



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Análise Comparativa de Redes Neurais Convolucionais para a Detecção de Câncer de Pulmão em Tomografias Computadorizadas

Alexandre Santana Sousa

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Orientadora  
Prof.a Dr.a Roberta Barbosa Oliveira

Brasília  
2023



Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# **Análise Comparativa de Redes Neurais Convolucionais para a Detecção de Câncer de Pulmão em Tomografias Computadorizadas**

Alexandre Santana Sousa

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Prof.a Dr.a Roberta Barbosa Oliveira (Orientadora)  
CIC/UnB

Prof. Dr. Camilo Chang Dórea    Prof. Dr. Vinicius Ruela Pereira Borges  
Universidade de Brasília                      Universidade de Brasília

Prof. Dr. João Luiz Azevedo de Carvalho  
Coordenador do Curso de Engenharia da Computação

Brasília, 7 de fevereiro de 2023

# Dedicatória

Dedico este trabalho ao meu avô, Laudelino Sousa (in memoriam), que a mim foi uma inspiração e motivação para correr atrás dos meus sonhos.

# Agradecimentos

À Deus pelo dom da vida e por me acompanhar nestes vinte e três anos de existência.

Aos meus pais e irmãos pela paciência nos momentos onde estive ausente e por toda ajuda provida quando necessário.

Aos meus amigos Aissi, Eric, João, Joab, Robert e Samuel que acompanharam de perto toda minha trajetória universitária até aqui e a todos os outros que em algum momento se fizeram presente em minha vida.

Ao Marcos Vinícius que desde criança tem sido um grande amigo, estando presente nos momentos bons e ruins. Se cheguei até aqui foi graças ao apoio e motivação que ele sempre me deu.

À minha professora orientadora Roberta Barbosa Oliveira que foi essencial durante todo esse período. Agradeço pela paciência, conselhos e pelo tempo dedicado a me orientar neste trabalho de conclusão.

Ao professor Alexandre Zaghetto que a mim foi uma inspiração desde os primeiros semestres que estive na UnB.

À Universidade de Brasília por promover um ensino público de qualidade, sendo referência no Brasil.

# Resumo

Um dos cânceres com a taxa de mortalidade mais alta no mundo é o câncer de pulmão. Há uma enorme dificuldade no diagnóstico desta doença em estágios iniciais, devido à ausência de sintomas claros. Por essa razão, ao identificá-la, o paciente possivelmente já encontra-se em estado avançado da doença ou até mesmo em processo metastático. Desta forma, com o objetivo de facilitar a detecção precoce e mitigar o número anual de mortes, encontra-se na *Convolutional Neural Network* (CNN) a possibilidade de utilizá-la como um apoio ao diagnóstico médico, em virtude dos bons resultados apresentados por esta arquitetura de rede neural na tarefa de classificação de imagens. Assim, no presente trabalho é feito um estudo comparativo utilizando os seguintes modelos de CNN para a tarefa de classificação de tomografias computadorizadas: VGG19, DenseNet, Inception e Xception. A partir da utilização destes modelos, objetiva-se verificar o comportamento dos modelos de CNN quando utilizados em bases de dados com classes desbalanceadas e balanceadas de tomografias computadorizadas do pulmão. Além disso, este trabalho também tem por objetivo a utilização das CNNs acima citadas aplicando a elas o conceito de transferência de aprendizado e modelo híbrido com classificadores tradicionais, como o *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), *Stochastic Gradient Descent* (SGD), Árvore de Decisão e Regressão Logística. Para fins de comparação entre os modelos de classificação, foram utilizadas as seguintes métricas: Acurácia, revocação, precisão e F1-score. No modelo de CNN completa, a arquitetura VGG teve o melhor desempenho, com 0.9724 de acurácia, 0.9537 de F1-score e 0.9582 de revocação. Para o modelo híbrido, os modelos VGG e Xception combinados com o algoritmo de Regressão Logística alcançaram os melhores desempenhos quando comparados aos modelos de CNN inicialmente utilizados. Por fim, realizadas as comparações é possível concluir que, para a tarefa de classificação de tomografias computadorizadas do pulmão, modelos híbridos podem ter melhor desempenho quando comparados aos modelos de CNN completa.

**Palavras-chave:** Redes Neurais Convolucionais, Classificação de Imagens, Modelos Híbridos, Câncer de Pulmão, Tomografias Computadorizadas

# Abstract

One of the cancers with the highest mortality rate in the world is lung cancer. There is a huge difficulty in diagnosing this disease at early stages, due to the absence of clear symptoms. For this reason, when it is identified, the patient is possibly already in an advanced stage of the disease or even in a metastatic process. Then, in order to facilitate early detection and mitigate the annual number of deaths, the possibility of using Convolutional Neural Networks (CNN) as support for medical diagnosis is being considered, due to the good results presented by this neural network architecture in the task of image classification. In this study, a comparative analysis is conducted using the following CNN models for the task of classifying computed tomography scans of the lungs: VGG19, DenseNet, Inception, and Xception. Through the use of these models, the aim is to verify the behavior of CNN models when used in databases with unbalanced and balanced classes of computed tomography scans of the lungs. In addition, this study also aims to apply the concept of transfer learning and hybrid models with traditional classifiers such as *K-Nearest Neighbors* (KNN), *Support Vector Machine* (SVM), *Stochastic Gradient Descent* (SGD), Decision Trees, and Logistic Regression to the previously mentioned CNN models. For the purpose of comparing the classification models, the following metrics were used: accuracy, recall, precision, and F1-score. In the complete CNN model, the VGG architecture had the best performance, with an accuracy of 0.9724, an F1-score of 0.9537, and a recall of 0.9582. For the hybrid model, the VGG and Xception models combined with the Logistic Regression algorithm achieved the best performance compared to the initially used CNN models. Finally, based on the comparisons, it is possible to conclude that for the task of classifying computed tomography scans of the lungs, hybrid models may have better performance compared to complete CNN models.

**Keywords:** Convolutional Neural Network, Image Classification, Hybrid Model, Lung Cancer, Computed Tomography

# Sumário

|          |                                                   |           |
|----------|---------------------------------------------------|-----------|
| <b>1</b> | <b>Introdução</b>                                 | <b>1</b>  |
| 1.1      | Objetivos . . . . .                               | 2         |
| 1.2      | Estrutura do documento . . . . .                  | 3         |
| <b>2</b> | <b>Fundamentação Teórica</b>                      | <b>4</b>  |
| 2.1      | Tomografia Computadorizada . . . . .              | 4         |
| 2.2      | Aprendizado de máquina . . . . .                  | 6         |
| 2.2.1    | Aprendizado Supervisionado . . . . .              | 6         |
| 2.2.2    | Classificadores . . . . .                         | 7         |
| 2.3      | Redes Neurais Convolucionais . . . . .            | 12        |
| 2.3.1    | Camadas Convolucionais . . . . .                  | 13        |
| 2.3.2    | Funções de Ativação . . . . .                     | 14        |
| 2.3.3    | Camadas de Pooling . . . . .                      | 15        |
| 2.3.4    | Camada Totalmente Conectada . . . . .             | 16        |
| 2.4      | Modelos de Redes Neurais Convolucionais . . . . . | 16        |
| 2.4.1    | VGG . . . . .                                     | 16        |
| 2.4.2    | Inception . . . . .                               | 17        |
| 2.4.3    | DenseNet . . . . .                                | 17        |
| 2.4.4    | Xception . . . . .                                | 18        |
| 2.5      | Aprendizado por Transferência . . . . .           | 18        |
| 2.6      | Balanceamento de dados . . . . .                  | 19        |
| 2.7      | Métricas de Avaliação de Desempenho . . . . .     | 20        |
| <b>3</b> | <b>Revisão de literatura</b>                      | <b>23</b> |
| <b>4</b> | <b>Metodologia</b>                                | <b>28</b> |
| 4.1      | Base de dados . . . . .                           | 29        |
| 4.2      | Etapa de pré-processamento de imagens . . . . .   | 30        |
| 4.3      | Experimentos propostos . . . . .                  | 35        |
| 4.3.1    | Modelo de CNN completa . . . . .                  | 35        |

|          |                                                       |           |
|----------|-------------------------------------------------------|-----------|
| 4.3.2    | Modelos híbridos . . . . .                            | 37        |
| <b>5</b> | <b>Resultados</b>                                     | <b>39</b> |
| 5.1      | Ferramentas utilizadas . . . . .                      | 39        |
| 5.2      | Experimento I - CNN completa . . . . .                | 40        |
| 5.2.1    | Modelo sem aumento de dados . . . . .                 | 40        |
| 5.2.2    | Modelo com balanceamento de dados . . . . .           | 45        |
| 5.3      | Experimento II - Modelo híbrido . . . . .             | 51        |
| 5.3.1    | Modelo híbrido sem balanceamento de classes . . . . . | 51        |
| 5.3.2    | Modelo híbrido com balanceamento de dados . . . . .   | 55        |
| <b>6</b> | <b>Conclusão e Trabalhos Futuros</b>                  | <b>64</b> |
|          | <b>Referências</b>                                    | <b>66</b> |



# Lista de Figuras

|      |                                                                                                                                     |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1  | Exemplo de formação de imagem tridimensional a partir de fatias de raio-x.                                                          | 5  |
| 2.2  | Exemplo de classificação de uma nova amostra utilizando o classificador KNN.                                                        | 8  |
| 2.3  | Exemplo de classificação binária utilizando o classificador SVM.                                                                    | 9  |
| 2.4  | Exemplo de classificação binária utilizando o classificador Árvore de Decisão.                                                      | 10 |
| 2.5  | Exemplo de um neurônio artificial.                                                                                                  | 11 |
| 2.6  | Exemplo de uma Rede Neural MLP.                                                                                                     | 12 |
| 2.7  | Arquitetura da Rede Neural Convolutiva LeNet.                                                                                       | 13 |
| 2.8  | Exemplo do processo de convolução entre um filtro $3 \times 3$ e uma entrada de tamanho $7 \times 7$ .                              | 13 |
| 2.9  | Gráfico exemplificando a função de ativação ReLU.                                                                                   | 14 |
| 2.10 | Gráfico exemplificando a função de ativação Softmax.                                                                                | 15 |
| 2.11 | Operação de Pooling máximo em um mapa de tamanho $4 \times 4$ .                                                                     | 15 |
| 2.12 | Modelo de CNN - VGG16.                                                                                                              | 17 |
| 2.13 | Esquemático representando uma DenseNet com 5 camadas.                                                                               | 18 |
| 2.14 | Matriz de Confusão para classificação de múltiplas classes.                                                                         | 21 |
| 4.1  | Fluxograma da metodologia a ser utilizada.                                                                                          | 28 |
| 4.2  | Exemplo de imagens das três classes encontradas na base de dados IQ-OTHNCCD.                                                        | 30 |
| 4.3  | Exemplo de imagem gerada pela técnica de <i>Oversample</i> .                                                                        | 32 |
| 4.4  | Exemplo de imagem gerada pela técnica SMOTE.                                                                                        | 32 |
| 4.5  | Exemplo de imagens geradas a partir da utilização de métodos de aumento de dados.                                                   | 33 |
| 4.6  | Modelo de CNN pré-treinada com a base de dados ImageNet.                                                                            | 34 |
| 4.7  | Modelo híbrido com pesos congelados na fase convolutiva, realizando a extração de características e posteriormente a classificação. | 34 |
| 4.8  | Fluxograma relacionado ao primeiro experimento.                                                                                     | 36 |
| 4.9  | Esquemático do modelo híbrido utilizado no segundo experimento.                                                                     | 37 |

|      |                                                                                                            |    |
|------|------------------------------------------------------------------------------------------------------------|----|
| 5.1  | Matriz de confusão dos modelos sem aumento de dados com função de perda Focal Loss. . . . .                | 42 |
| 5.2  | Matriz de confusão dos modelos sem aumento de dados com função de perda Categorical Cross Entropy. . . . . | 42 |
| 5.3  | Curva de acurácia dos modelos utilizando a função de perda Focal Loss. . .                                 | 43 |
| 5.4  | Curva de perda dos modelos utilizando a função de perda Focal Loss. . . .                                  | 43 |
| 5.5  | Curva de acurácia dos modelos utilizando a função de perda Categorical Cross Entropy. . . . .              | 44 |
| 5.6  | Curva de perda dos modelos utilizando a função de perda Categorical Cross Entropy. . . . .                 | 44 |
| 5.7  | Matriz de confusão dos modelos com balanceamento de classes utilizando <i>Oversample</i> . . . . .         | 46 |
| 5.8  | Matriz de confusão dos modelos com balanceamento de classes utilizando SMOTE. . . . .                      | 46 |
| 5.9  | Matriz de confusão dos modelos com <i>Oversample</i> e aumento de dados. . . .                             | 48 |
| 5.10 | Curva de acurácia dos modelos utilizando balanceamento de classes <i>Oversample</i> . . . . .              | 48 |
| 5.11 | Curva de perda dos modelos utilizando balanceamento de classes <i>Oversample</i> . . . . .                 | 49 |
| 5.12 | Curva de acurácia dos modelos utilizando balanceamento de classes SMOTE. . . . .                           | 49 |
| 5.13 | Curva de perda dos modelos utilizando balanceamento de classes SMOTE. . . . .                              | 50 |
| 5.14 | Matriz de confusão do modelo híbrido KNN sem balanceamento de classes. . . . .                             | 52 |
| 5.15 | Matriz de confusão do modelo híbrido de Regressão Logística sem balanceamento de classes. . . . .          | 53 |
| 5.16 | Matriz de confusão do modelo híbrido SGDC sem balanceamento de classes. . . . .                            | 53 |
| 5.17 | Matriz de confusão do modelo híbrido Árvore de Decisão sem balanceamento de classes. . . . .               | 54 |
| 5.18 | Matriz de confusão do modelo híbrido SVM sem balanceamento de classes. . . . .                             | 54 |
| 5.19 | Matriz de confusão do modelo híbrido KNN com balanceamento de classes SMOTE. . . . .                       | 56 |
| 5.20 | Matriz de confusão do modelo híbrido de Regressão Logística com balanceamento de classes SMOTE. . . . .    | 57 |
| 5.21 | Matriz de confusão do modelo híbrido de Árvore de Decisão com balanceamento de classes SMOTE. . . . .      | 57 |
| 5.22 | Matriz de confusão do modelo híbrido SGDC com balanceamento de classes SMOTE. . . . .                      | 58 |
| 5.23 | Matriz de confusão do modelo híbrido SVM com balanceamento de classes SMOTE. . . . .                       | 58 |

|      |                                                                                                                   |    |
|------|-------------------------------------------------------------------------------------------------------------------|----|
| 5.24 | Matriz de confusão do modelo híbrido KNN com balanceamento de classes<br>Oversample. . . . .                      | 60 |
| 5.25 | Matriz de confusão do modelo híbrido de Regressão Logística com balan-<br>ceamento de classes Oversample. . . . . | 61 |
| 5.26 | Matriz de confusão do modelo híbrido de Árvore de Decisão com balance-<br>amento de classes Oversample. . . . .   | 61 |
| 5.27 | Matriz de confusão do modelo híbrido SGDC com balanceamento de classes<br>Oversample. . . . .                     | 62 |
| 5.28 | Matriz de confusão do modelo híbrido SVM com balanceamento de classes<br>Oversample. . . . .                      | 62 |

# Lista de Tabelas

|     |                                                                                                                                                        |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | Resumo dos trabalhos abordados neste capítulo. . . . .                                                                                                 | 27 |
| 4.1 | Tabela com parâmetros customizados utilizados nos classificadores. . . . .                                                                             | 37 |
| 5.1 | Resultados alcançados com a utilização das funções de perda <i>Sigmoid Focal Cross Entropy</i> e <i>Categorical Cross Entropy</i> . . . . .            | 41 |
| 5.2 | Métricas alcançadas com a utilização dos métodos SMOTE e Oversample para balanceamento de dados. . . . .                                               | 45 |
| 5.3 | Métricas alcançadas com a utilização do SMOTE para balanceamento de classes e aumento de dados para gerar novas imagens durante o treinamento. . . . . | 47 |
| 5.4 | Resumo de melhores métricas obtidas nos experimentos. . . . .                                                                                          | 50 |
| 5.5 | Métricas dos modelos híbridos sem balanceamento de classes. . . . .                                                                                    | 52 |
| 5.6 | Modelos híbridos com classificador com balanceamento de classes utilizando SMOTE. . . . .                                                              | 56 |
| 5.7 | Modelos híbridos com classificador com balanceamento de classes utilizando Oversample. . . . .                                                         | 60 |
| 5.8 | Resumo de melhores métricas obtidas nos experimentos com modelos híbridos. . . . .                                                                     | 63 |

# Lista de Abreviaturas e Siglas

**ANN** Artificial Neural Network.

**CE** *Cross Entropy.*

**CNN** *Convolutional Neural Network.*

**CNPC** Câncer de não Pequenas Células.

**CPC** Câncer de Pequenas Células.

**CPM** *Competition Performance Metric.*

**CT** *Computed Tomography.*

**DITNN** *Deep Learning Instantaneously Trained Neural Network.*

**DNN** *Deep Neural Network.*

**IA** Inteligência Artificial.

**IQ-OTHNCCD** *Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases lung cancer dataset.*

**KNN** *K-Nearest Neighbors.*

**LIDC-IDRI** *Lung Image Database Consortium Image Collection.*

**MLP** *Multilayer Perceptron.*

**ReLU** *Rectified Linear Unit.*

**SGD** *Stochastic Gradient Descent.*

**SMOTE** *Synthetic Minority Over-sampling Technique.*

**SVM** *Support Vector Machine.*

# Capítulo 1

## Introdução

De acordo com a Sociedade Americana de Câncer [1], o câncer de pulmão é o segundo mais comum no mundo (excluindo lesões de pele), ficando atrás somente do câncer de próstata em homens e câncer de mama em mulheres. Sendo este câncer um dos principais responsáveis por mortes causadas por câncer, estima-se que em 2023 haverão mais de 230 mil novos casos e aproximadamente 127 mil mortes por esta doença nos Estados Unidos [1]. Neste país, 20% dos casos de morte por câncer são de câncer de pulmão, sendo esta uma taxa de mortalidade maior que as taxas de mortalidade por câncer de próstata, cólon e mama combinadas. Em 2020, no Brasil, este tipo de câncer foi, em homens e mulheres, o terceiro e quarto mais incidente respectivamente. Além disso, neste mesmo ano, houveram 28.620 mortes em decorrência de câncer de pulmão [2].

Existem dois principais tipos para esta doença, sendo eles o Câncer de não Pequenas Células (CNPC) e o Câncer de Pequenas Células (CPC) [1]. O primeiro tipo equivale a 80% dos casos de câncer, sendo eles classificados em três subtipos: Carcinoma de células escamosas, Carcinoma de grandes células e Adenocarcinoma. O CPC, apesar de apresentar uma menor taxa, é o tipo de câncer que pode ocorrer metástase em um menor tempo, uma vez que este se espalha rapidamente pelo pulmão.

O câncer de pulmão ocorre quando há uma mutação nas células do pulmão. Vários fatores influenciam diretamente na ocorrência dessa mudança, e estes normalmente estão ligados à exposição de substâncias tóxicas. Entre estes fatores pode-se citar o ato de fumar, que é considerado o principal gatilho para a doença em questão, uma vez que ao tragar tabaco, junto a ele diversas substâncias químicas consideradas maléficas para o pulmão são absorvidas. Além de fumar, a causa do câncer de pulmão também está relacionada à exposição ao radônio, à poluição do ar e às causas hereditárias [1].

A causa do câncer de pulmão ter a maior taxa de letalidade entre todos os outros tipos de câncer está relacionada à grande dificuldade que têm os médicos em diagnosticar esta doença, visto que a mesma não apresenta sintomas aparentes nos estágios iniciais. Assim,

quando obtém êxito na detecção, possivelmente este já está em estágios avançados ou até mesmo em estado metastático.

