UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE COMPUTAÇÃO

Exame de Qualificação de Mestrado

CLASSIFICAÇÃO DE IMAGENS RADIOGRÁFICAS PARA DETECÇÃO DE ANORMALIDADES MUSCULOESQUELÉTICAS

Candidato: Leodécio Braz da Silva Segundo

Orientador: Prof. Dr. Zanoni Dias

Coorientador: Prof. Dr. Hélio Pedrini

2019

Resumo

Distúrbios musculoesqueléticos representam um grande problema de saúde que afeta uma ampla parcela da população. Em geral, exames de imagens de raios-X são utilizados para averiguar se existe alguma anormalidade em algum osso ou músculo e localizá-la, em caso positivo. Dessa forma, sistemas de auxílio ao diagnóstico médico (*Computer Aided Diagnosis*, CAD) são ferramentas importantes para ajudar na tarefa de interpretação das imagens e classificação das mesmas. Neste contexto, Redes Neurais Convolucionais vêm sendo amplamente utilizadas na área de Visão Computacional e representam o estado da arte para a tarefa de classificação, dada sua capacidade de aprender características relevantes das imagens e classificá-las com precisão. Neste trabalho, esta tarefa possui um desafio a mais devido à grande variabilidade de formas e posições das imagens radiográficas. Este projeto visa desenvolver uma metodologia eficaz para a tarefa de classificação de imagens radiográficas. Alguns experimentos e análises preliminares foram realizados, a fim de obtermos mais informações sobre a base de dados a ser utilizada.

1. Introdução

Neste capítulo, caracterizamos o problema a ser investigado, apresentamos os principais objetivos e contribuições do trabalho, as questões de pesquisa, bem como a organização do texto.

1.1 Caracterização do Problema

Distúrbios musculoesqueléticos, muitas vezes caracterizados por dores agudas ou crônicas, limitam significativamente a mobilidade, aptidão e capacidade funcional do indivíduo, reduzindo, por exemplo, sua capacidade desenvolver atividades de rotina, trabalho ou lazer.

Segundo a Organização Mundial de Saúde[1] (OMS), as condições musculoesqueléticas mais comuns e incapacitantes são osteoartrite, dores nas costas e no pescoço, fraturas associadas à fragilidade óssea e lesões, entre outras diversas condições. Fatores psicossociais e ambientais, além do estilo de vida, podem contribuir para a manifestação de lesões musculoesqueléticas, alguns destes fatores estão relacionados a idade, sexo, capacidade física, obesidade e tabagismo [3].

Dor crônica no sistema musculoesquelético é relatada entre 35% a 50% da população adulta [3]. Condições musculoesqueléticas podem afetar as articulações, como osteoartrite e artrite reumatóide; os ossos, como osteoporose e fraturas traumáticas ou associadas à fragilidade; a espinha, como dor nas costas e pescoço; entre muitos outros casos[1].

Lesões musculoesqueléticos acontecem com frequência e por isso representam um problema significante, pois determinar se um estudo radiográfico apresenta a estrutura óssea "normal" ou "anormal", é uma tarefa crítica na radiologia, dado o fato de que, por exemplo, ao interpretar um estudo como "normal", elimina-se a necessidade dos pacientes se submeterem a procedimentos ou diagnósticos adicionais, o que pode representar um grave risco em caso de erro [30].

O processo de interpretar informações oriundas de exames de imagens é uma tarefa tipicamente complexa. Nesse contexto, muitos sistemas de auxílio ao diagnóstico médico (*Computer Aided Diagnosis*, ou CAD) vêm sendo desenvolvidos para auxiliar nesta tarefa [21, 27, 29, 33].

Para o processo de interpretar de forma automática imagens advindas de exames radiográficos faz-se necessário o uso de diversas técnicas, como por exemplo, Aprendizado Profundo, Processamento de Imagens e Visão Computacional.

Métodos de aprendizado profundo, em especial o uso de Redes Neurais Convolucionais (*Convolutional Neural Network*, ou CNN), têm sido amplamente utilizados e vêm sendo aplicados na construção de modelos que desempenham com sucesso tarefas de classificação de imagens, e representam atualmente o estado da arte para estes problemas [22, 32].

Entretanto, a tarefa de classificar imagens, por vezes, ainda é considerada complexa, dado o fato de que um dos grandes desafios ao se utilizar modelos de aprendizado profundo é que, para esta tarefa, é necessário que uma grande quantidade de dados esteja disponível, o que nem sempre é possível. Além deste fato, o treinamento de modelos de aprendizado profundo requer um certo poder computacional que muitas vezes pode ser limitado.

1.2 Objetivos e Contribuições

Este trabalho visa como principal objetivo, investigar técnicas de classificação de imagens radiográficas e propor uma metodologia eficaz para detectar anormalidades musculoesqueléticas através da

[1] https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions

exploração de métodos de treinamentos de redes neurais.

Para desenvolver a metodologia proposta e atingir os objetivos, dividimos a abordagem em etapas menores, chamadas aqui de objetivos específicos, a fim de contribuir para o cumprimento deste trabalho. Desta forma, os objetivos específicos são:

- Realização de um levantamento bibliográfico bem como um estudo das técnicas e abordagens que representam o estado da arte para a classificação de imagens.
- Realização de pré-processamentos na base de dados.
- Construção de um modelo de classificação utilizando Redes Neurais Convolucionais.
- Realização de experimentos e validação do modelo proposto.
- Avaliação do método proposto comparado à outras abordagens.

Como contribuições, este projeto pretende fornecer uma abordagem útil para a tarefa de classificação de imagens radiográficas musculoesqueléticas baseada em modelos de aprendizado profundo.

1.3 Questões de Pesquisa

Algumas das questões de pesquisa que motivaram e norteiam nosso trabalho são:

- Em termos de eficácia, são melhores múltiplos classificadores binários do que um classificador multi-classe?
- Utilizar pesos de redes pré-treinadas em domínios similares ao de imagens radiográficas é mais eficaz do que se utilizar pesos de redes pré-treinadas com o conjunto ImageNet?
- Utilizar os pesos de redes treinadas para imagens radiográficas de uma dada região do corpo auxilia no processo de treinamento das imagens radiográficas de outras regiões do corpo da base *MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs*?

1.4 Organização do Texto

Este trabalho está organizado em 5 capítulos. No Capítulo 1, descrevemos o problema de pesquisa abordado neste projeto, os objetivos e contribuições esperadas e as questões de pesquisas que motivaram este trabalho. No Capítulo 2, fazemos uma revisão bibliográfica onde são apresentados os conceitos teóricos abordados neste trabalho e os trabalhos relacionados ao tema. No Capítulo 3, indicamos os recursos computacionais que serão empregados, a base de dados que será utilizada, bem como são descritos os procedimentos metodológicos propostos e os experimentos preliminares realizados. No Capítulo 4, apresentamos o plano de trabalho e o cronograma de execução para a realização das atividades. No Capítulo 5, encerramos este documento com algumas considerações finais.

2. Revisão Bibliográfica

Este capítulo apresenta os conceitos teóricos que são importantes para a compreensão deste trabalho, bem como os estudos que norteiam e servem como base para o que está sendo proposto neste projeto.

2.1 Fundamentação Teórica

Nesta seção, são apresentados os fundamentos e conceitos que servem como base para este trabalho e que são necessários ao leitor, para que haja uma compreensão teórica do contexto em que o mesmo está inserido e das técnicas utilizadas.

