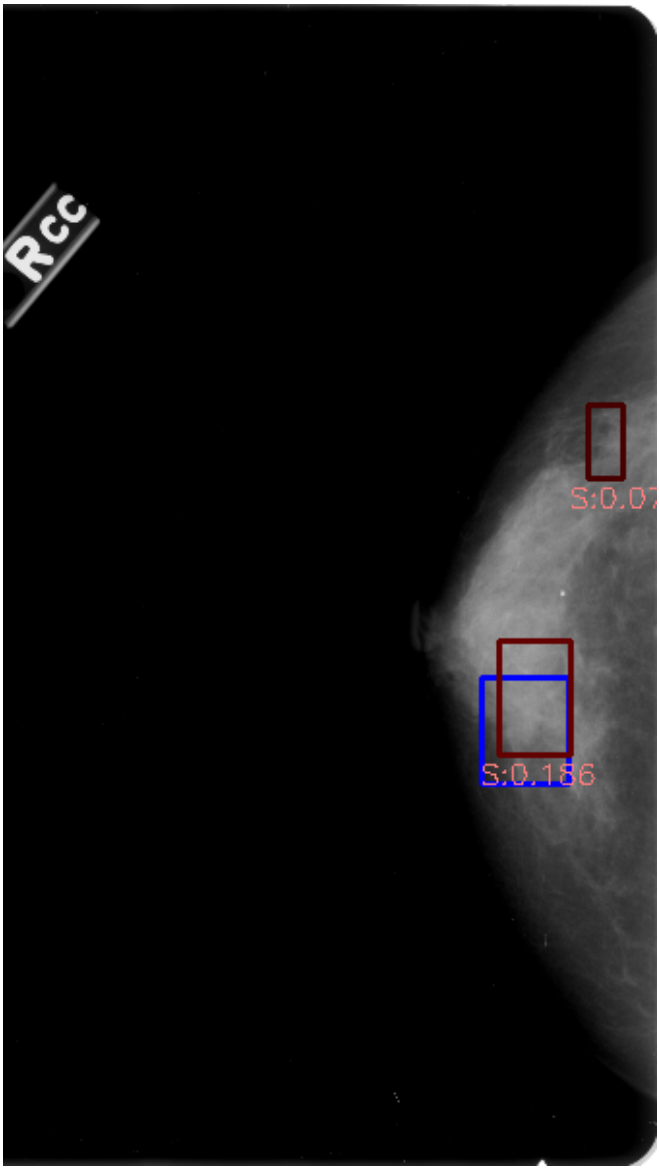

Figura A.5: Detecção em uma imagem com densidade nível 3, anomalia classificada com BI-RADS nível 3. O IoU entre a detecção correta e anotação é 0.452. O *score* da detecção correta é igual a 0.186. Observa-se um falso positivo, com *score* igual a 0.07.

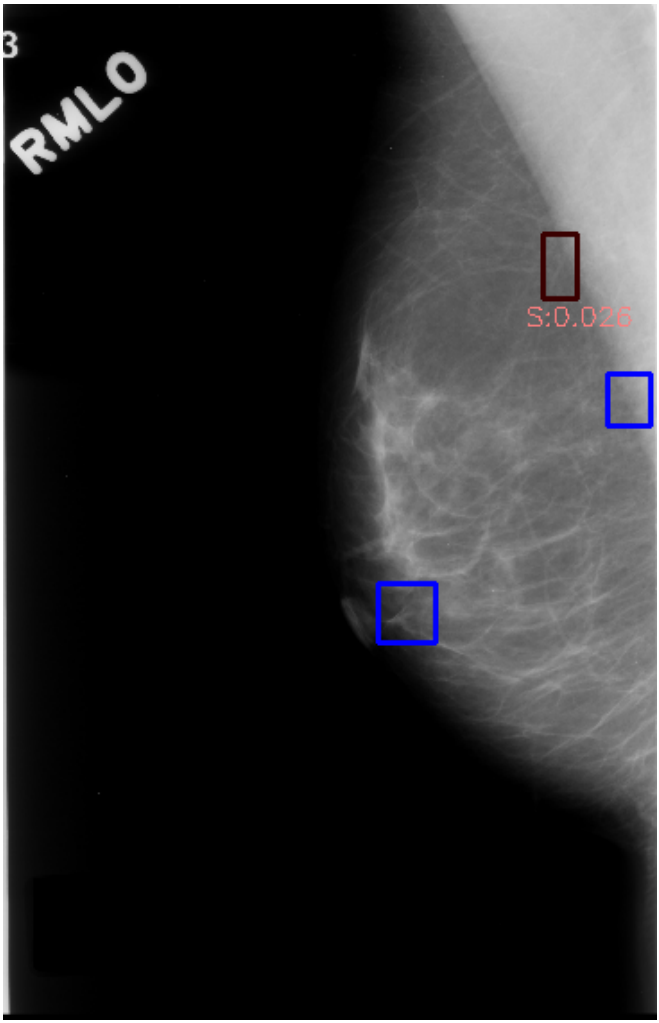

Figura A.6: Resultado em uma imagem com densidade nível 2, e anomalia classificada com BI-RADS nível 3. Observa-se que a anomalia não foi detectada, e um falso positivo com *score* igual a 0.477 é presente.