Sabendo disso, há na Inteligência Artificial a possibilidade de atuar utilizando técnicas que permitam a detecção e classificação de nódulos pulmonares ainda nos estágios iniciais. Portanto, no aprendizado de máquina as Redes Neurais Convolucionais, do inglês *Convolutional Neural Network* (CNN), se destacam nas tarefas onde são trabalhadas imagens [3]. Assim, com a utilização de CNN é possível facilitar o processo de diagnóstico de câncer pulmonar de forma precoce, com o intuito de reduzir substancialmente o alto número de mortes anuais.

Com o foco em propor soluções para a dificuldade de diagnóstico precoce de câncer de pulmão, na literatura modelos de CNN com foco em classificação de imagens têm sido utilizados, alcançando bons resultados, como indicam os trabalhos de Zhang *et al.* [4] e Togaçar *et al.* [5]. Além disso, modelos híbridos como o proposto por Polat *et al.* [6] têm demonstrado bons desempenhos quando utilizados modelos de CNN juntamente a classificadores tradicionais de aprendizado de máquina.

Em vista disso, este trabalho propõe um estudo comparativo utilizando diferentes modelos de CNN e modelos híbridos combinados a técnicas de aumento de dados e balanceamento de classes, focando na tarefa de classificação de imagens do pulmão. A base de dados utilizada, disponibilizada por um hospital Iraquiano, contém imagens de pulmões com tumor maligno e benigno, rotuladas manualmente por oncologistas e também contém imagens de pulmão saudável para possibilitar a classificação de múltiplas classes.

## 1.1 Objetivos

O principal objetivo do presente trabalho é auxiliar na classificação de imagens de tomografia computadorizada do pulmão a partir da análise de diferentes modelos de CNN completos e híbridos. Para o integral cumprimento do objetivo principal aqui citado, objetivos específicos foram considerados, são eles:

- Analisar a eficácia de diferentes modelos de CNN para extração de características e classificação de imagens do pulmão;
- Verificar, pesquisar e comparar funções de perda com o objetivo de melhorar curvas de aprendizado de modelos de CNN;
- Analisar o impacto em diferentes CNNs com a utilização de estratégias para lidar com o desbalanceamento de classes em problema de classificação entre diferentes tipos de câncer de pulmão;

- Desenvolver e avaliar o comportamento de diferentes modelos híbridos com classes balanceadas e desbalanceadas para problema de classificação multi-classe.

## **1.2 Estrutura do documento**

O presente documento está disposto da seguinte maneira: No Capítulo 2 são apresentados todos os conceitos teóricos necessários para o pleno entendimento deste trabalho. No Capítulo 3 é evidenciada a revisão de literatura, em que foram considerados estudos relacionados a doenças no pulmão que serviram de base para a proposição deste trabalho. No Capítulo 4 descreve toda a metodologia proposta para alcançar os objetivos. No Capítulo 5 são expostos e discutidos todos os resultados obtidos nos experimentos realizados e as devidas comparações entre os modelos. Por fim, no Capítulo 6 apresenta a conclusão sobre os resultados obtidos nos experimentos e sugestões para trabalhos futuros.



# Capítulo 2

## Fundamentação Teórica

O presente capítulo objetiva apresentar os principais conceitos teóricos que ao longo deste trabalho serão necessários para assim proporcionar o integral entendimento desta monografia. Inicialmente será apresentada a principal técnica de aquisição de imagens de pulmão. Logo após, serão definidos conceitos de aprendizado de máquina, tais como aprendizado supervisionado, classificadores e redes neurais convolucionais e seus modelos que serão utilizados neste trabalho. Em seguida, será definido o conceito de modelos híbridos e por fim, serão abordados os conceitos relacionados ao pré-processamento de imagens e as principais métricas que são utilizadas por diversos autores para avaliar o desempenho dos modelos propostos na literatura.

### 2.1 Tomografia Computadorizada

A tomografia computadorizada (CT) é um procedimento utilizado para obtenção de imagens médicas. Feixes de radiação são lançados e circulam o corpo de um paciente, tornando possível assim a obtenção de sinais que posteriormente são processados, possibilitando a formação da imagem computadorizada de uma determinada parte do corpo [7].

O processo de aquisição das imagens permite que sejam geradas várias imagens que se referem a fatias de uma determinada área e estas podem ser empilhadas formando, portanto, uma imagem tridimensional da área em questão. A Figura 2.1 representa forma como uma imagem tridimensional é construída a partir das fatias obtidas em tomografias computadorizadas.

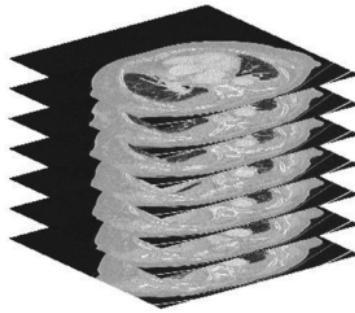


Figura 2.1: Exemplo de formação de imagem tridimensional a partir de fatias de raio-x (Fonte: [8]).

Para a captura de imagens do pulmão, tomografias computadorizadas normalmente são indicadas por médicos ao perceberem, durante os exames de raio-x, alguma anormalidade como tumores, hemorragias internas, lesões internas e algum tipo de dano em uma determinada região. Ao final do processo de aquisição de fatias de imagens do pulmão, são geradas imagens no formato DICOM que disponibiliza diversas informações, tanto sobre a área focada quanto sobre o paciente.

DICOM, que refere-se a *Digital Imaging and Communications in Medicine* (Comunicação Digital de Imagens em Medicina), é um protocolo que surgiu com a proposta de padronizar a comunicação de diagnósticos e informações na medicina. Este foi criado em 1983 a partir de uma ação conjunta do Colégio Americano de Radiologia e da Associação Nacional de Fabricantes Elétricos [9].

Baseado nos protocolos de comunicação ISO/OSI e TCP/IP, o padrão DICOM é envolvido por várias camadas, sendo elas: camada de estrutura de dados para imagens médicas, camada de serviços orientados a redes, camada de formato de mídias e para troca de dados, camada de gerenciamento de fluxo, camada de consistência e qualidade de apresentação e camada de requisitos. Na camada de estrutura de dados há o chamado *Information Object Definition* (Objeto de Definição de Informações) que é o principal componente desta camada. Nele, pode-se encontrar um cabeçalho onde disponibiliza todas as informações necessárias acerca de um paciente, tais como idade, sexo, comorbidades e outras informações que podem ser úteis para levantamento de dados e detecção de padrões de doenças.

Há no aprendizado de máquina a possibilidade de projetar modelos capazes de realizar diagnóstico automático utilizando o tipo de imagem aqui citado uma vez que pequenas mudanças em imagens DICOM podem ser detectadas a partir de algoritmos de aprendizado de máquina.

## 2.2 Aprendizado de máquina

De acordo com Carmo [10], “a aprendizagem é um fenômeno complexo caracterizado por mudanças observáveis, mensuráveis e permanentes ou duradouras no repertório de um indivíduo e que permite registrar o desempenho e traçar gráficos que demonstrem o maior ou menor progresso deste indivíduo”. Trazendo esse conceito para o universo da computação, pode-se dizer que o aprendizado de máquina refere-se à capacidade de adaptação a mudanças existentes em um sistema computacional. Essa adaptação se dá pelo fato de sistemas inteligentes serem capazes de prever comportamentos futuros a partir de informações passadas. Em suma, pode-se definir o aprendizado de máquina a partir do conceito definido por Franco [11]: “aprendizado de máquina é uma subárea da IA, cujo objetivo é ensinar os computadores a aprenderem de forma automática”.

No cotidiano estamos em constante contato com o aprendizado de máquina. Por exemplo, quando se pesquisa um vídeo em alguma plataforma, é bem provável que vídeos subsequentes estejam relacionados ao que foi pesquisado anteriormente. Há por trás da simples pesquisa algoritmos de aprendizado de máquina que reconhecem padrões utilizados pelo usuário para fazerem predições de escolhas e, se necessário, recomendações. Além deste exemplo há diversas outras utilizações do conceito de aprendizado de máquina. Para o trabalho aqui apresentado, um sistema aprende a partir de imagens médicas padrões para assim ser capaz de discernir e relacionar uma imagem a uma determinada classe.

Carmo [10] exemplifica que no aprendizado tradicional há diversas formas de se aprender, por exemplo, aprender através da escuta, da imitação, da leitura ou da escrita. No aprendizado de máquina não é diferente, uma vez que existem três principais modos de aprendizado, sendo eles aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço. A fim de orientar o leitor sobre os conceitos que serão abordados durante este trabalho, a subseção a seguir irá definir e exemplificar o aprendizado supervisionado.

### 2.2.1 Aprendizado Supervisionado

O aprendizado supervisionado é um dos tipos de aprendizado de máquina que, a partir de um conjunto de dados de treinamento, possibilita a predição de padrões. Este aprendizado requer que haja um conjunto com dados totalmente diferentes dos conhecidos durante o treinamento, com o fim de validar se um determinado algoritmo desempenha como o esperado.

Com respeito a um problema de classificação de imagens pode-se exemplificar o aprendizado supervisionado da seguinte forma:

Dado um conjunto de treinamento contendo  $N$  imagens e seus respectivos rótulos,

$$(imagem_1, rótulo_1), (imagem_2, rótulo_2), \dots (imagem_N, rótulo_N), \quad (2.1)$$

sendo que cada rótulo  $i$  é gerado a partir de uma função dada por:

$$rótulo_i = f(imagem_i), \quad (2.2)$$

o aprendizado supervisionado tem por objetivo encontrar uma função hipótese que aproxime-se ao real resultado da função  $f$  (Equação 2.2). Esta função hipótese busca um resultado ótimo relacionado ao mapeamento entre a imagem e seu respectivo rótulo, usando como base o conjunto de treinamento e o conjunto de testes [12].

## 2.2.2 Classificadores

A partir do conceito apresentado na subseção anterior, pode-se introduzir um dos principais conceitos a serem utilizados ao longo deste trabalho: classificação. A classificação, em aprendizado supervisionado, refere à capacidade que um sistema inteligente tem em prever uma saída a partir de um valor de entrada. Para essa tarefa, ao longo da história da computação e da matemática diversos métodos foram propostos com o fim de maximizar o desempenho de sistemas computacionais, métodos tais como, K-Vizinhos mais próximos, Máquina de Vetores de Suporte, Gradiente Descendente Estocástico, Árvore de Decisão e Regressão Logística.

### K-Vizinhos mais próximos

K-Vizinhos mais próximos (KNN) é um classificador de aprendizagem baseada em instâncias que considera a Distância Euclidiana entre duas amostras. Neste classificador não há a presença de um modelo, assim, para alcançar a classificação não são realizados treinamentos como em outros métodos classificadores. Na Equação 2.3 demonstra como é o comportamento obtido durante a tarefa de classificação. Neste caso, para uma amostra  $n$ -dimensional, a variável  $x^{nova}$  refere-se a uma nova amostra a ser classificada e a variável  $x$  refere-se a uma amostra vizinha à nova amostra.

$$distância(x^{nova}, x) = \sqrt{\sum_{j=1}^n (x_j^{nova} - x_j)^2}. \quad (2.3)$$

Este método tem como regra que o rótulo que será classificado para uma amostra de teste será igual à classe de maior ocorrência dentre os  $k$  vizinhos mais próximos das

amostras de treinamento. Portanto, fica clara a necessidade de ajustar o valor  $k$  para obter melhores resultados de acordo com o problema.

A Figura 2.2 exemplifica como um algoritmo projetado com o valor de  $k$  sendo igual a três faria a classificação de uma nova amostra. Percebe-se que, na Figura 2.2(b), das três amostras mais próximas, duas referem-se à Classe  $B$ , sendo assim, o resultado da predição seguirá a classe de maior peso dentre os vizinhos, neste caso, a Classe  $B$ .

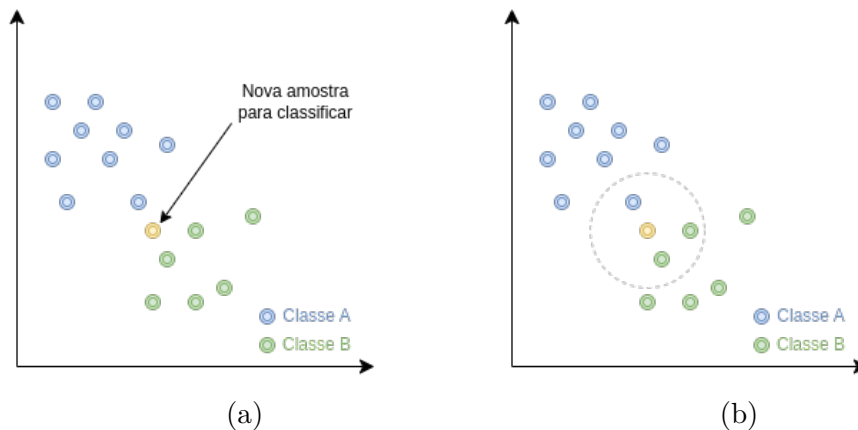


Figura 2.2: Exemplo de classificação de uma nova amostra utilizando o classificador KNN.

## Máquina de Vetores de Suporte

Assim como o método KNN mencionado anteriormente, a Máquina de Vetores de Suporte, do inglês *Support Vector Machine* (SVM), é um método de classificação baseado em aprendizado supervisionado que foi proposto por Vapnik [13] em 1995 em seu livro onde aborda a chamada teoria do aprendizado estatístico. Inicialmente projetado para tarefas de classificação binária, o método SVM implementa o hiperplano (Equação 2.4) chamado de separador de margem máxima que permite a otimização dos limites que separam conjuntos de amostras, permitindo assim uma melhor generalização do problema. Este hiperplano é definido como um limite de decisão que tem a maior distância possível das amostras a serem classificadas. A Figura 2.3 exemplifica o funcionamento do método de classificação SVM.

$$W^T X = 0, \quad (2.4)$$

sendo:

$W$  = Vetor normal ao hiperplano;

$X$  = Amostras de classes.

No gráfico da Figura 2.3(a) pode ser visto um problema de classificação binária de duas classes, representadas pelos pontos pretos e brancos. Neste gráfico há três limites que,

apesar de estarem classificando cada amostra em sua classe correta, há a possibilidade de perda quando classificadas novas amostras por não haver um limiar bem definido, possibilitando assim a ocorrência de classificações erradas. Assim, na imagem da Figura 2.3(b) surge como uma solução utilizando a SVM, onde, usando o separador de margem máxima, reduz significativamente a perda sobre os dados de treinamento.

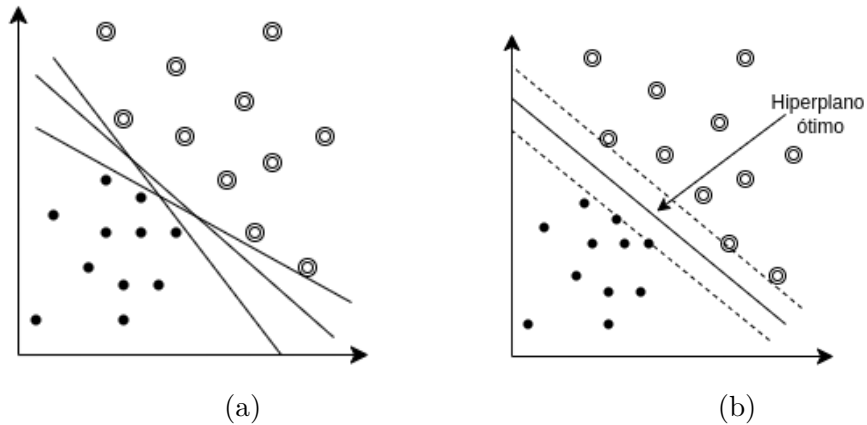


Figura 2.3: Exemplo de classificação binária utilizando o classificador SVM.

### Gradiente Descendente Estocástico

O método de classificação Gradiente Descendente Estocástico (SGD), também conhecido pelo conceito de retropropagação, é um método de classificação que propõe a redução de erros a partir da atualização de pesos em iterações até que se alcance a convergência [14]. A Equação 2.5 exemplifica como variáveis são atualizadas durante as  $i$  iterações.

$$w_{i+1} = w_i - \gamma_i \nabla_w Q(z_i, w_i), \quad (2.5)$$

no qual:

- $w$  = vetor de pesos;
- $\gamma$  = taxa de aprendizado;
- $x$  = amostra;
- $z = (x, \text{rótulo})$ ;
- $Q(z_t, w_t)$  = função de erro.

### Árvore de decisão

Árvore de decisão é uma função amplamente difundida na área da ciência da computação por sua simplicidade e sucesso [12]. Esta função recebe um vetor de atributos na entrada e retorna uma decisão. Para o caso do presente trabalho, este classificador receberá um vetor representando as características essenciais de uma imagem e retornará um valor relativo à

classe prevista para esta imagem. Para cada nó da árvore de decisão são definidos pesos, a partir do conceito de entropia, que são computados e a decisão realizada por este algoritmo é alcançada após diversos testes executados pelo método de dividir-para-conquistar feitos em cada um desses nós. Para o caso de classificação, neste método, um problema de classificação é dividido em vários subproblemas e, recursivamente, obtém o resultado da classificação. A Figura 2.4 exemplifica o funcionamento da função de árvore de decisão. Neste exemplo é simulado um algoritmo que decide se deverá ou não esperar uma mesa em um restaurante. No presente trabalho será utilizado o classificador de árvore de decisão utilizando como critério a função Gini. Nela, será medido o índice de heterogeneidade dos dados, permitindo medir a impureza de cada um dos nós.

$$Gini = 1 - \sum_{i=1}^n (p_i)^2, \quad (2.6)$$

sendo:

$n$  = número de classes;

$p$  = frequência relativa de cada classe em um nó.

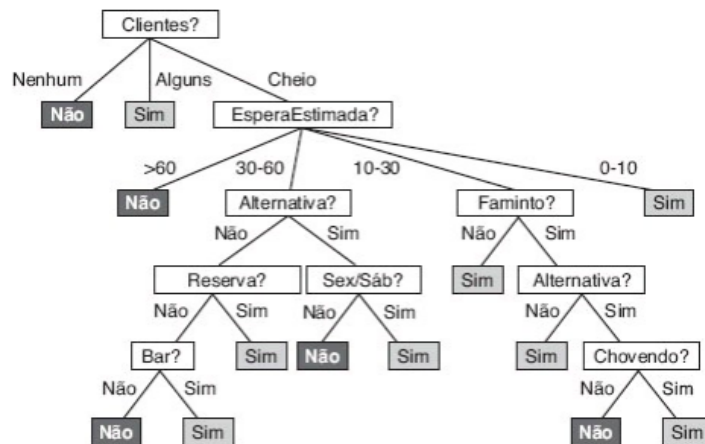


Figura 2.4: Exemplo de classificação binária utilizando o classificador Árvore de Decisão (Fonte: [12]).

## Regressão Logística

Também conhecida como modelo Logit, a classificação por regressão logística é um método de aprendizado supervisionado que, a partir de uma base de dados, permite estimar a possibilidade de ocorrência de um evento. A equação 2.7 demonstra como é obtida uma probabilidade para uma amostra  $x$  a partir do modelo de regressão logística.

$$\text{logit}(x) = \frac{1}{1 + \exp^{-x}}, \quad (2.7)$$

assim, sabendo que a saída da função  $\text{logit}(x)$  será um resultado entre 0 e 1 pode-se interpretar esta saída como a probabilidade de uma amostra pertencer a uma determinada classe [12].

## Perceptron Multicamadas

Antes de abordar o classificador baseado em *Multilayer Perceptron* (MLP), é necessário inicialmente introduzir o conceito de Redes Neurais Artificiais (ANN). Inspiradas no cérebro humano (especificamente no comportamento dos neurônios) as ANNs surgem com o intuito de simular o comportamento de neurônios biológicos. A Figura 2.5 ilustra o modelo proposto do que ficou conhecido como neurônio artificial, também chamado de Perceptron. Nesta Figura,  $n$  entradas adquirem pesos respectivos a cada uma delas e é feita a computação de um valor a partir da função de ativação.

O neurônio artificial é composto por suas entradas representadas por  $X$  e seus respectivos pesos representados por  $W$ . A entrada bias é um parâmetro cujo objetivo é, dependendo do seu sinal, alterar a entrada da função de ativação. Prosseguindo, há a fase onde são somados os produtos das entradas por seus pesos e por fim, esse resultado passa pela função de ativação que definirá o valor de saída.