2.1.1 Redes Neurais Convolucionais

Redes Neurais Convolucionais (*Convolutional Neural Networks*, CNN) [24] são um tipo de modelo de redes neurais profundas que representam atualmente o estado da arte para inúmeros problemas em Visão Computacional como classificação de imagens [22], reconhecimento e detecção de objetos [11, 31] e segmentação de imagens [16]. CNNs empregam filtros convolucionais que permitem extrair características de partes especificas de uma imagem, permitindo também uma representação mais eficaz dos dados [12].

As camadas convolucionais consistem de diversos neurônios responsáveis por aplicarem filtros em partes específicas da imagem de entrada. Cada neurônio está conectado a um conjunto de neurônios da camada anterior, e para cada conexão é atribuído um peso, chamado de peso sináptico. Os pesos da entrada de cada neurônio são combinados entre si, produzindo uma saída que é passada para a camada seguinte [37].

Os pesos atribuídos às conexões entre os neurônios desempenham o papel de um filtro convolucional aplicado no domínio espacial. Dessa forma, na etapa de treinamento da rede, estes filtros são ajustados para que sejam ativados na presença de características relevantes identificadas na entrada.

Enquanto em redes neurais tradicionais cada neurônio de uma camada está conectado a todas as unidades de neurônios da camada seguinte, chamadas de camadas totalmente conectadas (*Fully Connected Layer*), os neurônios das camadas convolucionais utilizam uma conectividade local, como pode ser vista na Figura 2.1 onde, cada neurônio na camada N está conectado a apenas alguns neurônios da camada $N + 1$, ao invés de se conectar a todos os neurônios da camada. Neurônios de uma mesma camada são agrupados e suas saídas formam mapas de características, como representado pela silhueta em azul na Figura 2.1. Um mapa de características é produzido a partir da aplicação de convolução da imagem de entrada com os filtros das camada convolucionais.

Um conceito importante em CNN é o passo (*stride*), pois o tamanho do filtro vai definir o tamanho da vizinhança que cada neurônio da camada irá processar [37]. Assim, esse valor define o tamanho do salto em pixeis entre cada fragmento da imagem. Quando o valor de *stride* é igual a 1, o filtro é movido em um pixel por vez, quando o valor é igual 2, o filtro salta dois pixeis por vez, e assim sucessivamente. Isto poderá produzir volumes de saídas menores.

Um outro conceito importante em Redes Neurais Convolucionais é a camada de agrupamento (*pooling*), que visa reduzir o tamanho da entrada. A operação de *pooling* reduz a dimensionalidade da informação de entrada, porém, mantendo informações importantes. Existem diferentes tipos de *pooling* tais como a soma (*sum pooling*), a média (*average pooling*) e o valor máximo (*max pooling*),

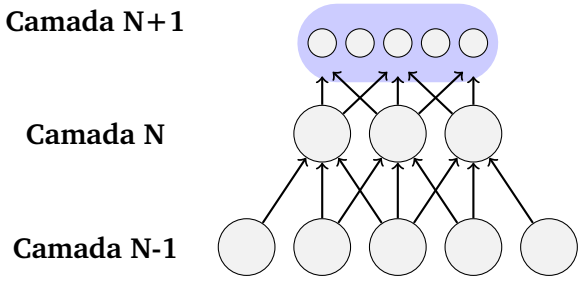


Figura 2.1: Exemplo da conectividade local de uma CNN. A silhueta em azul na camada $N + 1$, destacada na imagem, representa um mapa de características, definido como o agrupamento de neurônios de uma mesma camada.

este último consiste em substituir os valores de uma região pelo valor máximo da mesma. A Figura 2.2 mostra o resultado da aplicação da operação de *max pooling*, onde é possível notar a redução no tamanho da entrada, causando assim uma redução na quantidade de processamento necessário nas próximas camadas da rede.

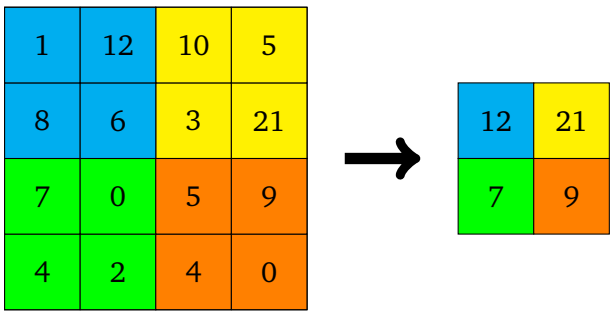


Figura 2.2: Aplicação da operação de *max pooling* em uma imagem 4×4 utilizando uma máscara 2×2.

A última camada de uma CNN é chamada Camada de Saída (*Output Layer*). É uma camada totalmente conectada e, na tarefa de classificação, é responsável por atribuir uma probabilidade para cada classe.

Algumas das arquiteturas de redes neurais convolucionais bastante comuns na literatura, que apresentam bons resultados em competições como *ImageNet* [32] e que também são utilizadas por estudos relacionados ao que estamos propondo, são VGGNet [34], ResNet [15], InceptionV3 [35] e InceptionV4 [36].

A Figura 2.3 apresenta uma comparação entre as arquiteturas de redes convolucionais. O eixo vertical mostra a acurácia obtida por cada modelo no conjunto *ImageNet* [32] e o eixo horizontal mostra o número de operações utilizadas por cada modelo. O tamanho do círculo representa a proporção de parâmetros utilizados na rede.

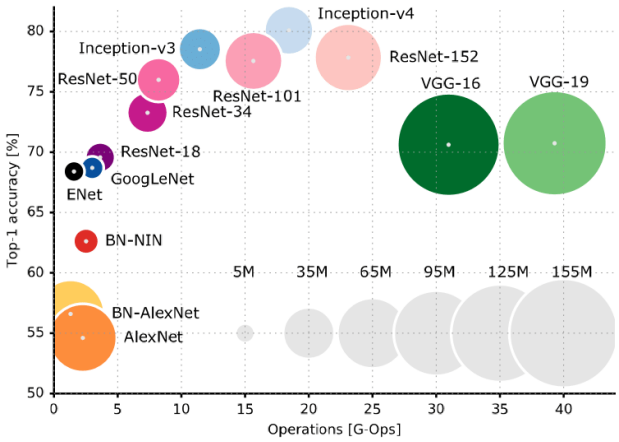


Figura 2.3: Comparação entre arquiteturas de redes neurais convolucionais [20].

2.1.2 Classificação

A tarefa de classificação, no paradigma de aprendizado de máquina supervisionado, é utilizada para predizer a qual classe (ou categoria) uma instância de dados pertence. Um modelo de classificação recebe como entrada um conjunto de dados rotulados, onde cada rótulo representa a classe a qual o dado pertence. Se esses rótulos forem textuais, podem ser atribuídas transformações que os convertem para vetores numéricos. Em uma classificação binária, esses rótulos geralmente são representados pelos valores inteiros 0 e 1. A saída de um modelo de classificação é, de fato, um classificador capaz de predizer as classes das instâncias as quais o modelo foi submetido.

Em aprendizado de máquina, há diversos modelos de classificação que são bastante robustos como *Random Forests* (RF), *Support Vector Machines* (SVM) e *k-Nearest Neighbors* (KNN). Modelos de redes neurais profundas apresentam ótimos resultados para a tarefa de classificação e representam atualmente o estado da arte para esta tarefa [15, 34, 35].

2.1.2.1 Avaliação de Desempenho

Para a avaliação da eficácia de um modelo de classificação, utilizam-se métricas para mensurar a qualidade da classificação e analisar o quanto o mesmo está adequado ao que foi proposto. Estas métricas baseiam-se no conceito de matriz de confusão, que oferece uma medida efetiva do modelo de classificação ao apresentar as classificações corretas e preditas para cada classe. Uma representação da matriz de confusão é mostrada na Tabela 2.1.