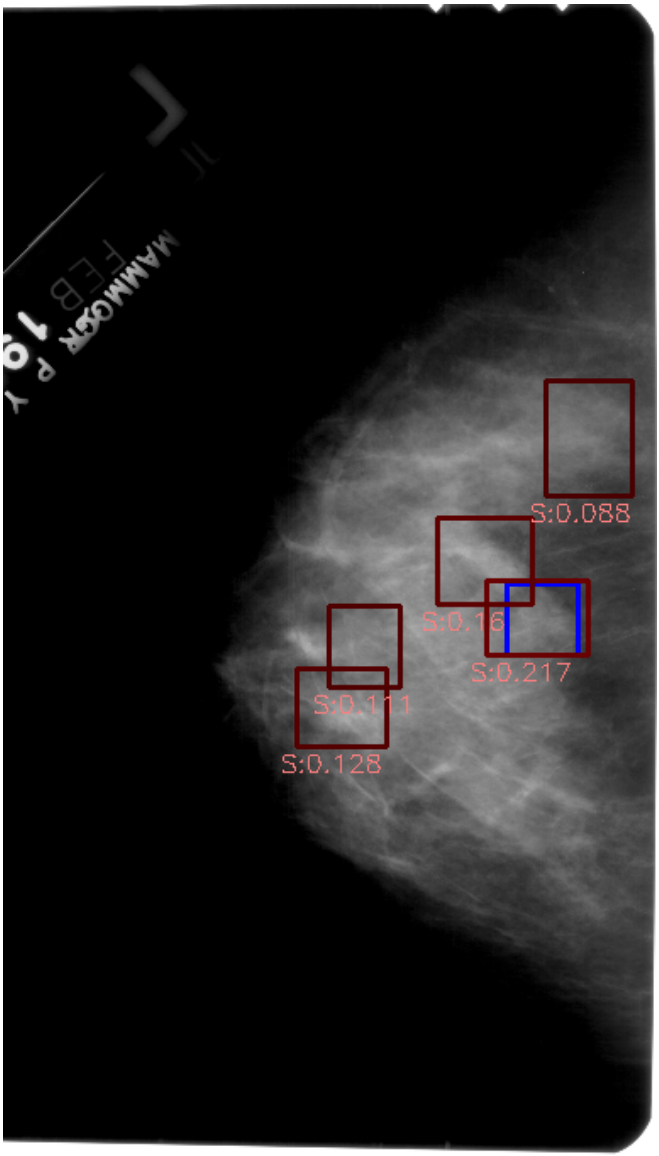

Figura A.7: Resultado em uma imagem com densidade nível 4, e anomalia classificada com BI-RADS nível 4. Observa-se a presença de diversos falsos positivos, devido às características do tecido da mama. A anomalia foi detectada corretamente, com IoU igual a 0.645.

Figura A.8: Resultado em uma imagem com densidade nível 4, e anomalia classificada com BI-RADS nível 4. Apesar da densidade alta do tecido, a anomalia foi detectada corretamente com $IoU = 0.572$ e $score = 0.929$.