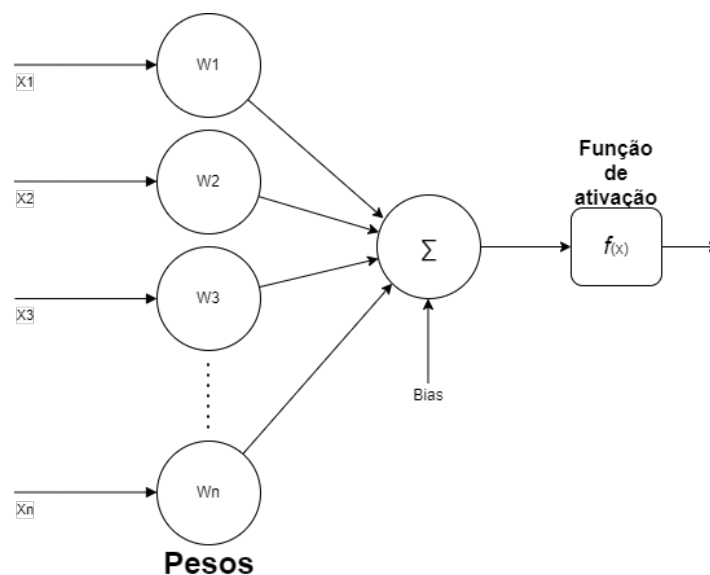


Figura 2.5: Exemplo de um neurônio artificial (Fonte: [15]).

Portanto, o MLP (Figura 2.6) é uma classe de redes neurais onde há uma camada de entrada e uma ou mais camadas ocultas e tem  $n$  saídas, sendo  $n$  o número de classes que amostras são rotuladas. A classificação a partir de MLP é feita utilizando-se do algoritmo



de retro-propagação do erro, onde o mesmo força um ajuste dos pesos <sup>1</sup> das camadas de acordo com um sinal de erro recebido por camadas mais à direita [16]. Neste trabalho, será utilizada uma MLP com o foco em classificação de múltiplas classes. Nela, será utilizada uma camada com três perceptrons que ao receberem os vetores de entrada relacionados às características das imagens, definirão na saída um resultado relativo à classe prevista.

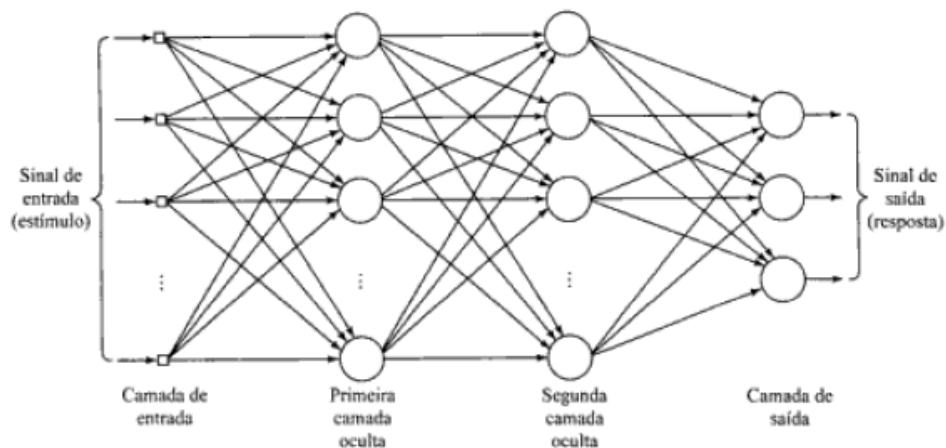


Figura 2.6: Exemplo de uma Rede Neural MLP (Fonte: [16]).

## 2.3 Redes Neurais Convolucionais

As *Convolutional Neural Network* (CNN) são um tipo de Redes Neurais de aprendizado profundo que se sobressai no que se refere a dados multidimensionais, como por exemplo imagens. Embora o conceito de CNNs tenha sido abordado inicialmente em 1959 por Hubel e Wisel [17], as CNNs começaram a ser difundidas no ano de 2012, quando, na competição *ImageNet Large Scale Visual Recognition Challenge [ILSVRC]*, o modelo AlexNet obteve sucesso com grande margem em relação a outros métodos [18]. Diferentemente das ANNs padrões, as CNNs tradicionais são compostas por três diferentes camadas, sendo elas: camadas convolucionais, camadas de pooling e por fim a camada totalmente conectada, que refere-se ao MLP, citado na Seção 2.2.1. A Figura 2.7, que é relacionada ao primeiro modelo de CNNs desenvolvido, conhecido como LeNet [19], demonstra a disposição de cada uma das camadas de uma CNN padrão: convolução, pooling e camada totalmente conectada. Nas subseções a seguir, serão detalhadas estas camadas e suas tarefas.

<sup>1</sup>O reajuste dos pesos é feito a partir do cálculo das derivadas parciais do erro anterior em função do peso atual.

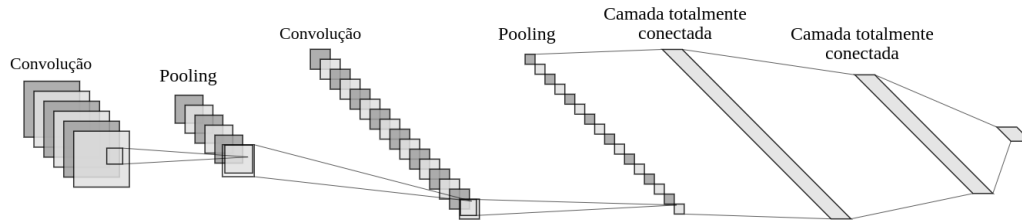


Figura 2.7: Arquitetura da Rede Neural Convolucional LeNet.

### 2.3.1 Camadas Convolucionais

A camada convolucional tem um papel muito importante em guiar como modelos de CNN irão operar. Com o uso de *kernels* esta camada possibilitará a geração de novas entradas para camadas seguintes. *Kernels*, também chamados de filtros, se referem a conjuntos de unidades tridimensionais que, durante o treinamento de uma CNN, têm seus pesos atualizados com o objetivo de aprender a extrair características importantes sobre as imagens[19].

A operação de convolução é um tipo de filtro linear que pode ser aplicado em imagens para extração de características. Para este fim, são aplicados filtros em cada posição possível de uma imagem de entrada e realiza um produto entre os pesos destes filtros e uma parcela do vetor de entrada com mesmas dimensões [20]. Na Figura 2.8 é exemplificada a operação de convolução, onde, utilizando um filtro com profundidade unitária e dimensões  $3 \times 3$ , realiza o produto escalar na entrada que possui dimensões  $7 \times 7$  e também dispõe da saída com uma redução nas dimensões iniciais resultante da operação de produto escalar entre o vetor de entrada e o filtro.

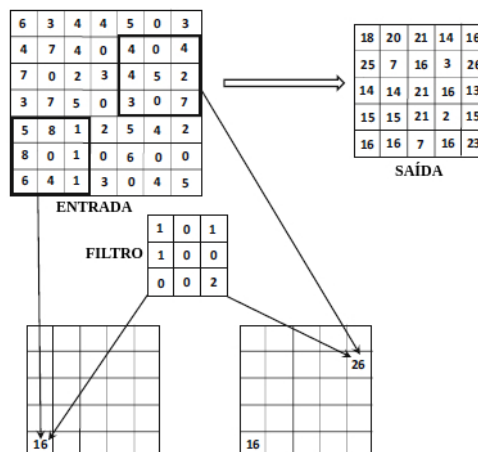


Figura 2.8: Exemplo do processo de convolução entre um filtro  $3 \times 3$  e uma entrada de tamanho  $7 \times 7$  (Fonte: [18]).

## 2.3.2 Funções de Ativação

As funções de ativação desempenham um papel fundamental em CNNs. Como citado na subseção 2.2.1 a respeito da MLP, durante o processo de treinamento de uma rede, funções de ativação são utilizadas para, com a ajuda de métodos matemáticos, definirem a saída de um Perceptron [18]. Em arquiteturas de CNNs após cada camada convolucional e camada totalmente conectada funções de ativação são utilizadas. Nesta subseção serão detalhadas duas das mais utilizadas funções de ativação [19]: ReLU e Softmax.

### Unidade Linear Retificada

Desde a primeira camada de convolução até a camada totalmente conectada pode-se ter a presença de ativadores não-lineares, sendo uma das mais utilizadas a função *Rectified Linear Unit* (ReLU) [18]. Com o propósito de aumentar a não-linearidade de uma entrada, o ativador ReLU aplica uma função que retorna o valor máximo entre 0 (zero) e o valor de entrada (Equação 2.8). O gráfico da Figura 2.9 exemplifica como aplicando-se esta função de ativação se obtém resultados positivos e permite o aumento da não-linearidade durante as etapas de convolução.

$$y = \max(0, x), \quad (2.8)$$

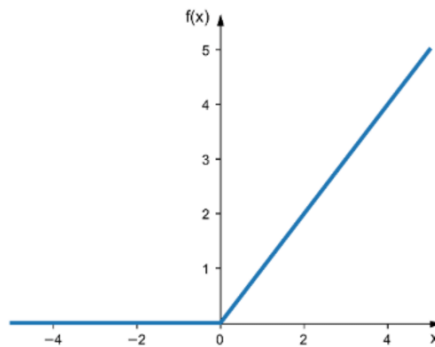


Figura 2.9: Gráfico exemplificando a função de ativação ReLU (Fonte: [21]).

### Softmax

Quando tratando-se de uma classificação de mais de duas classes, se faz necessária a utilização de uma função de ativação especial, pois há a necessidade de se computar a saída a partir de  $n$  entradas [18]. O softmax, conhecido também como regressão logística multinomial, é um método de classificação que tem por característica retornar probabilidades a partir de entradas multidimensionais, como mostra o gráfico da Figura 2.10. Nesta

Figura, os eixos representam uma entrada e sua saída após a utilização da função de ativação Softmax. Na Equação 2.9 demonstra como é realizado o cálculo da probabilidade da  $i$ -ésima imagem.

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}, \quad (2.9)$$

$x$  = vetor de entrada;

$N$  = número de classes.

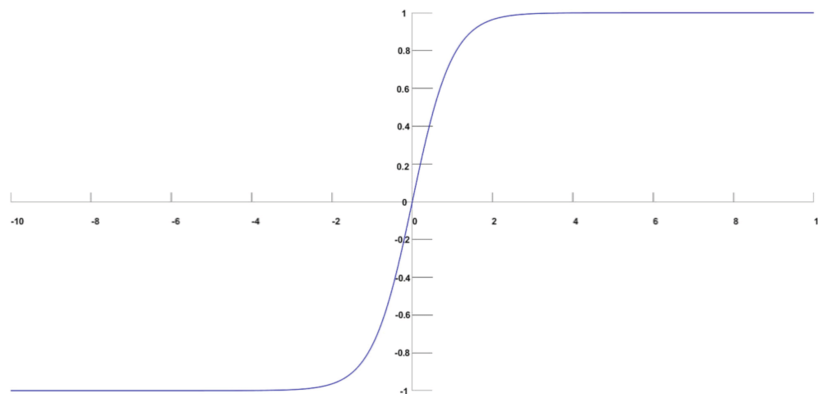


Figura 2.10: Gráfico exemplificando a função de ativação Softmax (Fonte: [22]).

### 2.3.3 Camadas de Pooling

A camada de Pooling, também conhecida como *Downsampling* (subamostragem), é responsável por reduzir as amostras dos mapas de características retornados das operações de convolução, mantendo sempre as mais dominantes. Há diferentes técnicas de Pooling, no entanto duas são as mais utilizadas [19]: Pooling máximo e Pooling por média.

Pooling máximo sendo o mais popular e mais utilizado [19], mapeia uma determinada região e extrai o seu valor máximo, enquanto o Pooling por média mapeia uma determinada região e extrai o seu valor médio. A imagem 2.11 exemplifica como uma operação utilizando um Pooling máximo com tamanho  $2 \times 2$  reduz o mapa de característica de uma imagem.

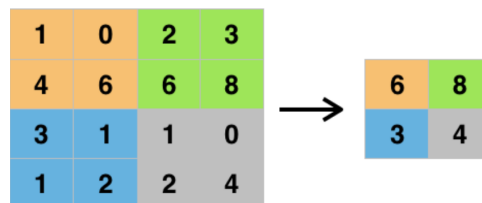


Figura 2.11: Operação de Pooling máximo em um mapa de tamanho  $4 \times 4$  (Fonte: [23]).

### 2.3.4 Camada Totalmente Conectada

Por fim, após a CNN ter suas características extraídas e reduzidas nas camadas de Pooling e de convolução, é utilizada a camada totalmente conectada com o objetivo de classificar um conjunto de imagens. Esta camada refere-se a um MLP que recebe as características extraídas em um formato vetorial em sua entrada, aplica-se a este vetor um produto escalar por seus pesos e por fim é aplicada a função de ativação<sup>2</sup>, como detalhado na Subseção 2.3.2, que retornará a probabilidade de uma imagem pertencer a determinada classe.

## 2.4 Modelos de Redes Neurais Convolucionais

Na seção anterior 2.3 foi introduzido o conceito de CNN e suas camadas básicas. A partir da combinação destas camadas, obtém-se o que é conhecido como modelo de CNN. Ao longo da história, desde o primeiro modelo de CNN, o LeNet [19], diversos outros tipos de modelos foram propostos para atuação em diferentes tarefas, por exemplo segmentação, detecção e classificação em imagens. Nesta seção serão apresentados os modelos que foram utilizados ao longo deste trabalho, para a tarefa de classificação de imagens, sendo eles: VGG, Inception, DenseNet e Xception.

### 2.4.1 VGG

*Very Deep Convolutional Network* mais conhecido como VGG é um modelo de CNN proposto por Karen e Andrew [24] em 2014 que teve como objetivo atingir o estado da arte quando utilizado na base de dados ImageNet em tarefas de classificação e detecção. Por fim, obtiveram sucesso na proposta onde alcançaram a marca de 92.7% de acurácia na base de dados do desafio ILSVRC [25]. Este modelo traz consigo uma rede profunda onde pode possuir dezesseis (VGG16) ou dezenove (VGG19) camadas e possibilita a classificação de imagens em 1000 classes. O modelo VGG16 possui treze camadas convolucionais, cinco camadas de pooling máximo e três camadas totalmente conectadas tendo assim, cerca de 138 milhões de parâmetros. A imagem da Figura 2.12 ilustra a organização das camadas deste modelo.

---

<sup>2</sup>Para o caso de classificação de múltiplas classes, a função de ativação Softmax.

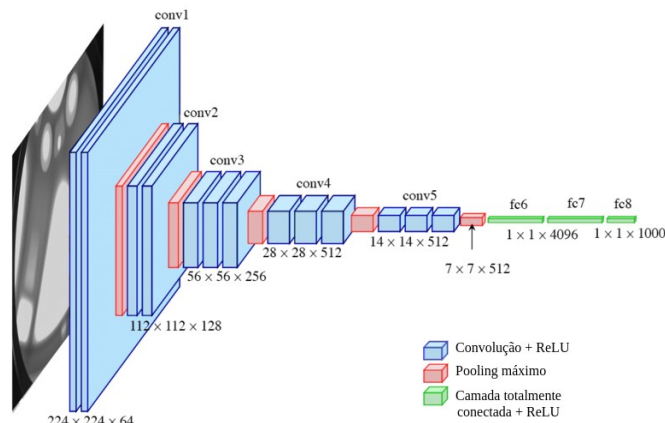


Figura 2.12: Modelo de CNN - VGG16 (Fonte: [26]).

## 2.4.2 Inception

Com o advento de várias arquiteturas de CNN, também surgiu a necessidade de pesquisas em busca de redes convolucionais mais performáticas. Pesquisadores concluíram que essa melhora na performance de CNNs viria juntamente com a melhoria das arquiteturas. Assim, Szegedy *et al.* [27] propuseram o modelo Inception que surgiu com o propósito de ter uma melhor performance quando se comparada à VGG e à AlexNet.

Sabendo que a forma de melhorar a performance de uma rede seria aumentar o tamanho dela, mas que isso também aumentaria a possibilidade de sobreajuste, a Inception surgiu com a proposta de trocar as camadas totalmente conectadas por uma *sparsely connected architecture* (arquitetura escassamente conectada). Assim, a ideia principal deste modelo é encontrar estruturas escassas locais ótimas a partir da combinação de camadas, possibilitando se aproximar ao que é coberto por uma estrutura completa. Além disso, é aplicada uma redução de dimensões utilizando-se de filtros em qualquer lugar que os recursos computacionais requeridos pudessem crescer muito. Por fim, este modelo alcançou uma marca de nove vezes menos parâmetros que o modelo AlexNet e três vezes menos parâmetros que a VGG [28].

## 2.4.3 DenseNet

*Densely Connected Convolutional Network* ou DenseNet, é um modelo proposto por Huang *et al.* [29], cujo objetivo é prover um padrão de conectividade entre as camadas para garantir que o máximo de informações possa ser trocadas entre elas. Assim, todas as camadas desta rede são conectadas entre si e todas as informações recebidas em uma camada são passadas para as subsequentes. O esquemático da Figura 2.13, exemplifica o

funcionamento da mesma. As diversas conexões entre as camadas permitem a chamada reutilização de características, no qual o modelo concatena características aprendidas por diferentes camadas e as utiliza como entrada em camadas posteriores, permitindo uma maior variedade de dados.

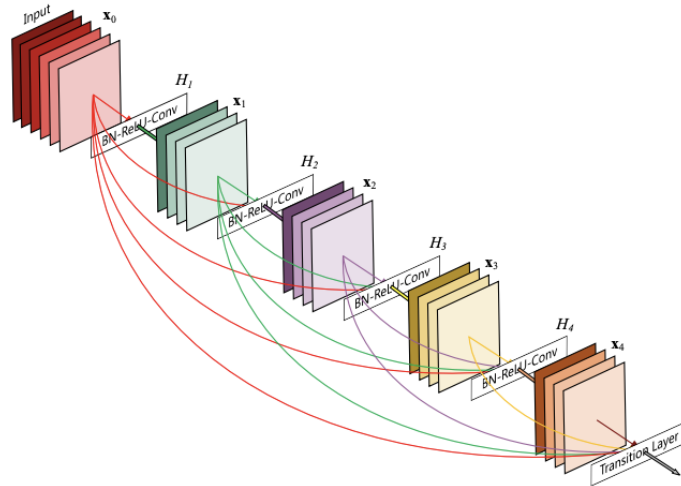


Figura 2.13: Esquemático representando uma DenseNet com 5 camadas (Fonte: [29]).

## 2.4.4 Xception

Modelo apresentado por Chollet [30], o Xception (*Extreme Inception*) surge com a proposta de melhoria do modelo Inception. A principal diferença entre os dois modelos está no que é chamado de camadas de convolução separáveis em profundidade. No modelo de camadas modificado Xception, há a ausência de funções não-lineares e uma alteração na ordem das operações em relação à Inception. E com essas duas mudanças o modelo Xception sobressaiu-se quando comparado aos modelos VGG16, ResNet e Inception no desafio da base de dados ImageNet [25].

## 2.5 Aprendizado por Transferência

*Transfer Learning* é um método de aprendizado de máquina utilizado para melhorar a capacidade preditiva de um modelo a partir da extração de conhecimento de dados específicos para posteriormente permitir a reutilização deste modelo aplicando-o a uma base de dados desconhecida reaproveitando assim o aprendizado obtido [31]. Para o presente trabalho, os modelos de CNN foram treinados na base de dados do Imagenet e, a partir dos pesos obtidos utilizando o *Transfer Learning*, os modelos se tornarão capazes de

classificar imagens em uma base de dados desconhecida, neste caso o conjunto de imagens de pulmão.