		Classe real	
		Positivo	**Negativo**
Resultado predito	**Positivo**	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Negativo	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Tabela 2.1: Matriz de confusão.

Uma métrica bastante utilizada é a acurácia (*accuracy*), que consiste na proporção do que foi predito corretamente, positivo ou negativo, sobre a quantidade total de amostras. É idealmente

utilizada quando as quantidades de amostras que pertencem a cada classe na base de dados são balanceadas. A acurácia visa indicar o quão frequente o classificador está correto e sua fórmula está representada na Equação 2.1. Quando a base é desbalanceada, idealmente utiliza-se a acurácia balanceada [4], que consiste na média da acurácia obtida em cada classe. Assim, é possível evitar valores elevados de acurácia apenas por conta do desbalanceamento da base. A fórmula da acurácia balanceada para uma classificação binária, é representada pela Equação 2.2.

$$ACC = \frac{VP + VN}{Total} \tag{2.1}$$

$$ACC_b = \frac{1}{2}\left(\frac{VP}{VP + FN} + \frac{VN}{FP + VN}\right) \tag{2.2}$$

A área sob a curva ROC (*Area Under the ROC Curve*, AUC) é uma medida da área sob a curva formada pela Taxa de Verdadeiros Positivos (TPR ou *Recall*) e a Taxa de Falsos Positivos (FPR), representadas pelas Equações 2.3 e 2.4, respectivamente.

$$TPR = \frac{VP}{VP + FN} \tag{2.3}$$

$$FPR = \frac{FP}{FP + VN} \tag{2.4}$$

A métrica AUC fornece a medida da qualidade das predições do modelo. Um exemplo de um gráfico contendo a curva ROC e o valor da área pode ser visto na Figura 2.4. Uma curva próxima da linha tracejada (área $= 0.5$) representa um modelo ruim, enquanto valores mais próximos de 1 são considerados bons.

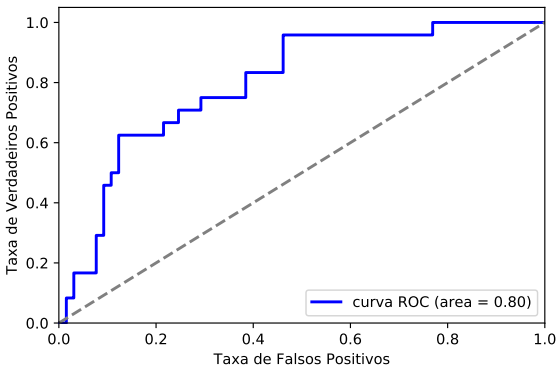


Figura 2.4: Área sob a curva (AUC). Mede-se toda a área do gráfico, do ponto $(0, 0)$ a $(1, 1)$, abaixo da curva ROC.

O coeficiente *Cohen's kappa* [9] é uma medida estatística para avaliar o nível de concordância entre dois conjuntos de dados em um problema de classificação. É uma medida mais robusta do que um simples cálculo de concordância percentual, já que este coeficiente leva em consideração a possibilidade desta concordância ocorrer por acaso.

O coeficiente kappa pode ser representado pela Equação 2.5:

$$kappa = \frac{P_o - P_e}{1 - P_e} \tag{2.5}$$

onde P_o é a concordância observada, cuja fórmula é análoga à da acurácia (Equação 2.1), e P_e é a concordância esperada, que está relacionada ao número de instâncias de cada classe, juntamente com o número de instâncias que o classificador predisse corretamente. Sua fórmula pode ser representada pela Equação 2.6.

$$P_e = \frac{(FP + VN) \times (FN + VN) + (VP + FN) \times (VP + FP)}{Total^2} \tag{2.6}$$

O valor do coeficiente kappa é sempre menor ou igual a 1. Um valor igual a 1 implica concordância perfeita e valores menores que 1 implicam menos concordâncias entre as anotações. Possíveis interpretações do coeficiente kappa, segundo Landis e Koch [23] estão apresentadas na Tabela 2.2.

Valor do coeficiente kappa	**Nível de concordância**
<0,00	Não existe concordância
0,00 - 0,20	Concordância mínima
0,21 - 0,40	Concordância razoável
0,41 - 0,60	Concordância moderada
0,61 - 0,80	Concordância substancial
0,81 - 1,00	Concordância perfeita

Tabela 2.2: Classificação dos diferentes níveis de concordância do coeficiente kappa.

A função de Perda Logarítmica (*log loss*) fornece uma medida de qualidade da classificação, onde a saída é um valor de probabilidade (entre 0 e 1). Este valor aumenta a medida que a classe predita diverge da classe real, sendo assim uma medida de confiança para as predições do modelo. Em uma classificação binária, a perda logarítmica pode ser calculada pela Equação 2.7. Em uma classificação multi-classe, a função de perda é calculada para cada classe observada e então os resultados são somados, como mostra a Equação 2.8:

$$\textit{loss\_binario} = -(y \log(p) + (1 - y) \log(1 - p)) \tag{2.7}$$

$$\textit{loss\_multi-classe} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{2.8}$$

onde M é o número de classes, y_c é um indicador binário se a classe c é a classificação correta para a observação o, e p é a probabilidade predita da observação o ser da classe c.

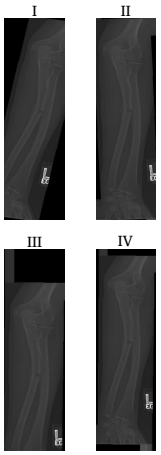
2.1.3 Aumentação de Dados

Aumentação de Dados (*Data Augmentation*) é uma técnica, ou um conjunto de técnicas, para gerar novas amostras a partir de outras presentes no conjunto de dados, a fim de elevar a generalidade do modelo. Esta técnica visa melhorar a eficácia de um modelo na etapa de treinamento dos dados, pois uma das grandes preocupações durante o processo de aprendizagem é quanto ao *overfitting*.

O *overfitting* é um problema sobretudo quando o conjunto de dados utilizado é pequeno ou não representativo o suficiente. Consiste da rede memorizar os dados de treinamento, aprendendo padrões específicos demais das amostras do conjunto e, quando submetido a um novo conjunto, não conseguir generalizar de forma satisfatória e, assim, não realizar uma boa classificação.

Há várias maneiras possíveis de realizar a aumentação de dados. Métodos comuns aplicam combinações de operações da área de processamento de imagens como translação, rotação, modificação de perspectiva, cisalhamento, entre outras técnicas. A Figura 2.5 ilustra o resultado do processo de aumentação de dados em que foram geradas 6 novas imagens a partir de uma amostra original (Figura 2.5a), onde foram aplicadas operações como rotações, translações e cisalhamentos em determinados ângulos e variações. O objetivo é que na etapa de treinamento de um modelo o mesmo não receba imagens exatamente iguais, ajudando a expor o modelo a aspectos mais relevantes dos dados, para que assim possa generalizar melhor.

(a) Imagem original.



(b) Imagens geradas a partir da aumentação de dados.

Figura 2.5: Exemplo de aumentação de dados em que 4 novas imagens (b) foram geradas a partir de uma imagem original (a). Os métodos aplicados foram rotação (I), cisalhamento (II), translação (III) e ampliação (IV).