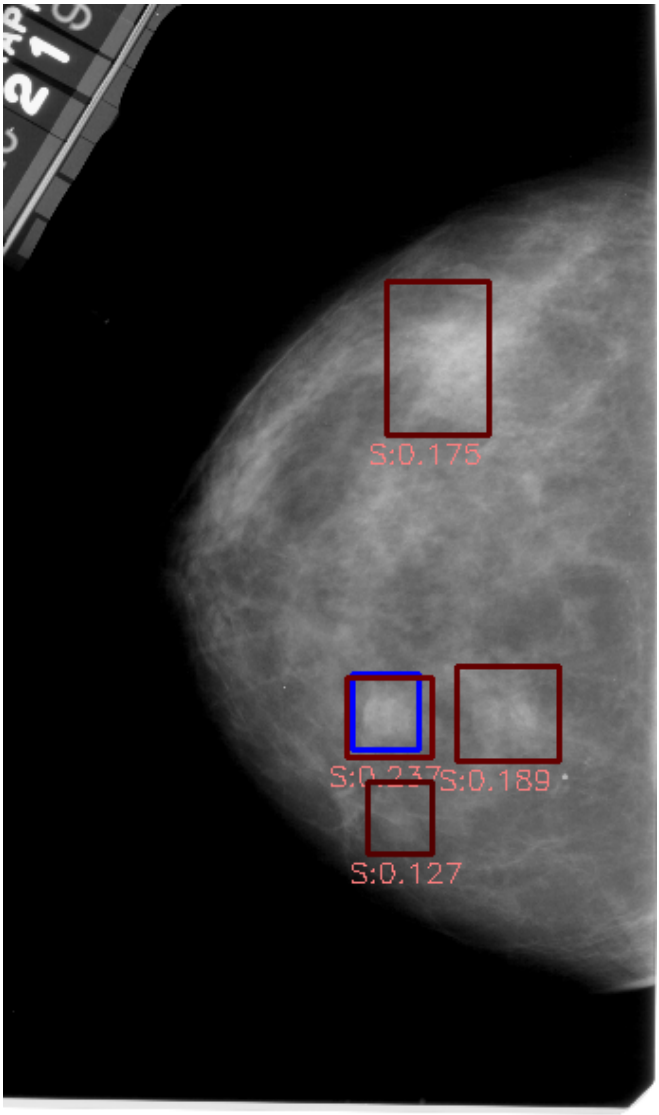

Figura A.9: Resultado em uma imagem com densidade nível 4, e anomalia classificada com BI-RADS nível 4. Nota-se que a anomalia foi detectada corretamente com $IoU = 0.685$ e $score = 0.237$, mas há presença de 3 falsos positivos.

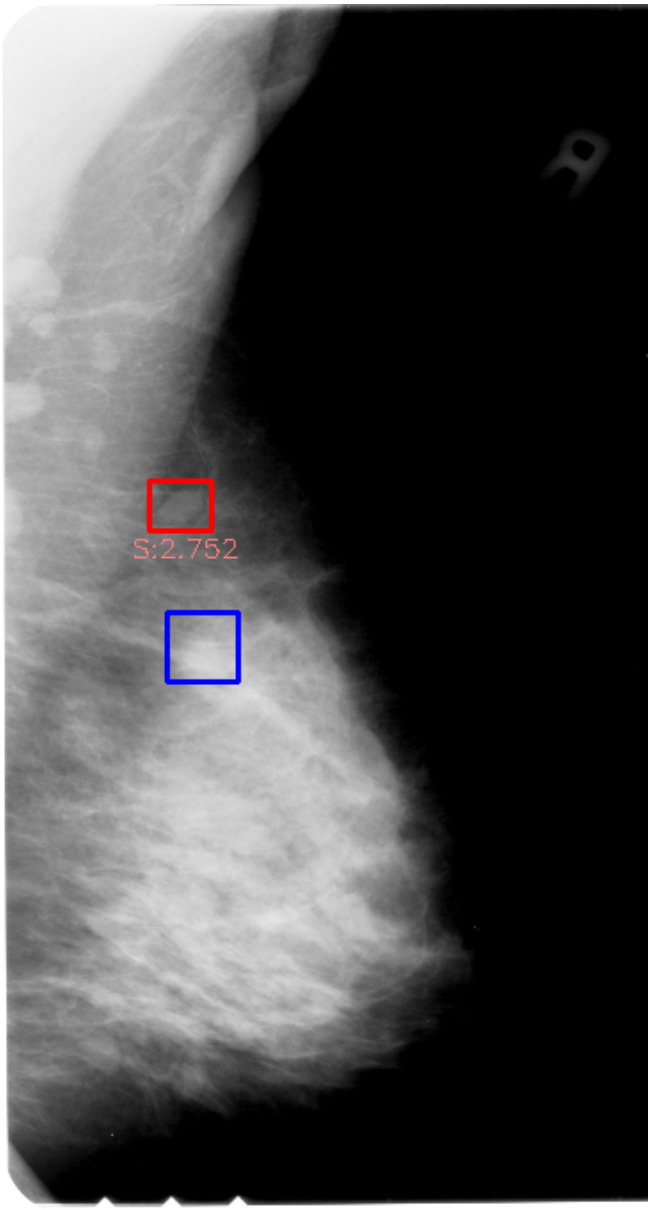

Figura A.10: Resultado em uma imagem com densidade nível 4, e anomalia classificada com BI-RADS nível 4. Assume-se que a anomalia não foi detectada devido à mesclagem do objeto com o tecido de alta densidade ao seu redor.