## 2.6 Balanceamento de dados

Em tarefas utilizando CNN podem ocorrer problemas durante o treinamento dos modelos. Entre eles está o sobre-ajuste que se refere a um caso onde um determinado modelo se ajusta muito bem a um dado previamente observado<sup>3</sup>, no caso de CNNs, o conjunto de treinamento. Assim, com o intuito de diminuir este problema, previamente à criação dos modelos, são incluídas técnicas que ajudam a reduzir a ocorrência de sobre-ajuste, entre elas o aumento e balanceamento de dados. No aumento de dados, são geradas novas imagens a partir da transformação de amostras iniciais para o conjunto de dados de treinamento [18]. No balanceamento de dados, técnicas são utilizadas com o objetivo de equiparar o número de amostras de uma classe minoritária com amostras de uma classe majoritária. Nesta subseção três delas serão abordadas: *Oversample*, *Undersample* e SMOTE [32].

### *Oversample*

A técnica de sobre-amostragem consiste em balancear o número de amostras em uma classe considerada minoritária com relação ao número de amostras de uma classe majoritária em tarefas de classificação. Esta técnica pode ser aplicada a um conjunto de dados de forma aleatória. Para a técnica de sobre-amostragem, amostras já existentes no conjunto de dados são replicadas, escolhendo amostras aleatoriamente, para aumentar a classe minoritária. Em imagens, é necessário o ajuste das dimensões das mesmas, mantendo-as como vetores unidimensionais para ser realizada a sobre-amostragem. Após a aplicação da técnica aqui citada, os vetores relacionados às imagens são redimensionados novamente, mantendo-os como tridimensionais.

### *Undersample*

A técnica de subamostragem consiste em reduzir as amostras das classes que apresentam um maior número de amostras em relação a uma classe minoritária. Essa técnica remove aleatoriamente amostras das classes majoritárias a fim de balancear o conjunto de dados.

---

<sup>3</sup>O problema do sobre-ajuste está no fato de que o modelo será muito bem ajustado a uma base de dados, porém será ineficaz ao tentar classificar novas amostras.



## SMOTE

*Synthetic Minority Over-sampling Technique* (SMOTE), proposta por Chawla *et al.* [32], é uma técnica de balanceamento de dados baseada no aumento das amostras da classe minoritária para problemas de aprendizado de máquina. Diferente do *Oversample*, esta tem por objetivo não somente o aumento de amostras de uma classe, mas também a criação de amostras sintéticas das mesmas. Assim, utilizando o algoritmo de K-Vizinhos mais próximos, amostras dentro do alcance deste algoritmo são aleatoriamente selecionadas. Por fim, é realizada uma operação onde se obtém a diferença entre estas amostras e seus vizinhos para, assim, multiplicá-las por um número aleatório entre 0 e 1, obtendo portanto, uma nova amostra para a classe minoritária. Em imagens, é necessário o ajuste das dimensões das mesmas, mantendo-as como vetores unidimensionais para ser realizada a sobreamostragem. Após a aplicação do SMOTE, os vetores das imagens originais e das sintéticas são redimensionados novamente, mantendo-os bidimensionais.

## 2.7 Métricas de Avaliação de Desempenho

Com o objetivo de validar o desempenho de arquiteturas de aprendizado de máquina, são feitas avaliações a partir de diversas métricas. Os modelos abordados neste trabalho são relacionados a um problema de classificação de múltiplas classes e as métricas utilizadas surgem a partir da combinação de resultados obtidos na matriz de confusão. São estas as métricas: Acurácia, Revocação, Precisão e F1-score.

### Matriz de Confusão

Também conhecida como Matriz de Erro, a Matriz de Confusão é uma tabela utilizada para a visualização do desempenho de modelos de aprendizado de máquina em tarefas de classificação. A partir dos resultados expressos nela, há a possibilidade de dedução de resultados de um modelo. Sua estrutura é formada por células que são referentes à classe real e a classe prevista, por um classificador, de uma imagem em um problema de classificação, assim como é expresso na tabela da Figura 2.14. Esta Figura exemplifica as matrizes de confusão para um modelo que faz classificação com três classes. É possível observar que existe uma matriz de confusão referente a cada uma das classes, mudando somente as posições onde se encontram os rótulos positivos e negativos.

- Verdadeiro Positivo (VP): Amostra pertencente à classe positiva e classificada corretamente nesta classe;

- Falso positivo (FP): Amostra pertencente à classe negativa e classificada como pertencente à classe positiva;
- Verdadeiro negativo (VN): Amostra pertencente à classe negativa e classificada corretamente nesta classe;
- Falso negativo (FN): Amostra pertencente à classe positiva e classificada como pertencente à classe negativa.

Para modelos com múltiplas classes, é gerada uma matriz de confusão  $n \times n$ , em que  $n$  refere-se ao número de classes de um determinado modelo. Portanto, as métricas aqui apresentadas surgem a partir dos valores herdados desta tabela e são obtidas a partir do cálculo da média dos resultados individuais para cada uma das  $n$  classes.

|      |  | VALOR PREDITO |    |    |
|------|--|---------------|----|----|
| REAL |  | VP            | FN | FN |
|      |  | FP            | VN | VN |
|      |  | FP            | VN | VN |

(a)

|      |  | VALOR PREDITO |    |    |
|------|--|---------------|----|----|
| REAL |  | VN            | FP | VN |
|      |  | FN            | VP | FN |
|      |  | VN            | FP | VN |

(b)

|      |  | VALOR PREDITO |    |    |
|------|--|---------------|----|----|
| REAL |  | VN            | VN | FP |
|      |  | VN            | VN | FP |
|      |  | FN            | FN | VP |

(c)

Figura 2.14: Matriz de Confusão para classificação de múltiplas classes.

## Acurácia

A Acurácia, definida pela equação 2.10, é a métrica que provê a taxa em que as previsões feitas por um modelo foram assertivas. Esta é comumente utilizada em algoritmos de aprendizado de máquina, no entanto, é necessário realizar uma combinação de métricas pois por vezes esta métrica pode levar a resultados equivocados. Em classes desbalanceadas, por exemplo, um modelo pode prever muito bem uma classe enquanto não prevê tão bem outra e ainda assim, a acurácia final deste modelo tende a ser alta.

$$Acurácia : \frac{VP + VN}{VP + FP + VN + FN}. \quad (2.10)$$

## Revocação

A Revocação, também referenciada como Sensibilidade ou Taxa de Verdadeiros Positivos, refere-se à taxa em que um modelo classifica corretamente a classe positiva em relação

ao total de amostras classificadas nesta classe. Esta é definida de acordo com a equação 2.11.

$$\text{Revocação} : \frac{VP}{VP + FN}. \quad (2.11)$$

## **Precisão**

A Precisão, definida pela equação 2.12, refere-se à taxa em que um modelo classifica corretamente a classe positiva, em relação ao total de amostras classificadas como positivas. Esta métrica permite concluir quão confiável é um modelo quando se fizer necessário classificar uma amostra como positiva.

$$\text{Precisão} : \frac{VP}{VP + FP}. \quad (2.12)$$

## **F1-score**

F1-score é uma métrica que avalia a acurácia e o desempenho de um modelo. É computada a partir da média harmônica entre a precisão e a revocação. Por ser baseada em uma média harmônica, caso a precisão ou a revocação tenha um valor baixo, a tendência é que o valor desta métrica também seja. Portanto, esta é uma métrica que pode desempenhar bem em bases de dados desbalanceadas, uma vez que ela permite avaliar, a partir das métricas aqui citadas, a ocorrência de classificações errôneas. A equação 2.13 detalha a definição desta métrica.

$$F_1 : 2 * \frac{\text{precisão} * \text{revocação}}{\text{precisão} + \text{revocação}}. \quad (2.13)$$

# Capítulo 3

## Revisão de literatura

O presente capítulo apresenta uma revisão de literatura a respeito da utilização de diferentes métodos de aprendizado de máquina em tarefas relacionadas a doenças pulmonares. As obras que aqui serão citadas influenciaram diretamente na proposta e elaboração deste trabalho.

Tendo conhecimento de que a melhor forma de aumentar a chance de sobrevivência de um paciente com câncer de pulmão é a detecção e diagnóstico precoce, em 2018, Lakshmanaprabu *et al.* [33] propuseram um modelo que consistia das etapas de pré-processamento, extração e redução de características e classificação das imagens de tomografias computadorizadas propondo o *Optimal Deep Neural Network (ODNN)*. Com a posse de 100 imagens de tumor maligno, tumor benigno e pulmão normal, os autores alcançaram uma acurácia de 96.52%, sensibilidade de 86.45% e especificidade de 92.1% utilizando a k-fold cruzada. Comparando o modelo apresentado com outros classificadores, tais como, ANN, DNN e KNN, os autores concluíram que a arquitetura proposta provê uma melhor classificação de tomografias computadorizadas quando comparada a outras técnicas.

Song *et al.* [34], publicaram um trabalho no qual foram propostas três arquiteturas, sendo uma de DNN, uma de CNN e uma de *stacked autoencoder*, como classificadores. Para aplicar as arquiteturas, os autores utilizaram a base de dados pública do *Lung Image Database Consortium Image Collection (LIDC-IDRI)* que possui um grande número de imagens de pulmão diferenciadas entre três classes: tumor benigno, tumor maligno e pulmão normal. Apesar de contar com um conjunto de imagens extenso, os autores consideraram que seria necessário aplicar o método de aumento de dados e afirmaram que o mesmo melhoraria efetivamente o desempenho das arquiteturas pois obteriam um maior número de amostras no treinamento das arquiteturas. Comparando-se as arquiteturas propostas, a CNN obteve os melhores resultados, tendo uma acurácia de 84.15%.

Utilizando a rede neural convolucional tridimensional proposta, Zhang *et al.* [4] propuseram um método para detecção de câncer de pulmão no qual foram utilizadas três bases de dados com diferentes propósitos. Para o treinamento prévio do modelo de CNN foi utilizada a base de dados de tomografias computadorizadas disponibilizada no desafio LUNA16 que ocorreu em 2016. Para o treinamento e validação do modelo proposto foram utilizadas tomografias computadorizadas do tórax disponibilizadas aos autores por um hospital local. E por fim, com o objetivo de testar o algoritmo, foram utilizados dados de 50 pacientes que passaram por cirurgia na área do pulmão.

Neste método Zhang *et al.* [4] utilizaram uma abordagem contando com dois tipos de imagens do pulmão, em que foram analisadas imagens de tumor maligno diagnosticado por oncologistas e imagens de outras doenças relacionadas ao pulmão. Os autores dividiram o método proposto em três módulos. No primeiro módulo foi feito um pré-processamento para serem isoladas as regiões em que continham somente o tecido pulmonar. Os dois módulos seguintes foram realizados com objetivo de detecção e validação do método. Por fim, utilizando-se validação cruzada, os autores obtiveram o melhor resultado com 6 *folds*, no qual obteve-se 83.7% de acurácia, 83.4% de sensibilidade e 83.9% de especificidade.

Utilizando redes neurais residuais profundas pré-treinadas e *transfer learning* (transferência de aprendizado), Schwyzer *et al.* [35] propuseram um método para detecção e classificação binária automática de tumor maligno no pulmão. Mesmo sabendo que a segmentação do pulmão e a combinação de diferentes tipos de imagens melhoraria a performance da rede neural residual, os autores optaram, inicialmente, pela utilização somente da rede neural residual proposta pré-treinada sem técnicas adicionais. A partir da validação cruzada com 10 *folds*, obteve-se uma acurácia de 97.1%. Portanto, mesmo sendo experimentos iniciais, a rede proposta demonstrou ser efetiva na tarefa de detecção de câncer de pulmão.

Shakeel *et al.* [36] apresentaram um método eficiente para a detecção de câncer de pulmão. Diferentemente das técnicas até então abordadas, os autores utilizaram técnicas para inicialmente fazer a limpeza das imagens a partir da remoção de *pixels* inconsistentes e remoção de ruído, para posteriormente aplicar estas imagens em uma rede neural. O método sequencialmente passou por três etapas: primeiramente foi feito a redução de ruído das imagens utilizando-se a equalização de histogramas, em seguida foi feita a segmentação do pulmão com o objetivo de focalizar o esforço dos algoritmos nas regiões de interesse, e por fim, foi realizada a classificação, em que usando uma DITNN, foi obtido o resultado da detecção, sendo câncer de pulmão ou pulmão normal.

Xiao *et al.* [37] propuseram uma metodologia que, utilizando as arquiteturas UNet e ResNet, possibilitou a criação de uma arquitetura para segmentação de câncer de pulmão a partir de tomografias computadorizadas: a 3D-Res2UNet. A partir de 1186 amostras

de câncer de pulmão, os autores alcançaram 95.3% no coeficiente Dice. Verificou-se um alto número de falsos positivos, porém ainda assim essa arquitetura mostrou-se eficaz na segmentação de pequenos nódulos no pulmão por apresentar um coeficiente dice e revocação maior que outras arquiteturas comparadas.

Com o objetivo de distinguir câncer de não-câncer em imagens de pulmão, Jung *et al.* [38] apresentaram uma CNN tridimensional para o problema de classificação. Utilizando a base de dados da competição LUNA16, os autores observaram que o número de imagens da classe de não-câncer era cerca de 480 vezes maior que o número de imagens da classe de câncer de pulmão e portanto estava ocorrendo um sobre-ajuste no modelo proposto. Assim, foi necessário um pré-processamento para reduzir o número de amostras de pulmões sem câncer. Por fim, foi aplicado o método de ensemble para agregar o resultado do modelo proposto com o modelo ESB-ALL treinados para assim melhorar o desempenho final. Baseando-se nas métricas utilizadas na competição LUNA16, os autores utilizaram a CPM, que verifica a média da sensibilidade em alguns pontos pré-definidos. Com ela, os autores obtiveram uma pontuação de 91%. Assim, a partir dos resultados obtidos autores consideraram o método como sendo efetivo na tarefa de distinção entre câncer de pulmão e não-câncer em imagens do pulmão.

Togaçar *et al.* [5] desenvolveram um método de classificação de tomografias computadorizadas utilizando o transfer learning a partir das arquiteturas LeNet, AlexNet e VGG16. Tendo posse de uma pequena base de dados contendo 100 imagens e com intuito de prevenir um sobre-ajuste, foram necessárias técnicas de aumento de imagem, tais como, rotação, amplificação, mudança na altura e largura da imagem. Aplicando as características extraídas aos classificadores de aprendizado de máquina, os autores puderam concluir sobre o desempenho dos modelos propostos. A partir da validação cruzada, utilizando-se 10 *folds*, tendo a arquitetura AlexNet juntamente ao método KNN como classificador, obtiveram os melhores resultados, no qual alcançou-se uma acurácia de 98.7%.

Jakimovski *et al.* [39] propuseram um método que, utilizando uma CNN profunda, objetivava ajudar na detecção precoce e facilitar o tratamento do câncer no pulmão. Contando com uma base de dados com 6080 imagens de tomografias do pulmão disponibilizada pela Universidade de Carolina do Sul, os autores iniciaram o modelo com um pré-processamento das imagens para que pudessem ser selecionadas somente as regiões de interesse e em seguida, foram realizados os testes na CNN dupla. Com o objetivo de validar o desempenho da arquitetura proposta, a mesma base de dados foi aplicada a uma DCNN sem as camadas convolucionais duplicadas. Por fim, como esperado pelos autores, o método proposto sobressaiu em relação à rede testada, em que obteve-se 99% de acurácia, sendo 12% maior que a acurácia da arquitetura sem as camadas convolucionais duplicadas.

Khan *et al.* [40], utilizando a arquitetura pré-treinada VGG19, implementaram um modelo que considera etapas de detecção, segmentação e classificação de câncer de pulmão a partir de imagens de tomografia. Para melhores resultados foi necessário realizar o pré-processamento, redimensionando as imagens que inicialmente eram bidimensionais e, após essa etapa, o modelo foi dividido em duas partes: a primeira relacionada à extração de características utilizando a rede pré-treinada e a segunda relacionada à segmentação e extração das características essenciais. Por fim, concatenando o resultado dos dois processos, foi realizada a classificação dos câncer de pulmão. Como resultado os autores obtiveram uma acurácia de aproximadamente 99% e concluíram que o modelo proposto obteve melhores resultados se comparado aos modelos VGG16, ResNet18, ResNet50 e AlexNet.

Polat *et al.* [6] pensando em formas de melhorar a eficiência de modelos de redes neurais em tarefas de classificação de doenças pulmonares propuseram um método que, utilizando CNN tridimensionais, proporcionou resultados considerados ótimos pelos autores [6]. O modelo em questão utilizou uma arquitetura híbrida, onde os autores implementaram uma rede neural convolucional para a tarefa de extração de características e por fim utilizaram o classificador SVM. Utilizando a base de dados contendo 2101 imagens de tomografia do pulmão, disponibilizada na competição que ocorreu em 2017 pelo Kaggle, os autores compararam o método proposto com arquiteturas tais como a GoogleNet e a AlexNet, que foram consideradas na literatura como sendo eficientes. A acurácia do modelo proposto foi de 91.8%, cerca de 4% maior que a mesma métrica nas arquiteturas comparadas.

A Tabela 3.1 sintetiza os trabalhos abordados nessa revisão de literatura, mostrando os problemas tratados, as técnicas utilizadas e os resultados alcançados.

Tabela 3.1: Resumo dos trabalhos abordados neste capítulo.

| <b>Autores</b>                    | <b>Problemas</b>                             | <b>Técnicas</b>                                                      | <b>Resultados</b>                     |
|-----------------------------------|----------------------------------------------|----------------------------------------------------------------------|---------------------------------------|
| Lakshmanaprabu <i>et al.</i> [33] | Classificação de câncer de pulmão            | DNN ótima e transferência de aprendizado                             | Acurácia: 96.52%<br>Revocação: 86.45% |
| Song <i>et al.</i> [34]           | Detecção e classificação de câncer de pulmão | Stacked Autoencoder e CNN                                            | Acurácia: 84.15%<br>Revocação: 83.96% |
| Zhang <i>et al.</i> [4]           | Detecção de câncer de pulmão                 | CNN                                                                  | Acurácia: 83.7%<br>Revocação: 83.4%   |
| Schwyzler <i>et al.</i> [35]      | Detecção e classificação de câncer de pulmão | DNN residual                                                         | Acurácia: 97.1%<br>Revocação: 95.9%   |
| Shakeel <i>et al.</i> [36]        | Segmentação e detecção de câncer de pulmão.  | DITNN                                                                | Acurácia: 94.5%<br>Revocação: 96.8%   |
| Xiao <i>et al.</i> [37]           | Segmentação de câncer de pulmão              | 3D-UNet e Res2Net                                                    | Dice: 95.3%<br>Revocação: 99.1%       |
| Jung <i>et al.</i> [38]           | Classificação de câncer de pulmão            | CNN tridimensional e ensemble de CNN                                 | CPM: 91%<br>Revocação: 93.3%          |
| Togaçar <i>et al.</i> [5]         | Classificação de câncer de pulmão            | CNNs LeNet, AlexNet e VGG-16 e transferência de aprendizado          | Acurácia: 98.74%<br>Revocação: 98.35% |
| Jakimovski <i>et al.</i> [39]     | Detecção de câncer no pulmão                 | CNN dupla                                                            | Acurácia: 99.62%<br>Revocação: 99.91% |
| Khan <i>et al.</i> [40]           | Detecção e classificação de câncer de pulmão | CNN VGG-SegNet e métodos de aprendizado de máquina: GLCM, LBP e PHOG | Acurácia: 99.68%<br>Revocação: 83.73% |
| Polat <i>et al.</i> [6]           | Classificação de câncer de pulmão            | CNNs AlexNet e GoogleNet                                             | Acurácia: 91.81%<br>Revocação: 88.53% |



# Capítulo 4

## Metodologia

O presente trabalho visa analisar o comportamento de diferentes modelos de CNNs. Além disso, estes modelos serão combinados com classificadores tradicionais de aprendizado de máquina, permitindo assim uma análise do comportamento de modelos híbridos em conjuntos de dados com múltiplas classes a respeito de imagens de tomografias computadorizadas do pulmão. Uma base principal será utilizada para o desenvolvimento da metodologia, em que a mesma disponibiliza imagens de pulmão normal, pulmão com tumor maligno e pulmão com tumor benigno, permitindo assim a aplicação de classificadores para o problema de múltiplas classes. A Figura 4.1 sintetiza a metodologia proposta no presente trabalho.