2.1.4 Métodos de Treinamento

Nesta seção, iremos descrever alguns métodos e técnicas de treinamento de redes neurais que são utilizados atualmente e que vêm obtendo bons resultados. Os métodos que iremos descrever são: Transferência de Aprendizado, Ajuste Fino, Aprendizagem em Multitarefas e *Warm Starting*.

2.1.4.1 Transferência de Aprendizado

A técnica de Transferência de Aprendizado (*Transfer Learning*) é utilizada com o objetivo de reusar parte do "aprendizado" já adquirido por um modelo, aproveitando-o com o intuito de resolver novas tarefas, reduzindo assim o tempo necessário para treinar um novo modelo de rede neural profunda [7]. Esta técnica visa reduzir o número de parâmetros a serem aprendidos, através da reutilização de informações aprendidas por um modelo anterior, onde essas informações são consideradas adequadas para uma nova tarefa.

Como modelos de aprendizagem profunda necessitam de uma grande quantidade de dados para serem treinados e nem sempre essa quantidade de dados se encontra disponível ou acessível, torna-se difícil a realização desta tarefa. Dessa forma, é comum treinar uma rede convolucional sob uma grande quantidade de dados como, por exemplo, o conjunto ImageNet [32] que contém $1{,}2$ milhões de imagens divididas em 1000 classes, e utilizar essa rede como uma inicialização dos pesos ou extrator de características para a nova tarefa.

Os principais cenários relacionados à transferência de aprendizado em redes convolucionais estão ligados ao seu uso para extração de características e o uso da estratégia de *Fine Tuning*. Este último será melhor descrito na Seção 2.1.4.2.

2.1.4.2 Ajuste Fino

O conceito de Ajuste Fino (*Fine Tuning*), assim como a técnica de transferência de aprendizado, vem sendo amplamente utilizados, e atuando quase sempre de forma conjunta. A técnica de *fine tuning* consiste em ajustar parâmetros de uma rede treinada para se adaptar a uma nova tarefa. Ou seja, utiliza um modelo já treinado com alguns dados e realiza um re-treinamento do modelo, ou de parte do modelo, em um novo conjunto de dados. Dessa forma, esta técnica consiste em desafixar ou liberar os pesos de algumas camadas de uma rede neural, para que sejam ajustados sob novos dados.

A Figura 2.6 ilustra uma rede neural, onde cada nó representa um neurônio da rede e cada aresta entre os neurônios representa um peso. A parte destacada por neurônios vermelhos e pesos em azul representa uma parte fixada da rede. A parte em branco representa os pesos da rede que estão liberados e na etapa de re-treinamento serão ajustados.

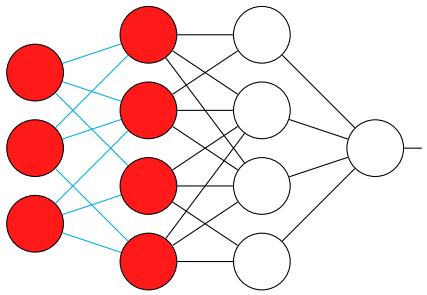


Figura 2.6: Ilustração de uma rede neural com *fine tuning*. A parte colorida representa camadas congeladas da rede cujos pesos não serão alterados. A parte não colorida ilustra camadas descongeladas cujos pesos serão ajustados durante o treinamento da rede.

A técnica de *fine tuning* ajusta as representações mais abstratas do modelo que está sendo reutilizado, a fim de torná-las mais relevantes para um novo problema em questão [8]. O *fine tuning* é geralmente utilizado aproveitando as camadas iniciais da rede, isto é, mantendo-as congeladas, e treinando somente as camadas mais avançadas, dado que as camadas iniciais de uma rede neural aprendem características mais gerais, e à medida que se avança na rede é que características mais complexas e padrões mais específicos para a tarefa vão sendo aprendidos.

2.1.4.3 Aprendizagem em Multitarefas

Aprendizagem em Multitarefas (*Multi-Task Learning*, MTL) [6] é o método de aprender simultaneamente várias tarefas que são relacionadas entre si, como classificação, regressão, segmentação,

localização e muitas outras [25, 38]. O método se caracteriza pelo compartilhamento de informações e parâmetros que são aprendidos em conjunto entre as diferentes tarefas [28], fornecendo uma maneira eficaz de aproveitar as informações específicas de cada domínio de tarefa para melhorar a eficácia do modelo.

Como as redes de MTL utilizam camadas compartilhadas que são treinadas em paralelo para todas as diferentes tarefas, as informações aprendidas por uma tarefa podem ajudar a melhorar no processo de aprendizagem das outras tarefas [6]. A Figura 2.7 ilustra um modelo MTL. A etapa de *backpropagation* é feita em paralelo para cada saída do modelo, atualizando os pesos das camadas compartilhadas, tornando assim as informações aprendidas universais para todas as tarefas [26].

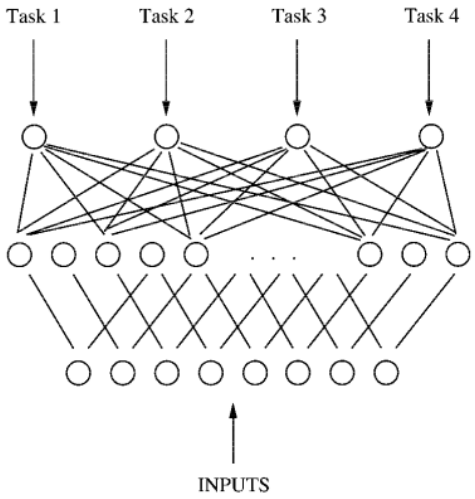


Figura 2.7: Modelo MTL de 4 tarefas para uma mesma entrada [6].

2.1.4.4 Warm Starting

Quando treinamos diversos modelos para uma mesma base de dados ou para bases de dados similares, o método de *Warm Starting* permite reutilizar características aprendidas pelos modelos anteriores, como uma inicialização próxima à ideal para um modelo seguinte de uma tarefa similar. Esta técnica é utilizada para resolver o problema de *Cold Starting*, que ocorre quando os parâmetros passados a um modelo não são adequados e o mesmo demora para convergir ou não é capaz de encontrar uma solução ótima.

É equivalente à técnica de Transferência de Aprendizado, porém, enquanto esta apresenta uma abordagem mais geral, isto é, reutiliza características aprendidas por modelos em domínios distintos, a técnica de *Warm Starting* apresenta uma abordagem mais específica, reutilizando características aprendidas por modelos em domínios similares ao que está sendo proposto, com o objetivo de fornecer uma melhor inicialização para o treinamento do modelo seguinte.

2.2 Trabalhos Correlatos

Nesta seção, são descritos os estudos e métodos de aprendizagem profunda aplicados no domínio de imagens médicas e que estão relacionados a este trabalho.

A tarefa de classificação automática de imagens médicas vem sendo amplamente explorada para diversos tipos de imagens de diferentes áreas da medicina e inúmeras abordagens de aprendizado de máquina surgem na literatura para a resolução destes problemas. Estas abordagens consistem normalmente de algumas etapas como pré-processamento, extração de características e, por fim, a etapa de classificação [18].

Para a extração de característica métodos como *wavelets* [1], Gabor [5] e redes convolucionais [19, 27] são bastante utilizados. A etapa de classificação inclui métodos como *Random Forests* [5], *Support Vector Machines* [14, 27] e Regressão Logística [19].