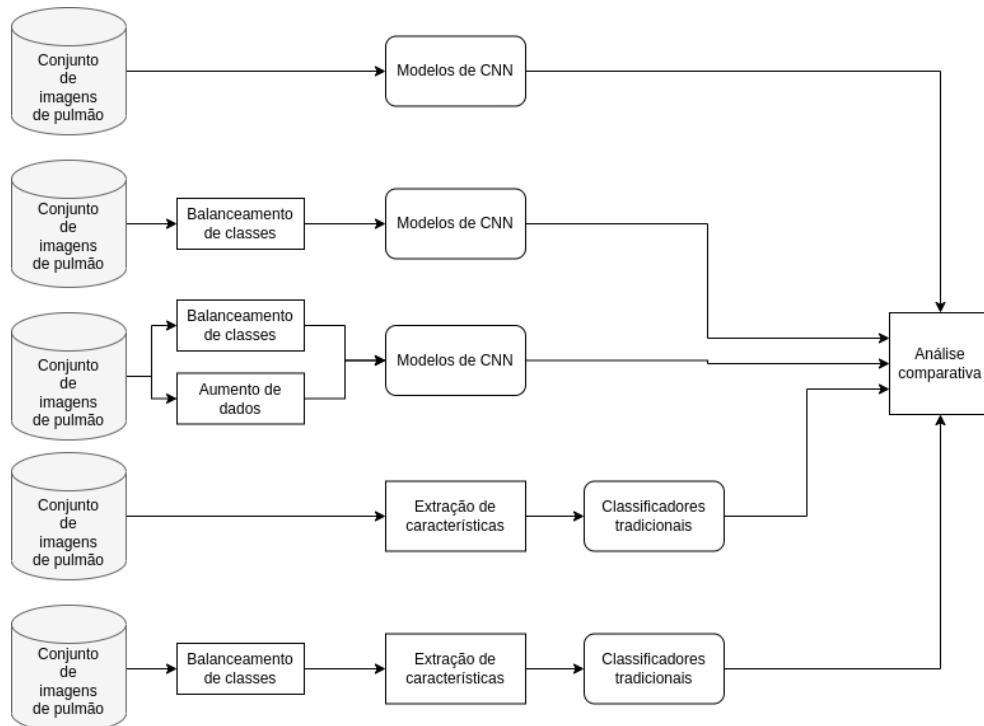


Figura 4.1: Fluxograma da metodologia a ser utilizada.

Portanto, neste capítulo serão detalhadas todas as etapas a serem executadas para a aplicação da metodologia proposta. Na Seção 4.1 será detalhada a base de dados que será utilizada nos modelos que serão apresentados. Em seguida, na Seção 4.2 é detalhada a etapa de pré-processamento das imagens e todos os testes a serem realizados para, por fim, determinar os métodos que melhor se ajustarão ao problema de classificação de múltiplas classes. Por fim, na Seção 4.3 serão abordados os modelos de CNN e as estratégias a serem utilizadas para a definição dos classificadores, além dos procedimentos que serão realizados em cada um dos experimentos propostos.

## 4.1 Base de dados

Neste trabalho a base de dados de imagens de tomografias computadorizadas do pulmão será utilizada objetivando a detecção de tumores em imagens de tomografias de pulmão. Contando com três tipos de imagens de pulmão, *Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases lung cancer dataset* (IQ-OTHNCCD) é uma base de dados livre.

Disponibilizada por dois hospitais do Iraque, sendo eles o Hospital Universitário de Oncologia e o Centro Nacional de Doenças Cancerígenas, a base de dados IQ-OTHNCCD [41] é composta de imagens coletadas em um período de três meses no ano de 2019. Nesta base de dados há tomografias computadorizadas de pacientes diagnosticados com câncer em diferentes estágios, assim como de indivíduos saudáveis.

O conjunto de dados contém 1097 imagens, sendo elas obtidas a partir de 80 a 200 fatias de tomografias computadorizadas de 40 casos diagnosticados como tumor maligno, 15 casos diagnosticados como tumor benigno e 55 casos de pulmão saudável. A imagem da Figura 4.2 exemplifica as amostras encontradas nessa base de dados. Do total de imagens deste conjunto, 120 são relacionadas a pulmões com tumor benigno, 561 são imagens de pulmões com tumor maligno e 416 imagens de pulmões saudáveis. A classe de pulmões com tumor benigno tem 20% do número total de amostras da classe de tumor maligno. Portanto, analisando a distribuição entre as classes fica notória a necessidade de utilização de métodos de balanceamento de dados durante a implementação dos modelos de CNN.

Por fim, a partir de amostras aleatórias e estratificadas das imagens, neste trabalho, esta base de dados será separada em três conjuntos distintos separados da seguinte forma: 70% das imagens para o conjunto de treinamento, 10% das imagens para o conjunto de validação e 20% das imagens para o conjunto de testes.

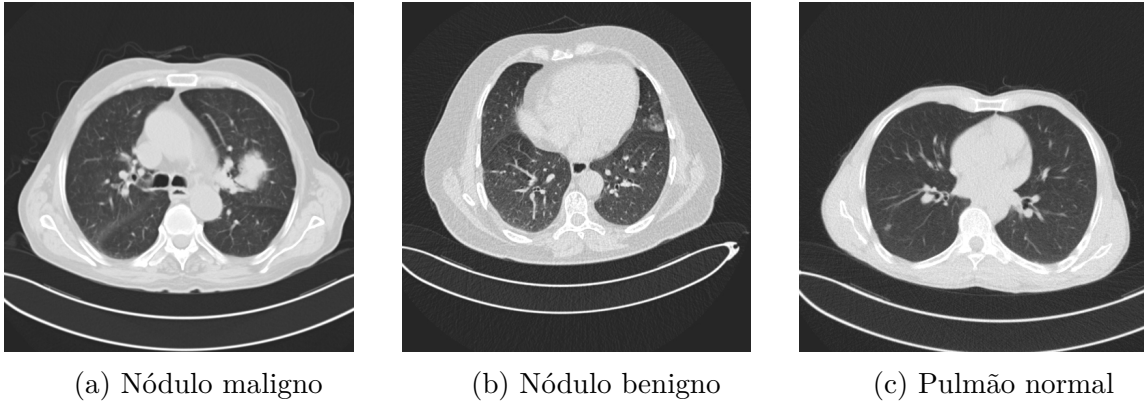


Figura 4.2: Exemplo de imagens das três classes encontradas na base de dados IQ-OTHNCCD.

Para o procedimento de aquisição das imagens foi utilizado o scanner SOMATOM da empresa Siemens em que foi construída uma base de dados no formato DICOM. Os dados das imagens analisadas por oncologistas e radiologistas dos hospitais acima citados variam em gênero, idade e localização e foram disponibilizadas ao público no formato JPEG, com dimensões de  $512 \times 512$  *pixels*. Em tomografias computadorizadas, os *pixels* obtidos de uma imagem são exibidos em relação à radiodensidade relativa e são obtidos de acordo com a escala Hounsfield. Para esta base de dados, durante a aquisição das imagens foram utilizados de 50 a 600 Unidades Hounsfield (HU).

A utilização de CNNs previamente treinadas requer que a base de dados esteja em dimensões pré-definidas. Assim, se faz necessário o redimensionamento das imagens da base de dados do IQ-OTHNCCD, visto que suas dimensões são de  $512 \times 512$  assim como citado anteriormente. Portanto, previamente às etapas de pré-processamento e treinamento dos modelos, é realizado este redimensionamento. De acordo com cada modelo de CNN a ser utilizado, é aplicada uma função de redimensionamento utilizando o método de interpolação bilinear. Nesta função, cada uma das imagens das bases de dados utilizadas é redimensionada e tem suas dimensões definidas de acordo com a documentação de cada modelo de CNN, como segue abaixo:

- $224 \times 224 \times 3$  para os modelos VGG19 e Densenet201;
- $299 \times 299 \times 3$  para os modelos Inception e Xception.

## 4.2 Etapa de pré-processamento de imagens

Se tratando de problema de classificação de imagens utilizando ANNs, algumas etapas devem ser executadas previamente à etapa de treinamento de modelos. Essa necessidade

surge da não adequação das bases de dados aos pré-requisitos necessários para a execução de modelos específicos, por exemplo, CNNs.

A utilização de CNNs neste trabalho motivou a necessidade de incluir, na etapa de pré-processamento das imagens, uma etapa específica de pré-processamento para cada um dos modelos de CNN utilizados. Além disso, com o objetivo de obter um número de amostras balanceado para todas as classes da base de dados a ser utilizada, se fez necessária a utilização dos métodos de aumento de dados, citados na Subseção 2.6. Assim, nesta seção será apresentada a etapa de pré-processamento de imagens de tomografias e serão detalhados os procedimentos aqui citados que serão realizados para obter imagens padronizadas.

## Pré-processamentos definidos de acordo com os modelos de CNN

Para cada modelo de CNN a ser utilizado é necessário que um pré-processamento customizado seja aplicado para o devido ajuste das bases de dados ao modelo a ser utilizado, de acordo com sua documentação<sup>1</sup>.

Para o modelo VGG19 as imagens que originalmente estavam com os espaços de cores RGB são convertidas a partir da reordenação destes espaços para o canal de cor BGR. Além disso, cada canal de cor destas imagens é zero centrado com respeito ao conjunto de dados ImageNet [42], ou seja, a média do canal de cor tende ao valor zero, variando os *pixels* entre valores positivos e negativos<sup>2</sup>. No modelo DenseNet os valores dos *pixels* de cada imagem são convertidos dividindo-os por 255 para obter uma variação entre 0 e 1. Além disso, para cada canal é aplicada a normalização *Zscore*<sup>3</sup>. Nos modelos Inception e Xception o procedimento de pré-processamento se mantém igual para ambos. Os *pixels* das imagens de entrada são convertidos para valores entre -1 e 1<sup>4</sup>.

## Balanceamento de Dados

Como mencionado anteriormente na Subseção 4.1, na base de dados a ser utilizada há um problema no que tange ao desbalanceamento nas classes. É notável que o conjunto de imagens referente a pulmão com tumor benigno contém cerca de 76.74% de imagens a menos que o conjunto de imagens de tumor maligno, o que potencialmente poderá causar um sobre-ajuste dos modelos para a classe majoritária. Portanto, nestes casos como o citado, se faz necessária a utilização de métodos que diminuam o desbalanceamento dessas classes, assim como foi exposto na Subseção 2.6. Neste trabalho as técnicas de balanceamento de

---

<sup>1</sup><https://keras.io/api/applications/>.

<sup>2</sup><https://keras.io/api/applications/vgg>.

<sup>3</sup><https://keras.io/api/applications/densenet>.

<sup>4</sup><https://keras.io/api/applications/xception/>.

dados *Oversampling* e SMOTE [32] serão utilizadas para redução do desbalanceamento de classes. As Figuras 4.3 e 4.4 exemplificam a geração das novas amostras de imagens quando aplicadas essas técnicas. Nota-se que a técnica de *Oversampling* poderia levar os modelos a um sobre-ajuste durante o processo de treinamento, uma vez que esta técnica somente gera amostras iguais às já existentes.

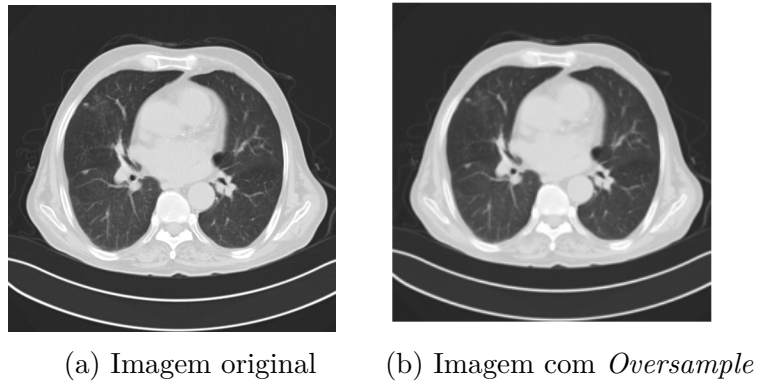


Figura 4.3: Exemplo de imagem gerada pela técnica de *Oversample*.

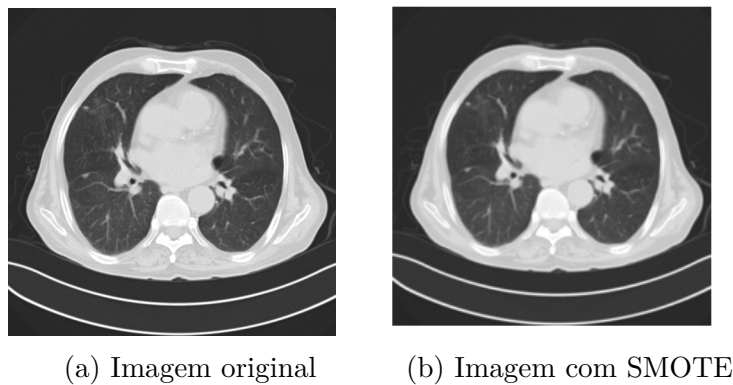


Figura 4.4: Exemplo de imagem gerada pela técnica SMOTE.

## Aumento de Dados

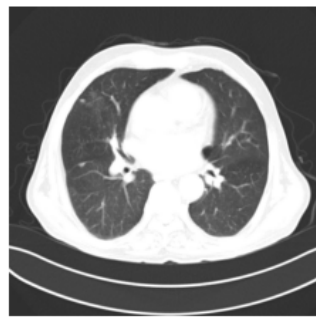
Além da aplicação de técnicas para evitar o desbalanceamento de classes, também será aplicado o aumento de dados com o objetivo de diversificar as amostras durante o treinamento e classificação dos modelos utilizados. A técnica de aumento de dados utiliza-se da biblioteca *Keras*<sup>5</sup> para, durante o treinamento dos modelos, gerar novas amostras, de forma aleatória, das imagens do conjunto de treinamento. Para o presente trabalho, quatro camadas de aumento de dados serão consideradas: brilho aleatório, rotação aleatória, inversão horizontal e zoom aleatório. As imagens da Figura 4.5 foram geradas a partir

---

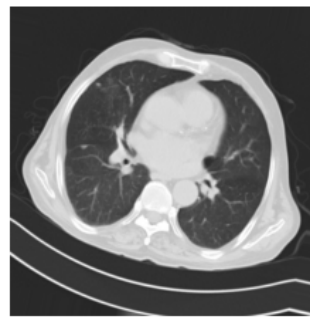
<sup>5</sup><https://keras.io>.

da imagem da Figura 4.3 e exemplificam cada uma das camadas a serem utilizadas. Por fim, com o objetivo de obter imagens diversificadas durante o treinamento dos modelos e evitando a perda de informações destas imagens com o uso das camadas aqui citadas, são utilizados valores obtidos empiricamente. Abaixo são abordadas as quatro camadas juntamente com os valores de entrada para cada uma delas.

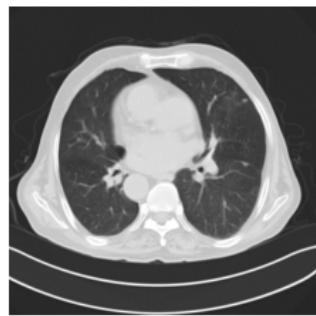
- Brilho aleatório: ajusta aleatoriamente o brilho das imagens com um valor entre 20% e 50% do brilho inicial;
- Rotação aleatória: rotaciona aleatoriamente a imagem em até 40% da sua posição inicial;
- Inversão horizontal: inverte a imagem horizontalmente;
- Ampliação aleatória: amplia aleatoriamente as imagens entre 20% e 30% do seu valor inicial.



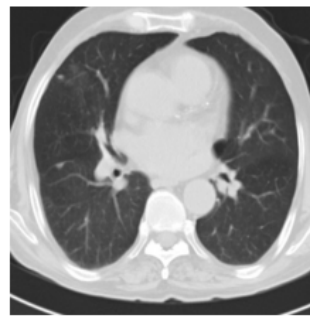
(a) Brilho aleatório



(b) Rotação aleatória



(c) Inversão horizontal



(d) Ampliação aleatória

Figura 4.5: Exemplo de imagens geradas a partir da utilização de métodos de aumento de dados.

## Transferência de Aprendizado

No presente trabalho, modelos híbridos são criados com o propósito de avaliar os classificadores tradicionais de aprendizado de máquina citados na Subseção 2.2.2, quando

utilizados juntamente a CNNs. Para isso se faz necessária a utilização de uma técnica de Transferência de Aprendizado para realizar a extração de características das imagens.

Como explicado na Seção 2.3, em CNN três principais camadas são utilizadas nestes modelos e são elas: Camada Convolutiva, Camada de Pooling e Camada Totalmente Conectada. Por tanto, com o objetivo de extrair características fundamentais das imagens para possibilitar a utilização de um classificador externo à Camada Totalmente Conectada, a definição do modelo de extração de características ocorre a partir da remoção da última camada da CNN e congelamento de todos os pesos das camadas inferiores. Assim, as imagens serão treinadas em uma rede com pesos já definidos pela base de dados ImageNet e, com o conhecimento previamente obtido a partir desta base, os modelos são capazes de prover em suas saídas somente as características fundamentais das imagens de entrada. Por fim, tendo posse deste conjunto de características, se torna possível a utilização de classificadores tradicionais, uma vez que somente será necessário redimensionar o conjunto de características tornando-o unidimensional.

A Figura 4.6 exemplifica um modelo arbitrário de CNN pré-treinada com pesos da base de dados ImageNet e a Figura 4.7 demonstra como o modelo pré-treinado deve ser estruturado para possibilitar a criação de um modelo híbrido com classificador tradicional de aprendizado de máquina. Em suma, o modelo híbrido utiliza-se de CNNs para a realização da extração de características e por fim o vetor das características obtidas é utilizado como valores de entrada para os classificadores citados na Subseção 2.2.2.

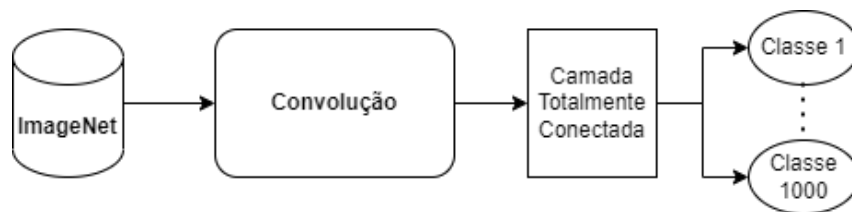


Figura 4.6: Modelo de CNN pré-treinada com a base de dados ImageNet.

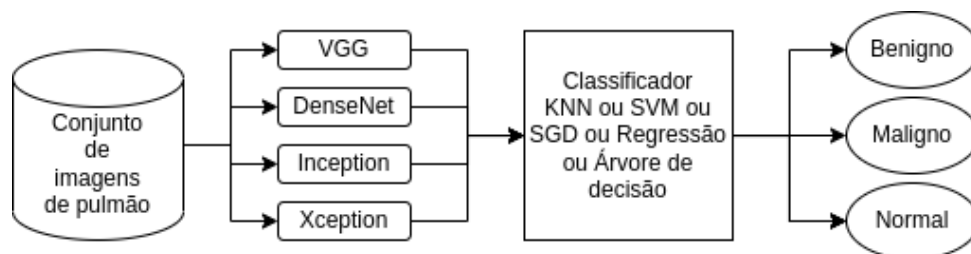


Figura 4.7: Modelo híbrido com pesos congelados na fase convolutiva, realizando a extração de características e posteriormente a classificação.

## 4.3 Experimentos propostos

Na tarefa de classificação de imagens de tomografias computadorizadas de pulmão são utilizados quatro modelos de CNN, sendo eles: VGG [24], Inception [27], DenseNet [29] e Xception [30]. Junto a estes, classificadores de aprendizado de máquina serão combinados aos modelos para permitir a criação de modelos híbridos, focando em comparar o desempenho dos mesmos. O presente trabalho conta com dois principais experimentos, onde os mesmos têm por objetivo levantar dados comparativos sobre modelos híbridos utilizando os modelos de CNN aqui citadas. O primeiro experimento refere-se à análise da tarefa de classificação utilizando o modelo de CNN, sem a inclusão de algoritmos de aprendizado de máquina. O segundo experimento objetiva analisar o comportamento dos modelos de CNN quando utilizados juntamente a algoritmos de aprendizado de máquina.