Estudos tais como os de Kawahara et al. [19] e Menegola et al. [27], investigaram o problema de classificação de imagens de lesões de pele. Kawahara et al. [19] propuseram o treinamento de um classificador linear baseado em características extraídas de imagens através de redes neurais convolucionais pré-treinadas. Os autores utilizaram a arquitetura AlexNet [22] pré-treinada com o conjunto ImageNet [32], convertendo as camadas totalmente conectadas (*fully connected layers*) em camadas convolucionais para que estas atuassem como filtros convolucionais, extraindo características da imagem. Os autores utilizaram a base *Dermofit Image Library*[1], composta por 1300 imagens de lesões de pele, com 10 categorias de lesões, e realizaram experimentos com classificações para 10, 5 e 2 classes de lesões, onde obtiveram valores de acurácia iguais a $81{,}8\%$, $85{,}8\%$ e $94{,}8\%$ respectivamente.

De maneira semelhante, Menegola et al. [27] propuseram um conjunto de experimentos com diferentes configurações de um modelo e analisaram os efeitos que, tanto o uso da transferência de aprendizado quanto o uso do *fine tuning*, resultavam. Para o treinamento e teste dos modelos propostos, os autores utilizaram as bases *Interactive Atlas of Dermoscopy* (Atlas) [2] e *ISBI Challenge 2016 / ISIC Skin Lesion Analysis Towards Melanoma Detection — Part 3: Disease Classification* (ISIC) [13], compostas por imagens de lesões de pele com diagnósticos que incluem lesões benignas, melanomas, carcinoma entre outros tipos de lesões. Os autores realizaram experimentos para os problemas de classificação de lesões entre Maligno $\times$ Benigno, Melanoma $\times$ Benigno e Melanoma $\times$ Carcinoma $\times$ Benigno. Através do uso de transferência de aprendizado a partir do conjunto ImageNet [32] e da técnica de ajuste fino, os autores obtiveram valores de AUC iguais $82{,}5\%$, $80{,}9\%$ e $83{,}6$ para os experimentos citados anteriormente.

O uso de redes neurais convolucionais pré-treinadas, caracterizado como uma transferência de aprendizado, para o problema proposto relacionado a imagens de pele foi capaz de obter bons resultados [19] e de forma análoga, o uso de *fine tuning* também propiciou uma melhora nos resultados da classificação [27].

Embora utilizem dados de imagens diferentes do que este trabalho propõe, os estudos citados acima abordam o mesmo conceito de sistemas de auxílio ao diagnóstico médico (CAD) assim como os estudos de Gale et al. [10] e Rajpurkar et al. [30] que utilizam imagens radiográficas de fraturas ósseas, mostrando o quão generalizável estes modelos podem ser.

Gale et al. [10] investigaram o problema de detecção de fraturas no quadril através de radiografias pélvicas. Em sua base de dados os autores utilizaram arquivos clínicos de radiologia do *Royal Adelaide Hospital*. Primeiramente, os autores propuseram uma arquitetura de rede convolucional que chamaram de *CNN-Frontal* e a treinaram para identificar radiografias pélvicas frontais, pois a base também incluía imagens laterais do quadril, tórax e coluna vertebral. Em seguida, propuseram outra arquitetura de rede convolucional que chamaram de *CNN-bounding* para realizar uma regres-

[1] https://licensing.eri.ed.ac.uk/i/software/dermofit-image-library.html

são, que foi treinada para localizar o colo do fêmur, que seria o local mais relevante onde se ocorre a fratura. E, por fim, uma terceira arquitetura foi proposta e chamada de *CNN-metal*, cujo objetivo era o de excluir casos com metais implantados nas fraturas e outras operações semelhantes. Para a classificação, Gale et al. [10] aplicaram então um modelo de rede convolucional conhecido como DenseNet [17] com 172 camadas de profundidade e utilizaram um estratégia de pesquisa em grade (*grid search*) para determinar os hiperparâmetros do modelo. Os autores reportaram o valor de acurácia obtido igual a 97%.

Rajpurkar et al. [30] propuseram uma base de dados de radiografias musculoesqueléticas com imagens demarcadas como normal ou anormal e validadas por radiologistas certificados pelo conselho do *Stanford Hospital*. A base contém radiografias da extremidades superiores do corpo sendo elas ombro, úmero, cotovelo, antebraço, punho, mão e dedo. Rajpurkar et al. [30] propuseram também um modelo para predizer a probabilidade de anormalidade em um estudo radiográfico pertencente a base proposta. Para isso, como Gale et al. [10], eles utilizaram um modelo baseado na arquitetura DenseNet mas com 169 camadas convolucionais e com seus pesos inicializados com os pesos de um modelo pré-treinado com o conjunto ImageNet, e modificaram a camada final totalmente conectada (*fully connected*) para possuir saída única. Os autores avaliaram a performance do modelo para cada região do corpo pertence a base de dados e obtiveram uma média do coeficiente kappa igual a $70{,}5\%$. Os autores também fornecem a performance de 3 radiologistas onde, os radiologistas 1, 2, 3 obtiveram uma média do coeficiente kappa igual a $73{,}1$, $76{,}3$ e $77{,}8$ respectivamente.

A Tabela 2.3 fornece um comparativo de técnicas e tarefas entre os estudos citados anteriormente e este trabalho.

Modelo	**Tarefa**	**Tipo**	**Aumentação de Dados**	**Transf. Aprendizado**	**Ajuste Fino**
Kawahara et al. [19]	Classificação	Lesões de pele	Sim	Sim	Não
Menegola et al. [27]	Classificação	Lesões de pele	Sim	Sim	Sim
Gale et al. [10]	Classificação	Radiografia	Sim	Não	Não
Rajpurkar et al. [30]	Classificação	Radiografia	Sim	Sim	Não
Proposto	Classificação	Radiografia	Sim	Sim	Sim

Tabela 2.3: Comparação entre as técnicas utilizadas pelos trabalhos relacionados e o proposto.

3. Metodologia

Neste capítulo, são apresentados os passos de implementação a serem realizados neste trabalho, assim como informações sobre a base de dados e os pré-processamentos necessários, os recursos computacionais e os experimentos que serão conduzidos.

3.1 Recursos Computacionais

Os métodos propostos serão implementados na linguagem Python, que provê uma gama de bibliotecas e recursos para Visão Computacional e Aprendizado de Máquina, como `NumPy`[1], `Pandas`[2], `OpenCV`[3], `Matplotlib`[4], `scikit-learn`[5], `scikit-image`[6], `Keras`[7] e `TensorFlow`[8].

Os experimentos deste projetos serão conduzidos no Laboratório de Informática Visual (LIV) do Instituto de Computação da Unicamp. As máquinas que serão utilizadas para realização dos experimentos seguem as especificações descritas na Tabela 3.1.

Processador	Intel i7-3770 3.5GHz
Memória RAM	32GB
Placa de vídeo	NVidia GeForce GTX 1080
Memória	11GB
Sistema Operacional	Ubuntu 16.04 LTS

Tabela 3.1: Especificações das máquinas utilizadas nos experimentos.

3.2 Bases, Análises e Pré-Processamento dos Dados

A base de dados utilizada neste trabalho é a base *MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs* [30] que consiste de um conjunto de imagens de raios-X ósseos.

A base de dados *MURA* possui 40561 imagens radiográficas musculoesquelética de 14863 casos de estudo a partir de 12173 pacientes. As imagens estão divididas entre classes que representam estudos radiográficos de diferentes regiões do corpo que são: ombro, úmero, cotovelo, antebraço, pulso, mão e dedo. Cada um dos casos de estudo foi demarcado manualmente por radiologistas como "normal" ou "anormal".