Como forma de avaliar o desempenho dos modelos a serem utilizados nos experimentos, as métricas de avaliação de desempenho citadas na Seção 2.7 serão utilizadas, são estas: acurácia, objetivando validar classificações assertivas, revocação e precisão, com o objetivo de verificar a taxa de positivos e por fim, F1-score, que tem por objetivo generalizar o desempenho dos modelos a partir da combinação da precisão e revocação.

Portanto, nesta seção serão abordadas todas as técnicas, procedimentos e experimentos realizados para possibilitar o sucesso nos objetivos propostos.

### 4.3.1 Modelo de CNN completa

Por se tratar de uma tarefa de classificação de múltiplas classes, ao construir os modelos é essencial a customização de alguns parâmetros alterando seus valores iniciais, para se adequar à tarefa a ser realizada. Para a utilização dos modelos de CNN abordados na Seção 2.4 não será alterada a estrutura dos mesmos, utilizando-se os valores pré-estabelecidos pela biblioteca *Keras*<sup>6</sup>.

Como citado anteriormente, para construir modelos utilizando-se dos pesos dos modelos previamente treinados, se faz necessária a remoção da camada totalmente conectada e incluir uma nova com um classificador específico para o problema abordado. Portanto, ao fim da última camada da parte convolucional dos modelos é incluída uma camada de Pooling por média (Subseção 2.3.3) com o objetivo de mapear as características extraídas e retornar um conjunto unidimensional. Assim, na camada totalmente conectada é definida como função de ativação a Softmax que mostra-se eficaz na classificação tanto de classes binárias quanto de múltiplas classes. A Figura 4.8 exemplifica o fluxo a ser realizado neste experimento.

---

<sup>6</sup><https://keras.io/api/applications/>.



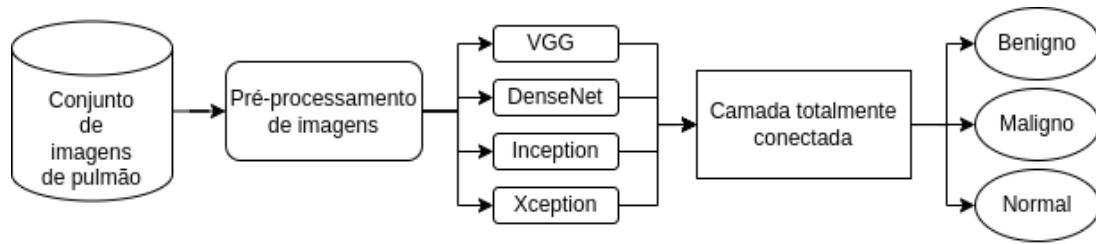


Figura 4.8: Fluxograma relacionado ao primeiro experimento.

No procedimento experimental é utilizado o otimizador Adam com taxa de aprendizado pré-definida de 0.05 e uma função de perda alternativa às comumente utilizadas nos modelos de classificação: a *Sigmoid Focal Cross Entropy*. Introduzida por Tsung *et al.* [43], esta é uma função de perda focada em problemas com classes desbalanceadas. Focada nas classes com menores quantidades de amostras, a função de perda aqui citada diminui os pesos das classes majoritárias e, uma vez que o problema aqui proposto conta com uma base de dados desbalanceada, foram feitos experimentos com esta função com o objetivo de verificar se a mesma se sobressairia em relação à comumente utilizada na literatura.

Assim, o primeiro experimento objetiva a análise do comportamento dos modelos quando aplicado a eles a base de dados citada da Seção 4.1. Como parte do experimento, alguns testes se fazem necessários. Primeiramente, com o objetivo de validar o desempenho do modelo quando utilizado o *Oversampling* para balanceamento de classes, os modelos são treinados utilizando as duas técnicas de balanceamento de classe citadas na Seção de pré-processamento [4.2]. Também com o objetivo de validar a utilização da função de perda *Focal Loss* contrariamente à função *Categorical Cross-entropy*, é realizado o treinamento dos modelos e, com as devidas comparações, será possível concluir acerca da melhor função de perda para este experimento no Capítulo 5.

As arquiteturas de CNN foram configuradas para serem treinadas por 100 épocas, porém passado um número específico destas épocas notou-se, em experimentos preliminares, que não haviam melhoras substanciais nas métricas aqui utilizadas. Em vista disso, para melhorar o desempenho computacional do modelo, evitando o treinamento por 100 épocas, e para prevenir a ocorrência de sobreajuste do modelo, será utilizado o *callback*<sup>7</sup> de parada precoce. Assim, a cada época é feita uma verificação para observar se houveram mudanças nas métricas escolhidas e caso negativo, encerra a execução do treinamento do modelo. Desta forma, utilizando 10 épocas<sup>8</sup> como parâmetro de melhora, nota-se que, em experimentos preliminares, os modelos convergem bem antes das 100 épocas inicialmente definidas.

<sup>7</sup>[https://keras.io/api/callbacks/early\\_stopping/](https://keras.io/api/callbacks/early_stopping/)

<sup>8</sup>10% do valor total de épocas estabelecido.

### 4.3.2 Modelos híbridos

Seguindo os experimentos propostos, nesta etapa, modelos híbridos são criados, sendo cinco modelos para cada um dos quatro modelos de CNN utilizadas neste trabalho: VGG19, DenseNet, Inception e Xception, totalizando assim, vinte modelos híbridos. A Figura 4.9 exemplifica o fluxograma que é seguido até a classificação das imagens. Ao fim da última camada convolucional são retornadas as principais características das imagens em um formato bidimensional. Assim, para aplicar esses dados extraídos da imagem a um classificador externo, se faz necessário ajustar as dimensões destes dados de forma a se ter dados unidimensionais. Para os classificadores KNN, SVM, SGD, Árvore de Decisão e Regressão Logística, valores precisam ser determinados a fim de se obter uma melhor classificação, estes podem ser observados na Tabela 4.1. Os valores customizados dos parâmetros foram selecionados de maneira empírica, onde, a partir de diversas execuções dos modelos, foram sendo selecionados os que melhor se adequaram ao objetivo proposto. Além disso, os parâmetros que não foram citados aqui mantiveram os seus valores pré-definidos, de acordo com a biblioteca *Scikit-learn*<sup>9</sup>.

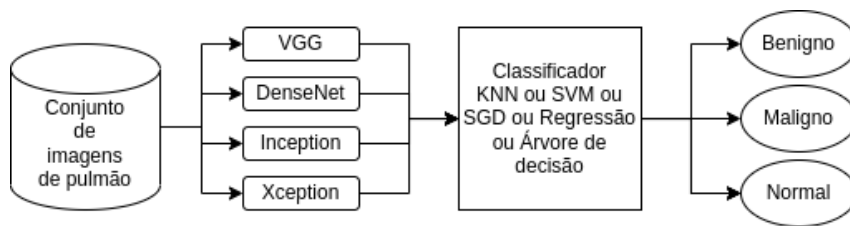


Figura 4.9: Esquemático do modelo híbrido utilizado no segundo experimento.

Tabela 4.1: Tabela com parâmetros customizados utilizados nos classificadores.

| Classificador       | Configuração                      |
|---------------------|-----------------------------------|
| Regressão Logística | Otimizador: LBFGS                 |
|                     | Iterações máximas: 500            |
| KNN                 | Número de vizinhos próximos: 5    |
| SVM                 | Estimativa probabilística         |
|                     | Número de folds: 5                |
| SGD                 | Função de perda: Huber modificado |
|                     | Iterações máximas: 1000           |
|                     | Taxa de aprendizado otimizada     |
| Árvore de Decisão   | -                                 |

<sup>9</sup><https://scikit-learn.org/stable/>.

O experimento utilizando modelos híbridos inicialmente objetiva verificar o comportamento dos modelos quando treinados sem aumento de dados e posteriormente com o aumento dos dados com foco em balanceamento das classes, objetivando evidenciar o bom desempenho no conjunto de dados de treinamento e possibilitar maximizar os resultados obtidos. Neste experimento será verificada a utilização dos métodos SMOTE e *Over-sampling* [32] a partir de sub-experimentos utilizando os modelos híbridos citados nesta Seção.

# Capítulo 5

## Resultados

No presente capítulo são apresentados todos os resultados obtidos a partir dos experimentos descritos no Capítulo 4 e junto a eles as devidas comparações entre os modelos de CNN e os métodos de aumento e balanceamento de dados quando utilizados na base de dados detalhada na Subseção 4.1. Inicialmente são detalhadas as ferramentas utilizadas e em seguida são apresentados os resultados obtidos a partir da aplicação dos modelos baseados em CNN. E por fim são expostos os resultados dos experimentos com modelos híbridos.

### 5.1 Ferramentas utilizadas

Para o desenvolvimento e reprodução dos experimentos que aqui serão apresentados foi utilizada a linguagem de programação Python, que traz consigo uma grande quantidade de métodos, além de bibliotecas que tornaram a construção dos modelos mais simples. Entre estas bibliotecas, as utilizadas foram: Tensorflow, Keras, Scikit-learn, Numpy, Matplotlib, Imblearn e Tabulate.

Objetivando a redução no tempo de processamento dos modelos apresentados nos experimentos, foi utilizado o Google Colaboratory<sup>1</sup>, uma ferramenta que permite a execução de códigos Python a partir do navegador da web. Este serviço provê máquinas virtuais com GPUs, permitindo que casos como o treinamento de CNN, atinjam melhores desempenhos. Os experimentos foram realizados em uma máquina virtual com CPU Intel(R) Xeon(R) CPU @ 2.00GHz, GPU Tesla T4 16 GB e 12 GB de memória RAM.

---

<sup>1</sup><https://colab.research.google.com/>.

## 5.2 Experimento I - CNN completa

O primeiro experimento refere-se aos modelos em que foram utilizadas CNNs previamente treinadas, se alterando somente a camada totalmente conectada para assim poder adaptar estes modelos à tarefa de classificação de imagens de pulmão, como citado na Subseção 4.3.1. Neste experimento foram considerados experimentos específicos para verificar o comportamento dos modelos de CNN quando utilizada a base de dados desbalanceada e balanceada posteriormente.

### 5.2.1 Modelo sem aumento de dados

Como destacado na Seção 4.3, este experimento inicial objetivou a verificação de uma função de perda considerada na literatura como válida para tarefas utilizando dados desbalanceados: a *Sigmoid Focal Cross Entropy*. Assim, foram realizadas duas execuções dos modelos de CNN, a primeira utilizando a função aqui citada e a execução seguinte utilizando a função *Categorical Cross Entropy*, mantendo-se apenas os pré-processamentos de imagens necessários, para poder validar o comportamento desta função em classes desbalanceadas.

Para a realização desta verificação, foram utilizadas 100 épocas de treinamento e o *callback* parada precoce <sup>2</sup> com o objetivo de evitar o sobreajuste dos modelos. Na Tabela 5.1 são demonstrados os resultados obtidos no conjunto de testes quando executados os quatro modelos: VGG, DenseNet, Inception e Xception. Em negrito encontram-se as melhores métricas alcançadas.

Assim, é evidente que para a função de perda Focal CE o modelo Xception melhor desempenhou, enquanto o Inception foi o que teve os menores resultados. Por outro lado, na função de perda Categorical CE o modelo VGG teve os melhores resultados e diferentemente da função de perda Focal CE o modelo Xception teve o pior desempenho. Por fim, pode-se concluir que analisando os resultados o modelo VGG utilizando a função de perda Categorical CE obteve o melhor resultado quando treinado com base de dados desbalanceada.

É notável que a função de perda *Categorical Cross Entropy* apresentou os melhores resultados quando comparada à função de perda *Focal Cross Entropy*. As Imagens das Figuras 5.1 e 5.2, referentes aos resultados obtidos nas execuções utilizando as funções *Focal CE* e *Categorical CE* respectivamente, demonstram os resultados da classificação em três classes. Nota-se que comparando as imagens das Figuras, fica aparente a dificuldade inicial que têm os modelos em classificar imagens de pulmão com tumor benigno. Entre as causas deste problema o pequeno número de amostras da classe de tumor benigno pode

---

<sup>2</sup>[https://keras.io/api/callbacks/early\\_stopping/](https://keras.io/api/callbacks/early_stopping/)

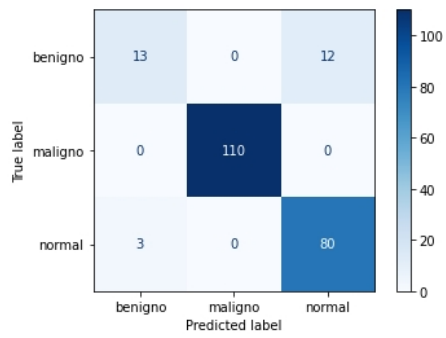
Tabela 5.1: Resultados alcançados com a utilização das funções de perda *Sigmoid Focal Cross Entropy* e *Categorical Cross Entropy*.

| Perda                 | Modelo    | Acurácia      | <i>F-score</i> | Revocação     |
|-----------------------|-----------|---------------|----------------|---------------|
| <i>Focal CE</i>       | VGG       | 0.9220        | 0.8619         | 0.8488        |
|                       | DenseNet  | 0.9311        | 0.8494         | 0.8279        |
|                       | Inception | 0.9220        | 0.8573         | 0.8478        |
|                       | Xception  | <b>0.9350</b> | <b>0.8830</b>  | <b>0.8609</b> |
| <i>Categorical CE</i> | VGG       | <b>0.9495</b> | <b>0.9116</b>  | <b>0.9018</b> |
|                       | DenseNet  | 0.9403        | 0.8892         | 0.8825        |
|                       | Inception | 0.9449        | 0.8974         | 0.8865        |
|                       | Xception  | 0.9266        | 0.8594         | 0.8342        |

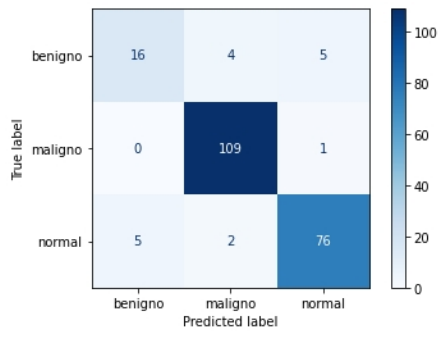
afetar os resultados, o que levou a utilização de métodos de balanceamento de classes. Além dos métodos de balanceamento de classes a utilização função de perda *Categorical Cross Entropy* foi considerada, uma vez que, quando comparada com a função de perda *Sigmoid Focal Cross Entropy*, demonstrou resultados melhores.

Além disso, as Figuras 5.3, 5.4, 5.5 e 5.6 apresentam as curvas de acurácia e perda, respectivamente, ao longo das épocas para as funções de perda aqui abordadas. Nota-se uma considerável distância entre os conjuntos de treinamento e validação, exceto para o modelo VGG que as curvas de treinamento e validação cresceram e diminuíram proporcionalmente. Casos assim podem levar à conclusão de que os modelos estão muito ajustados ao conjunto de treinamento fazendo-se necessária a utilização de técnicas e métodos que diminuam este problema.

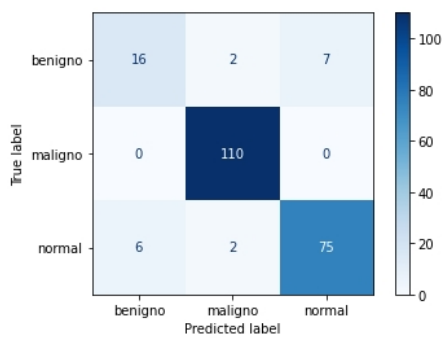
Desta forma, a partir dos próximos experimentos foram utilizadas algumas técnicas de balanceamento de dados que ajudaram a reduzir substancialmente este problema. Além disso, foi mantida a função de perda *Categorical Cross Entropy*, uma vez que os resultados desta sobressaíram-se aos resultados da função *Sigmoid Focal Cross Entropy*.



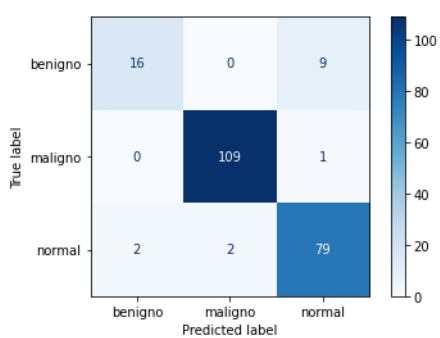
(a) DenseNet



(b) VGG

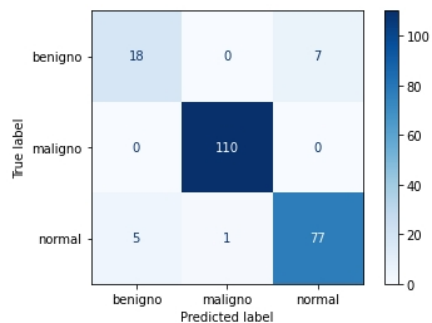


(c) Inception

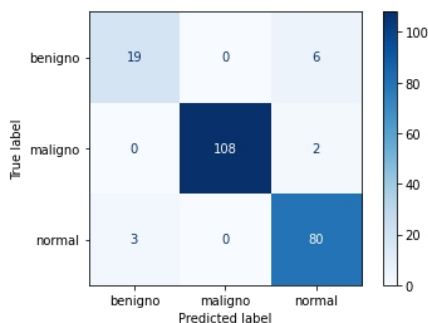


(d) Xception

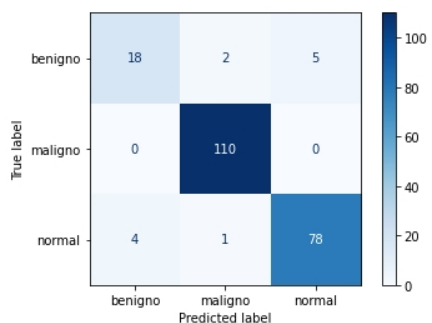
Figura 5.1: Matriz de confusão dos modelos sem aumento de dados com função de perda Focal Loss.



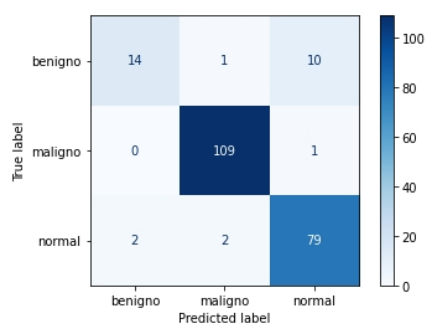
(a) DenseNet



(b) VGG

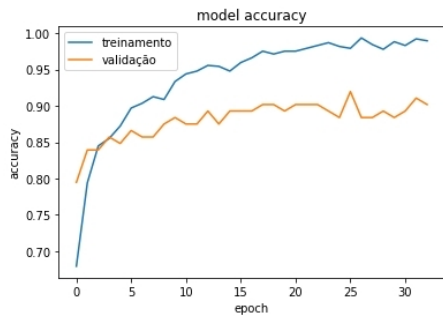


(c) Inception

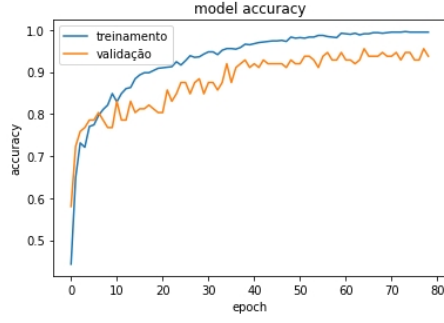


(d) Xception

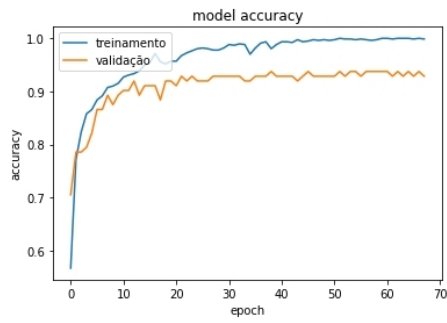
Figura 5.2: Matriz de confusão dos modelos sem aumento de dados com função de perda Categorical Cross Entropy.