O conjunto de dados é composto por três subconjuntos: Treinamento, Validação e Teste. Originalmente, os conjuntos de treinamento, validação e teste possuíam, respectivamente, um total de 36808, 3197 e 556 imagens. Como não possuímos acesso direto à base de teste, realizamos um rearranjo dos conjuntos, onde o conjunto de validação passou a ser nosso conjunto de teste e para a criação do novo conjunto de validação separamos 20% do conjunto de treinamento original.

A Tabela 3.2 mostra o arranjo das imagens em cada região e classe nos conjuntos de treinamento, validação e teste. É possível notar um desbalanceamento na quantidade de imagens em

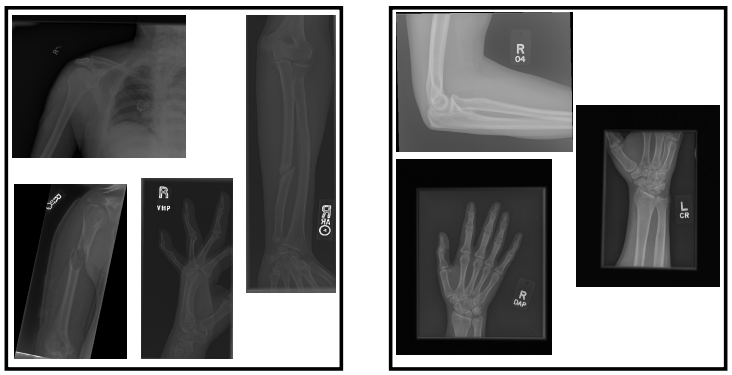
[1] https://numpy.org/
[2] https://pandas.pydata.org/
[3] https://opencv.org/
[4] https://matplotlib.org/
[5] https://scikit-learn.org/
[6] https://scikit-image.org/
[7] https://keras.io/
[8] https://www.tensorflow.org/

determinadas classes para cada região do corpo. Apenas a região 'Ombro' apresenta uma distribuição mais igualitária entre as classes, tanto no conjunto de treinamento quanto nos de validação e teste.

Classes	**Treinamento**		**Validação**		**Teste**		Total
	Normal	Anormal	Normal	Anormal	Normal	Anormal	
Pulso	4612	3189	1153	798	364	295	**10411**
Ombro	3368	3334	843	834	285	278	**8942**
Mão	3247	1187	812	297	271	189	**6003**
Dedo	2510	1574	628	394	214	247	**5567**
Cotovelo	2340	1604	585	402	235	230	**5396**
Antebraço	931	528	233	133	150	151	**2126**
Úmero	538	479	135	120	148	140	**1560**
Total	17546	11895	4389	2978	1667	1530	**40005**

Tabela 3.2: Arranjo das imagens em cada região do corpo e classe nos conjuntos de treinamento, validação e teste.

As imagens contidas na base de dados são monocromáticas e estão no formato *Portable Network Graphics* (PNG). Algumas imagens apresentam uma certa rotação. As resoluções das imagens variam e estão entre 132×512 e 512×512 pixels. A Figura 3.1 ilustra algumas das imagens presentes na base de dados.



(a) $Anormais$ (b) $Normais$

Figura 3.1: Exemplos de imagens contidas na base de dados e suas respectivas anotações. (a): Ombro, Antebraço, Úmero e Dedo. (b): Cotovelo, Pulso e Mão.

Como pré-processamento, um dos procedimentos que iremos realizar será aplicar sobre o conjunto de treinamento técnicas de aumentação de dados como as citadas na Seção 2.1.3, a fim de causar um maior balanceamento entre as classes das bases de dados e, consequentemente, tornar o modelo mais robusto. Aplicaremos também procedimentos utilizando redes convolucionais, como em Gale et al. [10], com o objetivo de remover dados e valores espúrios como marcas ou textos presentes nas imagens.

3.3 Construção de uma Metodologia de Classificação

Para a construção da metodologia de classificação, definiremos um protocolo utilizando os métodos e técnicas citados na Seção 2.1.4.

Na etapa de treinamento do modelo proposto, após ser cumprida toda a etapa de pré-processamento, dividiremos o processo de classificação em duas partes. Na primeira parte, a partir das imagens contidas na base de dados com suas respectivas anotações (*labels*) em relação à região do corpo, visamos treinar um classificador apenas para distinguir a qual região do corpo uma amostra de imagem pertence. Na segunda fase, consideraremos as imagens de cada uma das 7 regiões do corpo como uma base de dados distinta, com cada imagem classificada como normal ou anormal. Na primeira etapa desta pesquisa, iremos propor um método baseado em MTL, onde cada tarefa será uma classificação binária para cada uma das 7 bases. Dessa forma, receberemos como entrada estas 7 bases de dados para realizar o treinamento deste modelo multitarefas. Como as imagens das bases de dados são bastante semelhantes, é possível obter características relevantes que as mesmas compartilham. A Figura 3.2 ilustra os passos a serem utilizados nas etapas de pré-processamento, treinamento e teste da metodologia proposta.

Em uma segunda etapa, utilizaremos as informações aprendidas na etapa anterior, isto é, os pesos obtidos pelas camadas compartilhadas do modelo MTL, como uma inicialização para o treinamento de uma rede neural em uma base de região do corpo, por exemplo, a base de "Dedo", utilizando a técnica de *Warm Starting*. A partir desse ponto, iremos realizar um treinamento "incremental" utilizando os pesos obtidos no treinamento das bases de dados anteriores, ou seja, os pesos obtidos após o treinamento para a base de dados de "Dedo" serão utilizados como inicialização para o treinamento na base de "Mão" e assim sucessivamente para todas as 7 bases de dados. Ao final dessa fase, obteremos um classificador binário para cada região do corpo.

Por fim, na fase de teste, dada uma amostra de imagem iremos submetê-la ao classificador oriundo da parte 1 para determinar à qual região do corpo a mesma pertence, bem como determinar qual classificador proveniente da parte 2 devemos utilizar para então realizar a classificação final (normal ou anormal).

Acreditamos que tanto MTL quanto *Warm Starting* sejam técnicas complementares e, que através da combinação das mesmas, poderemos melhorar o aprendizado dos modelos de redes neurais e aumentar a eficácia da classificação das imagens radiográficas.

3.4 Experimentos e Resultados Preliminares

Com o intuito de analisar a qualidade de modelos clássicos de redes neurais convolucionais, como os presentes na Figura 2.3, realizamos alguns experimentos iniciais ainda com as imagens originais, isto é, sem a etapa de pré-processamento citada anteriormente para que depois possamos comparar os efeitos causados pela mesma.

Verificamos inicialmente a capacidade de uma rede conseguir identificar bem cada região do corpo e classificá-la corretamente. Para isto, utilizamos a arquitetura VGG-16 [34] pré-treinada com o conjunto ImageNet [32], onde removemos sua última camada e adicionamos nossa própria camada de classificação (7 classes). Desafixamos também os pesos de toda a rede a fim de realizar a técnica de ajuste fino.