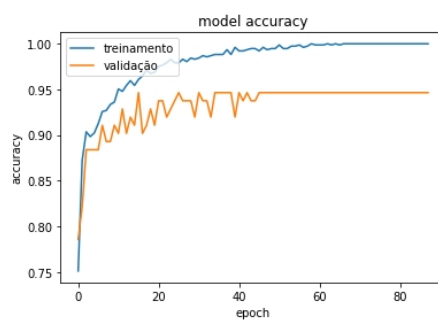
(a) DenseNet



(b) VGG

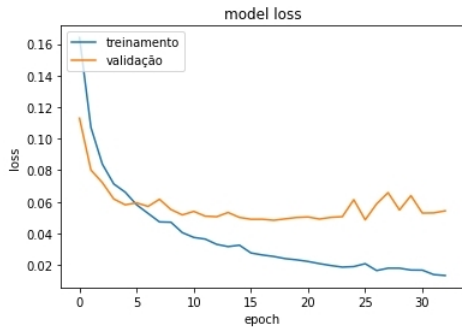


(c) Inception

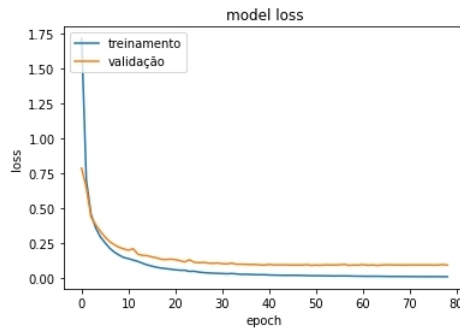


(d) Xception

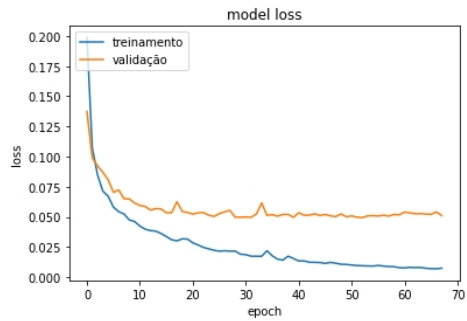
Figura 5.3: Curva de acurácia dos modelos utilizando a função de perda Focal Loss.



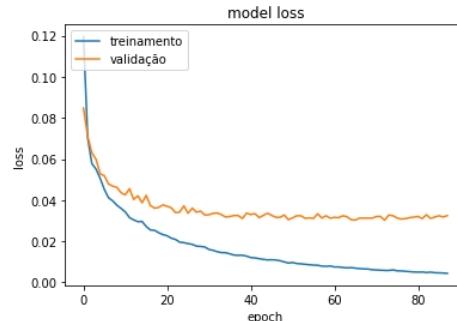
(a) DenseNet



(b) VGG



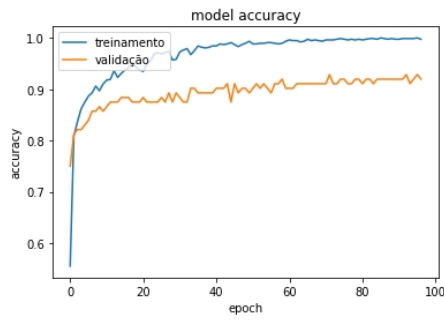
(c) Inception



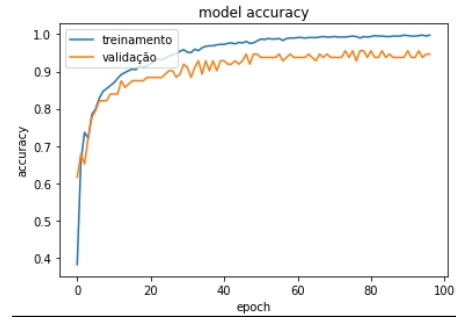
(d) Xception

Figura 5.4: Curva de perda dos modelos utilizando a função de perda Focal Loss.

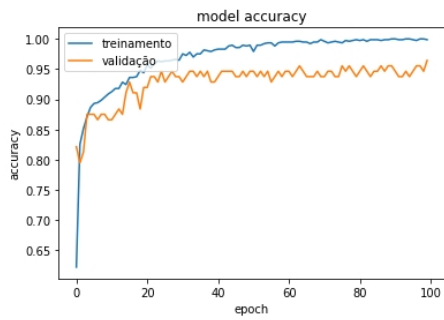




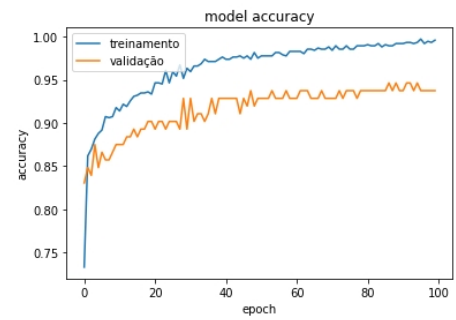
(a) DenseNet



(b) VGG

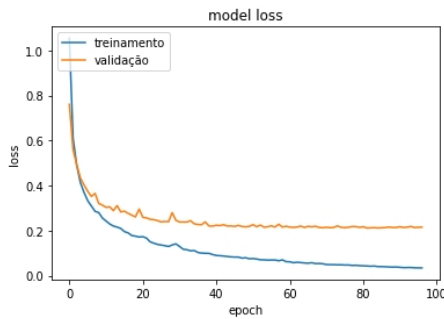


(c) Inception

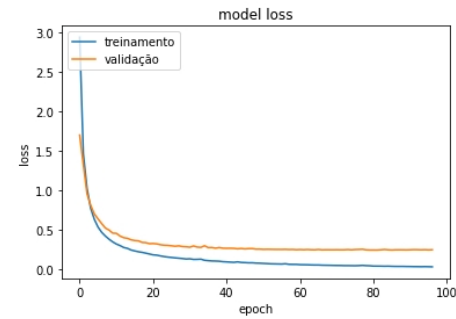


(d) Xception

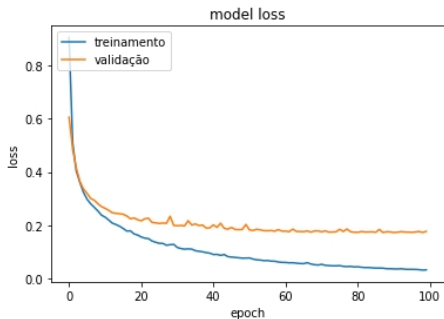
Figura 5.5: Curva de acurácia dos modelos utilizando a função de perda Categorical Cross Entropy.



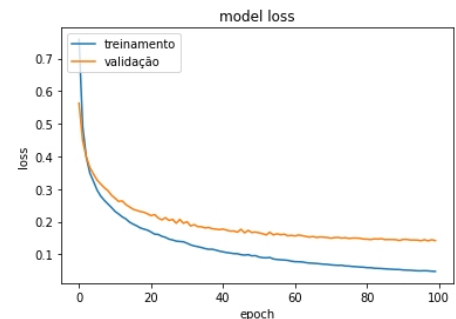
(a) DenseNet



(b) VGG



(c) Inception



(d) Xception

Figura 5.6: Curva de perda dos modelos utilizando a função de perda Categorical Cross Entropy.

## 5.2.2 Modelo com balanceamento de dados

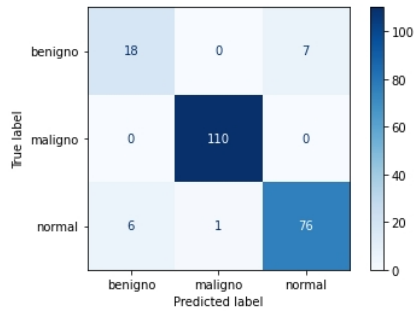
Com o objetivo de reduzir o número de classificações incorretas, entendeu-se que seria necessário o balanceamento de classes no conjunto de imagens, com o fim de se ter, durante o treinamento, uma igualdade no número de amostras das três classes. Inicialmente foram propostos dois métodos de balanceamento e, objetivando validar o desempenho dos mesmos, testes foram executados em ambos. São eles: SMOTE e *Oversample*.

A Tabela 5.2 refere-se aos resultados obtidos pelos modelos quando aplicados a eles os métodos de balanceamento aqui citados. Para o experimento utilizando a técnica de aumento de dados SMOTE, a CNN VGG alcançou o melhor resultado onde obteve um F1-score de 92.42%. Já para o experimento utilizando a técnica de Oversample, o modelo VGG também teve o melhor desempenho, alcançando um F1-score de 95.37%. E por fim, pode-se concluir que o modelo de CNN VGG, utilizando a técnica de aumento de dados Oversample, obteve o melhor resultado geral.

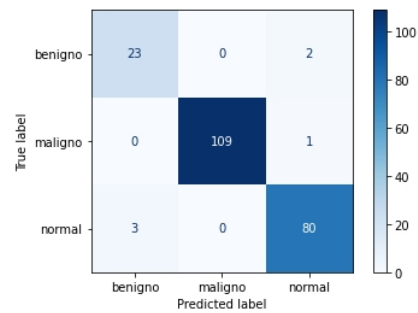
Se comparando aos resultados obtidos pelos modelos sem aumento de dados (Tabela 5.1) utilizando a função *Categorical Cross Entropy*, fica claro que a acurácia dos modelos com as bases de dados balanceadas está relativamente maior, no entanto, não se pode concluir acerca do melhor modelo apenas utilizando como base esta métrica, como foi citado na Subseção 2.7. As Figuras 5.7 e 5.8 expõem a melhoria substancial na diminuição de classificações incorretas em relação aos primeiros modelos, quando comparadas as classes que previamente eram menos classificadas. Como concluído acerca do modelo que melhor desempenhou nesse experimento, a Figura 5.7 evidencia que o modelo de CNN VGG obteve melhores resultados nas classificações, tendo errado 6 das 128 amostras de teste classificadas.

Tabela 5.2: Métricas alcançadas com a utilização dos métodos SMOTE e Oversample para balanceamento de dados.

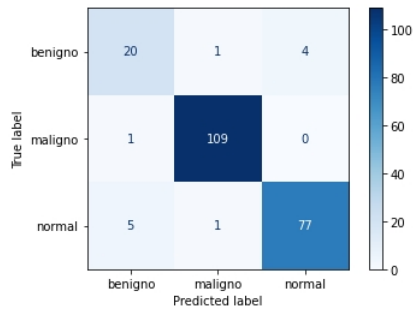
| Balanceamento | Modelo    | Acurácia      | <i>F-score</i> | Revocação     |
|---------------|-----------|---------------|----------------|---------------|
| SMOTE         | VGG       | <b>0.9541</b> | <b>0.9242</b>  | 0.9059        |
|               | DenseNet  | 0.9403        | 0.8892         | 0.8825        |
|               | Inception | 0.9495        | 0.9073         | 0.9009        |
|               | Xception  | 0.9266        | 0.8899         | <b>0.9107</b> |
| Oversample    | VGG       | <b>0.9724</b> | <b>0.9537</b>  | <b>0.9582</b> |
|               | DenseNet  | 0.9357        | 0.8819         | 0.8785        |
|               | Inception | 0.9449        | 0.9032         | 0.9062        |
|               | Xception  | 0.9357        | 0.8974         | 0.9074        |



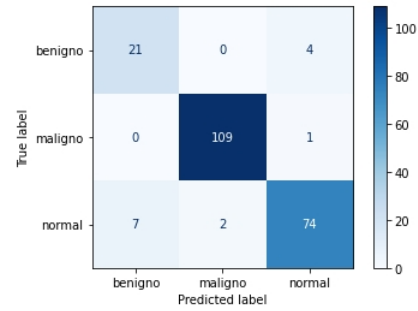
(a) DenseNet



(b) VGG

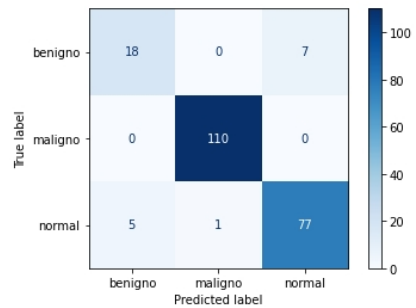


(c) Inception

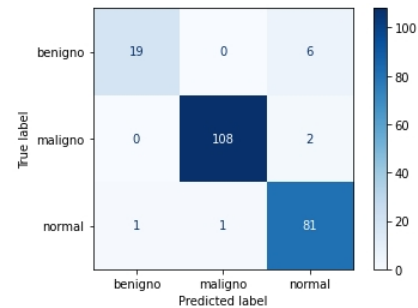


(d) Xception

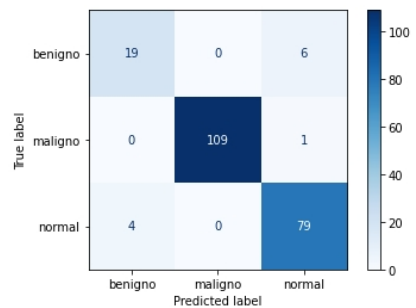
Figura 5.7: Matriz de confusão dos modelos com balanceamento de classes utilizando *Oversample*.



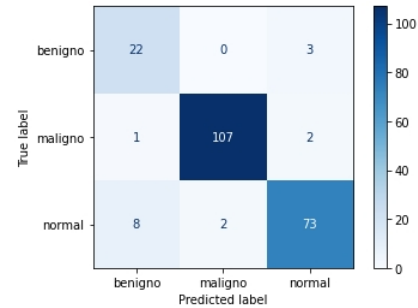
(a) DenseNet



(b) VGG



(c) Inception



(d) Xception

Figura 5.8: Matriz de confusão dos modelos com balanceamento de classes utilizando SMOTE.

Além disso, para validar o sucesso na utilização de métodos de balanceamento das classes, as Figuras 5.10, 5.11, 5.12 e 5.13 demonstram as curvas de aprendizado e validação das acurácias e perdas dos modelos utilizando os métodos SMOTE e *Oversample*. Comparando-se com as curvas de acurácia e perda dos modelos sem balanceamento (Figuras 5.3, 5.4, 5.5 e 5.6), fica clara a melhora obtida. Inicialmente observa-se que os modelos não rodaram as cem épocas como ocorreu no primeiro modelo. A condição de parada precoce evitou a ocorrência de sobre-ajuste parando a execução previamente. Além disso, as curvas de treinamento e validação para os modelo VGG crescem e diminuem na mesma proporção, o que indica que o sobre-ajuste que outrora poderia estar ocorrendo, foi reduzido consideravelmente.

## Modelo com balanceamento de classes e aumento de dados

Por fim, foi aplicado aos modelos com classes balanceadas aumento de dados objetivando um maior número de amostras durante o treinamento destes. A Tabela 5.3 indica as métricas obtidas quando realizados os testes com o balanceamento de classes e o aumento de dados. Apesar de serem métricas consistentes, identificou-se um grande número de classificações incorretas quando se comparado aos modelos com somente balanceamento de classes. Além deste problema, as curvas de treinamento e validação comprovaram um baixo desempenho, quando nos gráficos observou-se uma grande distância entre as mesmas, o que demonstra um sobre-ajuste do modelo ao conjunto de treinamento. Há diversos fatores que podem ter influenciado a ocorrência deste problema, como por exemplo, baixa qualidade de imagens após o aumento dos dados e a base de dados com poucas amostras.

Tabela 5.3: Métricas alcançadas com a utilização do SMOTE para balanceamento de classes e aumento de dados para gerar novas imagens durante o treinamento.

| Modelo    | Acurácia      | <i>F-score</i> | Revocação     |
|-----------|---------------|----------------|---------------|
| VGG       | <b>0.9082</b> | <b>0.8423</b>  | <b>0.8480</b> |
| DenseNet  | 0.8577        | 0.7832         | 0.8068        |
| Inception | 0.8899        | 0.8169         | 0.8237        |
| Xception  | 0.8623        | 0.7361         | 0.7343        |

Nota-se que após a utilização de aumento de dados durante o treinamento para geração de novas amostras, o modelo utilizando a CNN VGG obteve os melhores resultados. Porém, comparando estes resultados aos obtidos nos modelos somente com balanceamento de classes, é notável que não houveram melhoras. A revocação obtida demonstra que o modelo não foi capaz de classificar com sucesso as imagens de uma determinada classe,

como demonstra a Figura 5.9. Nela, é possível verificar que o modelo piorou quando observados o número de erros de classificação.

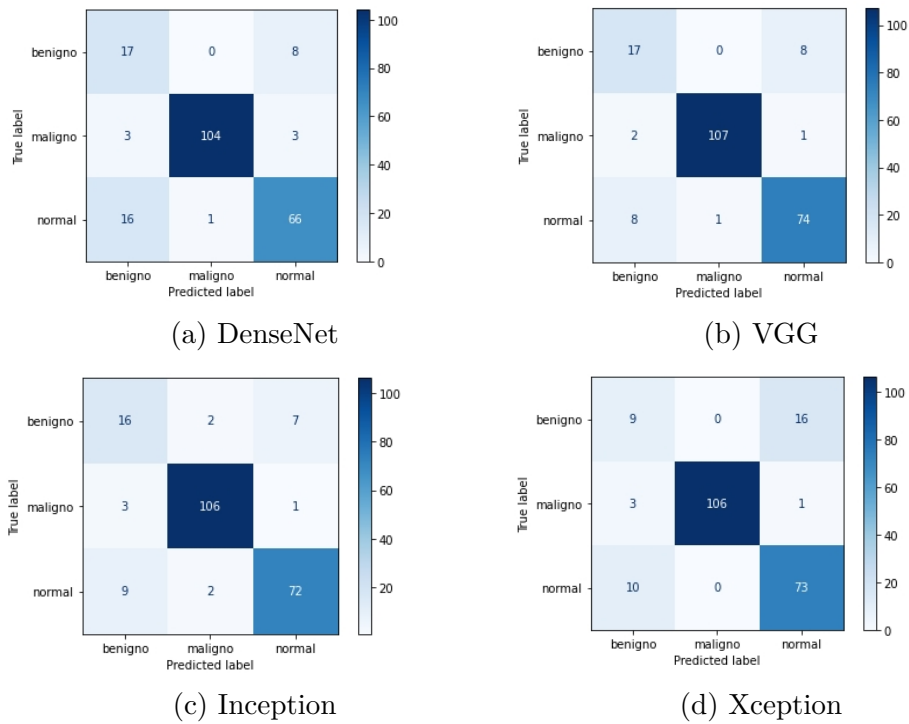


Figura 5.9: Matriz de confusão dos modelos com *Oversample* e aumento de dados.

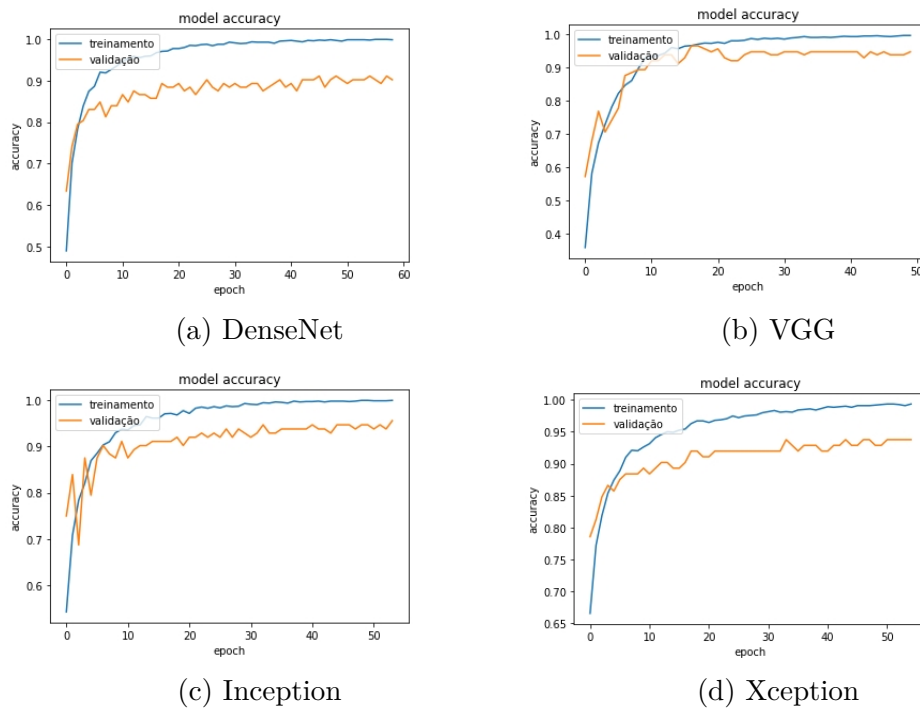
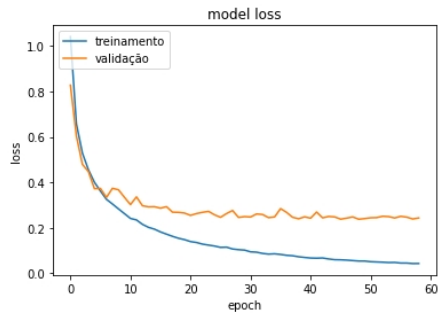
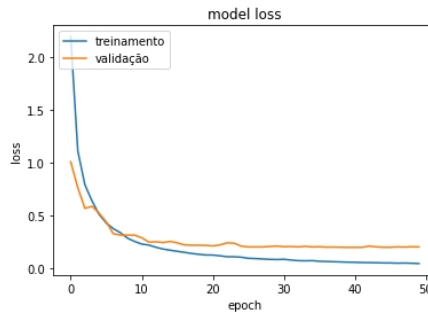


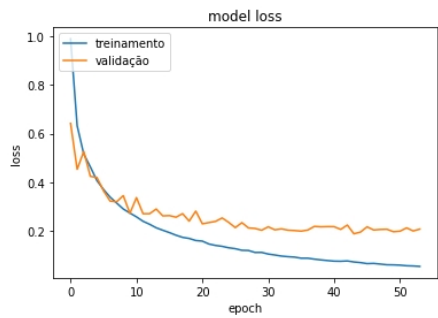
Figura 5.10: Curva de acurácia dos modelos utilizando balanceamento de classes *Over-sample*.