Treinamos esta rede durante 30 épocas, utilizando o otimizador RMSprop com taxa de apren-

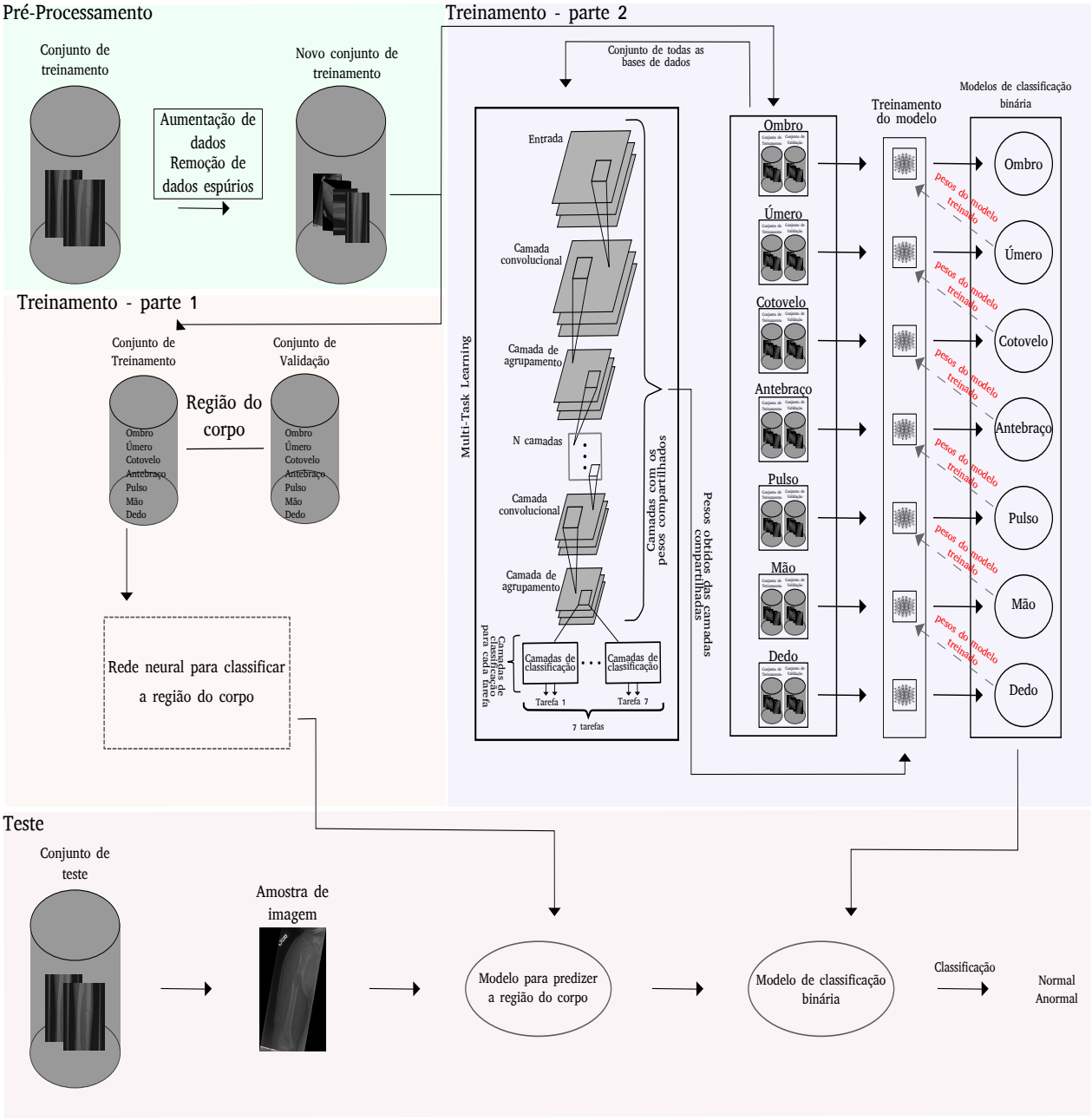


Figura 3.2: Metodologia proposta para classificação de imagens radiográficas.

dizado igual a 2×10^{-5} e Perda Logarítmica como função de perda. A Tabela 3.3 apresenta os resultados que foram obtidos. Os resultados indicam que a rede utilizada é capaz de distinguir bem cada região do corpo.

Métricas (%)		
AUC	ACC_b	kappa
0,98	0,95	0,96

Tabela 3.3: Classificação das 7 regiões do corpo presentes na base de dados.

Como a tarefa de distinguir cada região do corpo obteve bons resultados, seguimos realizando alguns experimentos, dessa vez, com a tarefa de classificação binária, com o objetivo de determinar para cada imagem de cada região do corpo se a mesma está normal ou anormal.

Consideramos cada região do corpo (ombro, úmero, cotovelo, antebraço, pulso, mão e dedo) como uma base de dados distinta, onde cada uma possui apenas as imagens com seus respectivos rótulos (normal ou anormal).

Utilizamos as arquiteturas de redes neurais VGG-16 [34], InceptionV3 [35] e DenseNet [17] para realizar a tarefa de classificação binária. Para todas estas redes, removemos sua última camada e adicionamos nossa própria camada de classificação (2 classes), também desafixamos os pesos de toda a rede a fim de realizar a técnica de ajuste fino.

Treinamos estas redes durante 60 épocas, utilizando o otimizador SGD com taxa de aprendizagem igual a 1×10^{-3} e Perda Logarítmica como função de perda. A Tabela 3.4 apresenta os resultados obtidos neste experimento para cada arquitetura de rede neural.

Região	**Arquitetura**								
	VGG-16			InceptionV3			DenseNet		
	AUC	ACC_b	Kappa	AUC	ACC_b	Kappa	AUC	ACC_b	Kappa
Ombro	0,70	0,70	0,41	0,61	0,61	0,22	0,51	0,51	0,03
Cotovelo	0,73	0,73	0,49	0,53	0,57	0,14	0,50	0,42	0,00
Antebraço	0,64	0,64	0,30	0,56	0,56	0,13	0,70	0,76	0,43
Úmero	0,72	0,72	0,44	0,66	0,66	0,33	0,73	0,74	0,47
Mão	0,64	0,64	0,34	0,52	0,52	0,05	0,52	0,70	0,07
Dedo	0,71	0,71	0,43	0,54	0,54	0,07	0,58	0,64	0,18
Pulso	0,79	0,79	0,59	0,58	0,58	0,16	0,55	0,54	0,10
Média	**0,70**	**0,70**	**0,43**	0,57	0,58	0,16	0,58	0,62	0,18

Tabela 3.4: Resultados da classificação binária para cada região do corpo.

Com base neste experimento, notamos que os melhores resultados foram obtidos utilizando a rede VGG-16, porém ainda bem abaixo dos resultados que representam o estado da arte para este problema [30].

Realizamos mais um experimento para verificar se, ao treinar um modelo de rede neural para a tarefa de classificação multi-classe, iríamos obter resultados melhores em relação à tarefa de classificação binária do experimento anterior.

Para isto, unimos todas as bases de regiões do corpo e, para cada subclasse normal e anormal pertencentes a elas, transformamos em uma classe da forma: Ombro_normal, Ombro_anormal, Cotovelo_normal, Cotovelo_anormal, e assim por diante. Dessa forma, passamos a ter um conjunto formado por 14 classes.

Selecionamos então a rede VGG-16, pois obteve os melhores resultados no experimento anterior e utilizamos as mesmas configurações de hiperparâmetros para realizar o treinamento deste conjunto com 14 classes. Após a etapa de treinamento, avaliamos o modelo para cada região do corpo individualmente, onde, por exemplo, se uma imagem for Ombro_normal e for predita como Ombro_normal atribuímos como uma classificação correta e, para qualquer outra predição diferente, atribuímos como uma classificação errada.

A Tabela 3.5 apresenta os resultados obtidos neste experimento com a rede VGG-16. É possível observar uma melhora na média dos valores de AUC, acurácia balanceada (ACC_b) e kappa para

cada conjunto correspondente a uma região do corpo.

Região	Arquitetura		
	VGG-16		
	AUC	ACC_b	Kappa
Ombro	0,84	0,84	0,68
Cotovelo	0,89	0,88	0,79
Antebraço	0,91	0,79	0,76
Úmero	0,90	0,92	0,84
Mão	0,74	0,74	0,57
Dedo	0,91	0,91	0,81
Pulso	0,88	0,88	0,79
Média	0,87	0,85	0,75

Tabela 3.5: Resultados obtidos para as imagens de cada região do corpo através do modelo treinado com 14 classes.

Notamos que os resultados alcançados com os experimentos realizados até o momento, sem nenhuma etapa de pré-processamento ou otimização, ficaram equiparáveis com os reportados por Rajpurkar et al. [30], citados na Seção 2.2. Estes experimentos são úteis, pois servem como um *baseline* para este trabalho.

4. Plano de Trabalho

O plano de trabalho é formado pelas etapas descritas abaixo:

1. Obtenção dos créditos obrigatórios em disciplinas;
2. Pesquisa bibliográfica e estudo das principais técnicas e abordagens a serem utilizadas;
3. Exame de Qualificação do Mestrado (EQM);
4. Análise e preparação dos dados;
5. Experimentos iniciais com redes neurais genéricas;
6. Definição de uma metodologia de classificação e realização de experimentos com a mesma;
7. Participação do Programa de Estágio Docente (PED);
8. Análise dos resultados obtidos;
9. Escrita da dissertação e publicação dos resultados;
10. Defesa da dissertação de mestrado.

A Tabela 4.1 apresenta o cronograma de execução das atividades propostas, dividido em trimestres.

Atividades	**2019**				**2020**				**2021**
	1°	**2°**	**3°**	**4°**	**1°**	**2°**	**3°**	**4°**	**1°**
Obtenção dos créditos obrigatórios em disciplinas.	●	●	●	●					
Revisão da literatura	●	●	●	●	●				
Exame de Qualificação do Mestrado (EQM)				●					
Análise e preparação dos dados		●	●	●					
Experimentos iniciais com modelos de redes neurais		●	●	●					
Construção de uma metodologia de classificação				●	●	●			
Experimentos com a metodologia definida				●	●	●			
Participação do Programa de Estágio Docente (PED)					●	●			
Análise e validação dos resultados obtidos					●	●	●		
Publicação dos resultados						●	●	●	●
Escrita da dissertação				●	●	●	●	●	●
Apresentação da dissertação de mestrado									●

Tabela 4.1: Cronograma de execução das atividades, dividido em trimestres.

5. Considerações Finais

Detectar anormalidades em imagens de radiografias musculoesqueléticas é uma tarefa rotineira de profissionais radiologistas, que podem ser submetidos a inúmeros casos durante seus plantões.

Sistemas que possam auxiliar radiologistas nesta tarefa representam uma aplicação importante na área. Este projeto pretende investigar o problema de detectar anormalidades em imagens radiográficas de diferentes regiões do corpo e desenvolver um método eficaz para a tarefa.

Resultados obtidos através da execução de alguns experimentos preliminares mostraram que as arquiteturas experimentadas não conseguiram obter bons resultados, indicando que a tarefa de classificar um caso de estudo como “normal” ou “anormal” não é trivial.

Esperamos que, com a implementação completa da nossa metodologia, possamos obter bons resultados, obtendo contribuições significativas tanto no contexto cientifico quanto no social.

Bibliografia

[1] M. Al-Ayyoub, I. Hmeidi, and H. Rababah. Detecting Hand Bone Fractures in X-Ray Images. *Journal of Multimedia Processing and Technologies*, 4:155–168, 2013.

[2] G. Argenziano, H. Soyer, V. De Giorgi, D. Piccolo, P. Carli, and M. Delfino. Interactive atlas of dermoscopy (Book and CD-ROM). *EDRA Medical Publishing & New Media*, 2000.

[3] S. Bergman. Public health perspective–how to improve the musculoskeletal health of the population. *Best Practice & Research Clinical Rheumatology*, 21(1):191–204, 2007.

[4] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *20th International Conference on Pattern Recognition (ICPR)*, pages 3121–3124, 2010.

[5] Y. Cao, H. Wang, M. Moradi, P. Prasanna, and T. F. Syeda-Mahmood. Fracture detection in X-Ray images through stacked random forests feature fusion. In *12th International Symposium on Biomedical Imaging (ISBI)*, pages 801–805. IEEE, 2015.

[6] R. Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, 1997.

[7] M. C. Carvalho. Esquemas de transferência para aprendizado profundo em classificação de imagens. *Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação*, 2015.

[8] F. Chollet. *Deep Learning with Python*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2017.

[9] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[10] W. Gale, L. Oakden-Rayner, G. Carneiro, A. P. Bradley, and L. J. Palmer. Detecting hip fractures with radiologist-level performance using deep neural networks. *arXiv:1711.06504*, 2017.

[11] R. Girshick. Fast R-CNN. In *15th International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[13] D. Gutman, N. C. Codella, E. Celebi, B. Helba, M. Marchetti, N. Mishra, and A. Halpern. Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC). *arXiv:1605.01397*, 2016.

[14] J. C. He, W. K. Leow, and T. S. Howe. Hierarchical classifiers for detection of fractures in X-Ray images. In *12th International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 962–969, 2007.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *16th International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *30th Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

[18] A. Jiménez-Sánchez, A. Kazi, S. Albarqouni, S. Kirchhoff, A. Sträter, P. Biberthaler, D. Mateus, and N. Navab. Weakly-supervised localization and classification of proximal femur fractures. *arXiv:1809.10692*, 2018.

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

[20] C. Kawatsu, F. Koss, A. Gillies, A. Zhao, J. Crossman, B. Purman, D. Stone, and D. Dahn. Gesture recognition for robotic control using deep learning. In *9th NDIA Ground Vehicle Systems Engineering and Technology Symposium (GVSETS)*, 2017.

[21] T. Kooi, G. Litjens, B. Van Ginneken, A. Gubern-Mérida, C. I. Sánchez, R. Mann, A. den Heeten, and N. Karssemeijer. Large scale deep learning for computer aided detection of mammographic lesions. *Medical Image Analysis*, 35:303–312, 2017.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *25th Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[23] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

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

[25] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. NAACL, 2015.

[26] X. Liu, P. He, W. Chen, and J. Gao. Multi-Task Deep Neural Networks for Natural Language Understanding. *arXiv:1901.11504*, 2019.

[27] A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle. Knowledge transfer for melanoma screening with deep learning. In *14th International Symposium on Biomedical Imaging (ISBI)*, pages 297–300, 2017.

[28] T. L. Nwe, T. H. Dat, and B. Ma. Convolutional Neural Network with Multi-Task Learning Scheme for Acoustic Scene Classification. In *9th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1347–1350, 2017.

[29] A. Peixinho, S. Martins, J. Vargas, A. Falcão, J. Gomes, and C. Suzuki. Diagnosis of human intestinal parasites by deep learning. In *5th ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing (VipIMAGE)*, pages 107–112, 2015.

[30] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball, C. Langlotz, K. Shpanskaya, M. Lungren, and A. Ng. Mura: Large dataset for abnormality detection in musculoskeletal radiographs. *arXiv:1712.06957*, 2017.

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *28th Advances in Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

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

[33] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5):1285–1298, 2016.

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

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *29th Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *AAAI Conference on Artificial Intelligence*, 2017.

[37] A. C. G. Vargas, A. Paes, and C. N. Vasconcelos. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In *29th Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 1–4, 2016.

[38] Y. Yang and T. Hospedales. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. *arXiv:1605.06391*, 2016.