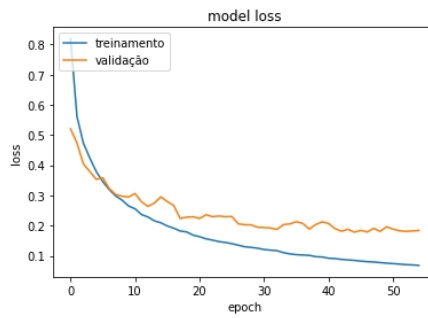
(a) DenseNet



(b) VGG

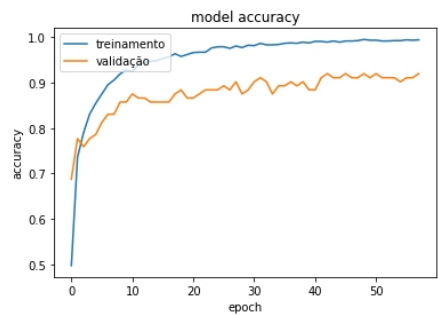


(c) Inception

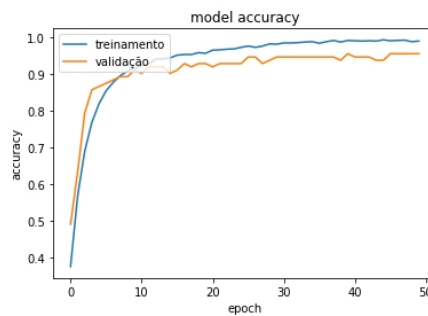


(d) Xception

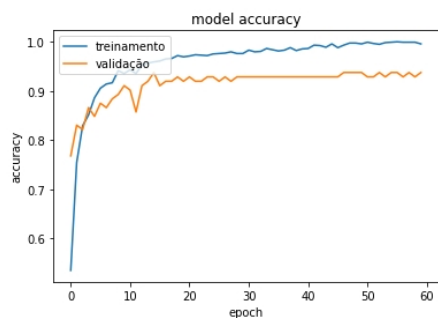
Figura 5.11: Curva de perda dos modelos utilizando balanceamento de classes Oversample.



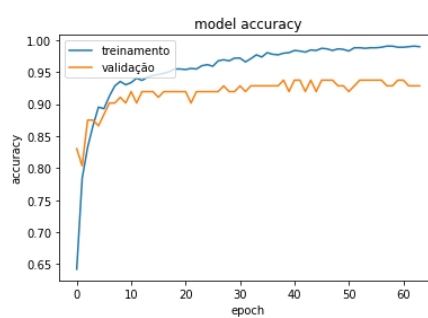
(a) DenseNet



(b) VGG



(c) Inception



(d) Xception

Figura 5.12: Curva de acurácia dos modelos utilizando balanceamento de classes SMOTE.

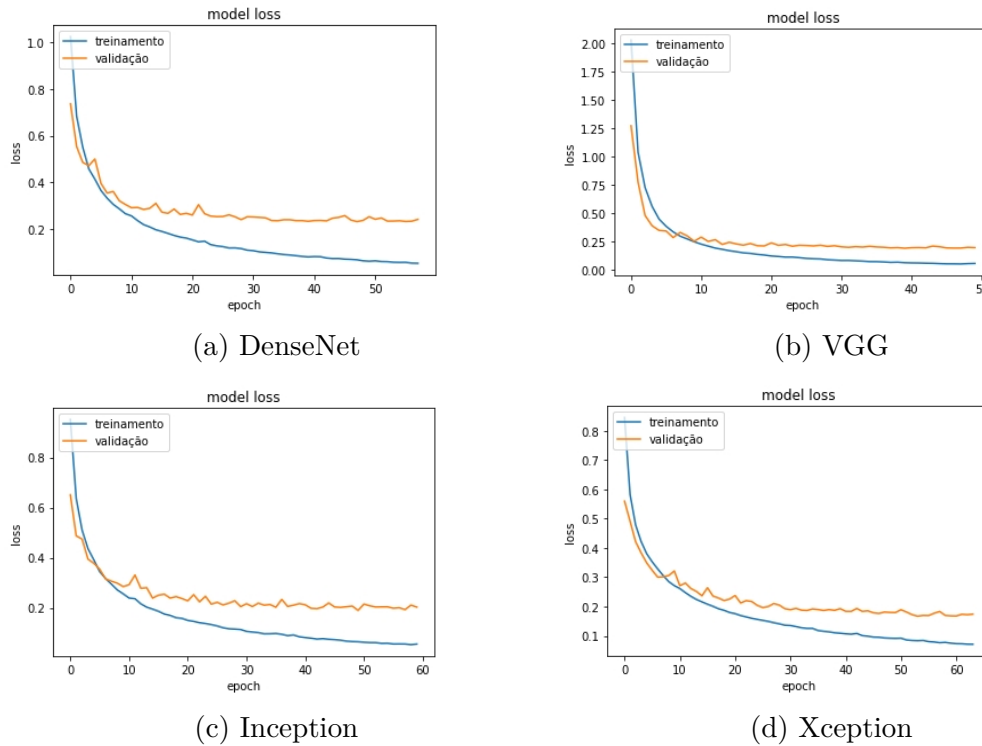


Figura 5.13: Curva de perda dos modelos utilizando balanceamento de classes SMOTE.

### Síntese dos experimentos

Assim, tendo realizado todas as execuções dos experimentos propostos nesta seção, ficou claro que a utilização da função de perda *Categorical CE* alcançou melhores resultados na base de dados utilizada ao comparar com a função de perda *Focal CE*. Além disso, foi evidenciado que a utilização de métodos de balanceamento de classes melhorou substancialmente os resultados para os todos os modelos de CNN aqui utilizados. Por fim, verificou-se que a utilização de aumento de dados no processo de treinamento dos modelos diminuiu a assertividade dos modelos. A Tabela 5.4 sumariza os melhores resultados obtidos nos experimentos de CNN completa.

Tabela 5.4: Resumo de melhores métricas obtidas nos experimentos.

| Experimento                      | Modelo | Acurácia      | <i>F-score</i> | Revocação     |
|----------------------------------|--------|---------------|----------------|---------------|
| Sem balanceamento                | VGG    | 0.9495        | 0.9116         | 0.9018        |
| <i>Oversample</i>                | VGG    | <b>0.9724</b> | <b>0.9537</b>  | <b>0.9582</b> |
| Balanceamento e aumento de dados | VGG    | 0.9082        | 0.8423         | 0.8480        |

## 5.3 Experimento II - Modelo híbrido

Neste experimento modelos de CNNs previamente treinadas foram utilizados removendo suas camadas totalmente conectadas obtendo somente as características essenciais das imagens. Junto a estas, foram utilizados classificadores tradicionais de aprendizado de máquina para tarefa de classificação de imagens de tomografias computadorizadas do pulmão. As subseções a seguir detalham os resultados obtidos nas execuções sem balanceamento de classes e posteriormente execuções com balanceamento de classes utilizando os métodos SMOTE e *Oversample*.

### 5.3.1 Modelo híbrido sem balanceamento de classes

Inicialmente, com o fim de validar o desempenho dos modelos híbridos na tarefa de classificação quando utilizados em bases de dados desbalanceadas, este experimento foi realizado sem utilizar os métodos de balanceamento e aumento de dados no conjunto de imagens, mantendo somente os pré-processamentos de imagens necessários. Na Tabela 5.5 é possível observar as métricas alcançadas para este experimento. Nota-se que mesmo sem balanceamento de classe, os resultados obtidos nos modelos híbridos são superiores aos modelos de CNN completa, dispostos na Tabela 5.7. Portanto, o modelo híbrido de VGG juntamente com o classificador de regressão linear obteve o melhor resultado, tendo um F1-score de 96.53%. Além disso, observa-se que todos os modelos utilizando o classificador de árvore de decisão alcançaram resultados muito abaixo, quando comparados com resultados obtidos em outros classificadores.

Por fim, as imagens das Figuras 5.14, 5.15, 5.16, 5.17 e 5.18 ilustram os resultados obtidos na matriz de confusão de cada um dos classificadores. É possível notar que, a imagem da Figura 5.15 evidencia o resultado discutido no parágrafo anterior, mostrando que o modelo VGG com o classificador de regressão logística sobressaiu-se, errando 4 das 128 imagens classificadas.

Nota-se que, apesar de bons valores de F1-score nos modelos híbridos, ainda há uma dificuldade em classificar amostras da classe de tumor benigno, o que levou à proposição dos métodos de balanceamento de dados nos próximos experimentos.



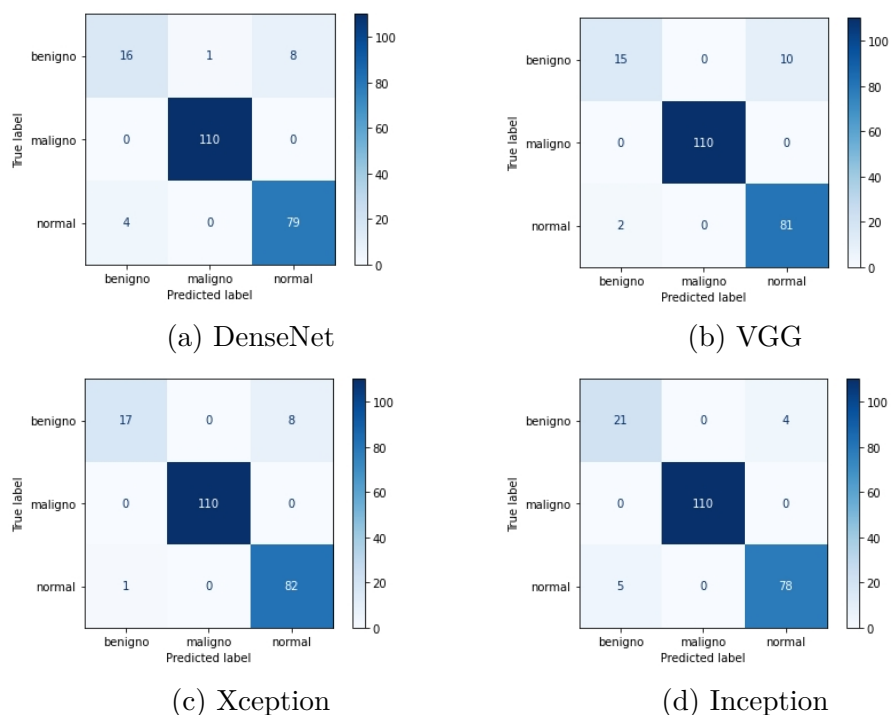
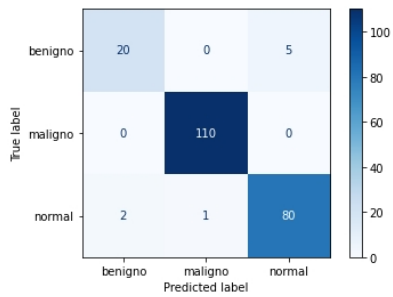


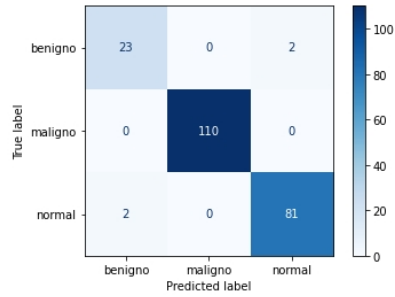
Figura 5.14: Matriz de confusão do modelo híbrido KNN sem balanceamento de classes.

Tabela 5.5: Métricas dos modelos híbridos sem balanceamento de classes.

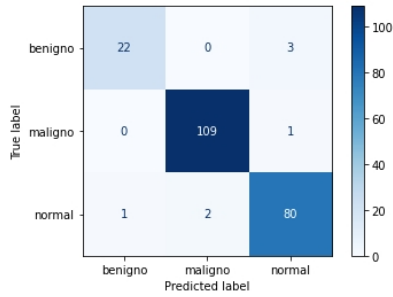
| Classificador     | Modelo    | Acurácia      | $F$ -score    | Revocação     |
|-------------------|-----------|---------------|---------------|---------------|
| Regressão         | VGG       | <b>0.9816</b> | <b>0.9653</b> | <b>0.9653</b> |
|                   | Xception  | 0.9678        | 0.9537        | 0.9449        |
|                   | DenseNet  | 0.9633        | 0.9329        | 0.9212        |
|                   | Inception | 0.9633        | 0.9333        | 0.9212        |
| KNN               | VGG       | 0.9449        | 0.8817        | 0.8586        |
|                   | Xception  | <b>0.9587</b> | 0.9128        | 0.8893        |
|                   | DenseNet  | 0.9403        | 0.8786        | 0.8639        |
|                   | Inception | <b>0.9587</b> | <b>0.9229</b> | <b>0.9265</b> |
| SGDC              | VGG       | <b>0.9724</b> | 0.9483        | 0.9479        |
|                   | Xception  | 0.9770        | <b>0.9589</b> | <b>0.9715</b> |
|                   | DenseNet  | 0.9587        | 0.9184        | 0.9079        |
|                   | Inception | 0.9495        | 0.9135        | 0.9205        |
| Árvore de Decisão | VGG       | <b>0.9082</b> | <b>0.8637</b> | <b>0.8770</b> |
|                   | Xception  | 0.8807        | 0.8114        | 0.7999        |
|                   | DenseNet  | 0.8899        | 0.8294        | 0.8237        |
|                   | Inception | 0.8944        | 0.8314        | 0.8306        |
| SVM               | VGG       | <b>0.9770</b> | <b>0.9572</b> | <b>0.9612</b> |
|                   | Xception  | 0.9403        | 0.9004        | 0.8928        |
|                   | DenseNet  | 0.95412       | 0.9171        | 0.9132        |
|                   | Inception | 0.9541        | 0.9171        | 0.9132        |



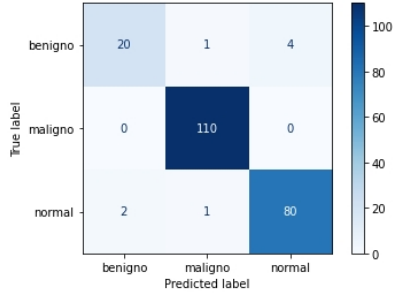
(a) DenseNet



(b) VGG

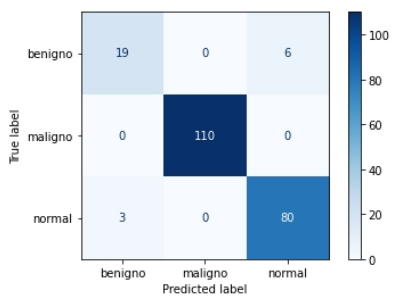


(c) Xception

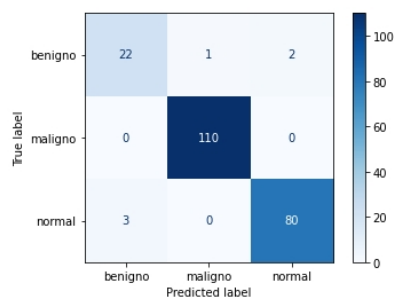


(d) Inception

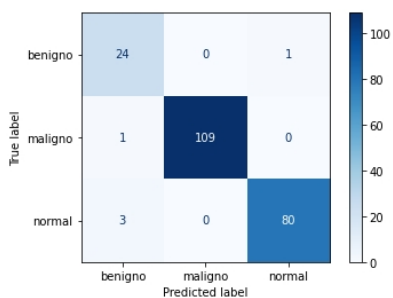
Figura 5.15: Matriz de confusão do modelo híbrido de Regressão Logística sem balanceamento de classes.



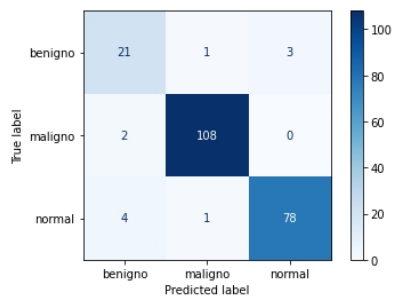
(a) DenseNet



(b) VGG



(c) Xception



(d) Inception

Figura 5.16: Matriz de confusão do modelo híbrido SGDC sem balanceamento de classes.

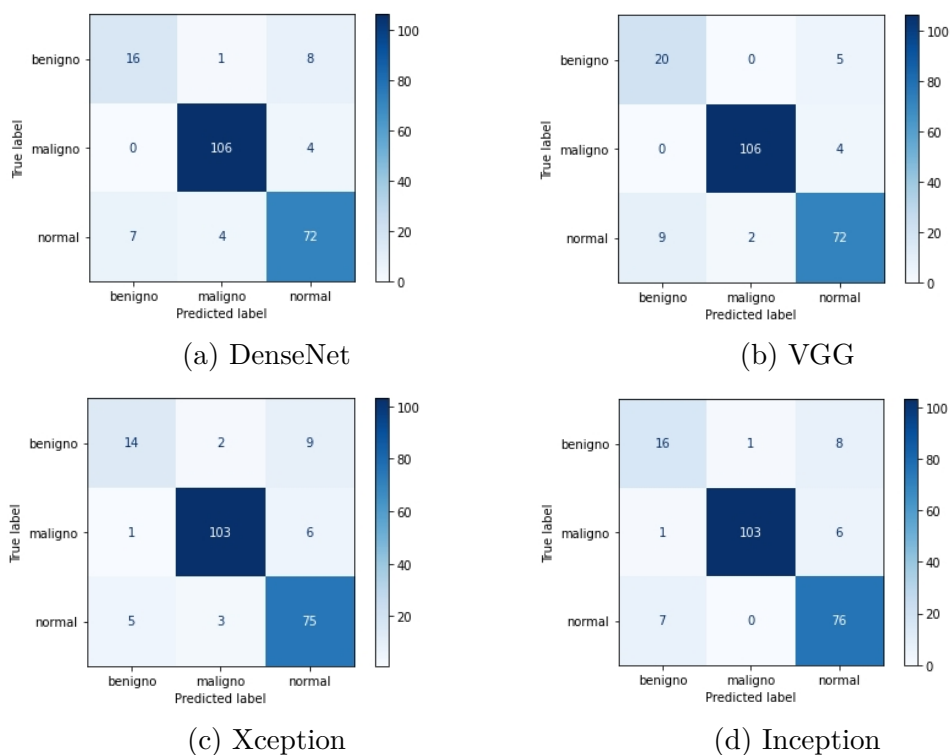


Figura 5.17: Matriz de confusão do modelo híbrido Árvore de Decisão sem balanceamento de classes.

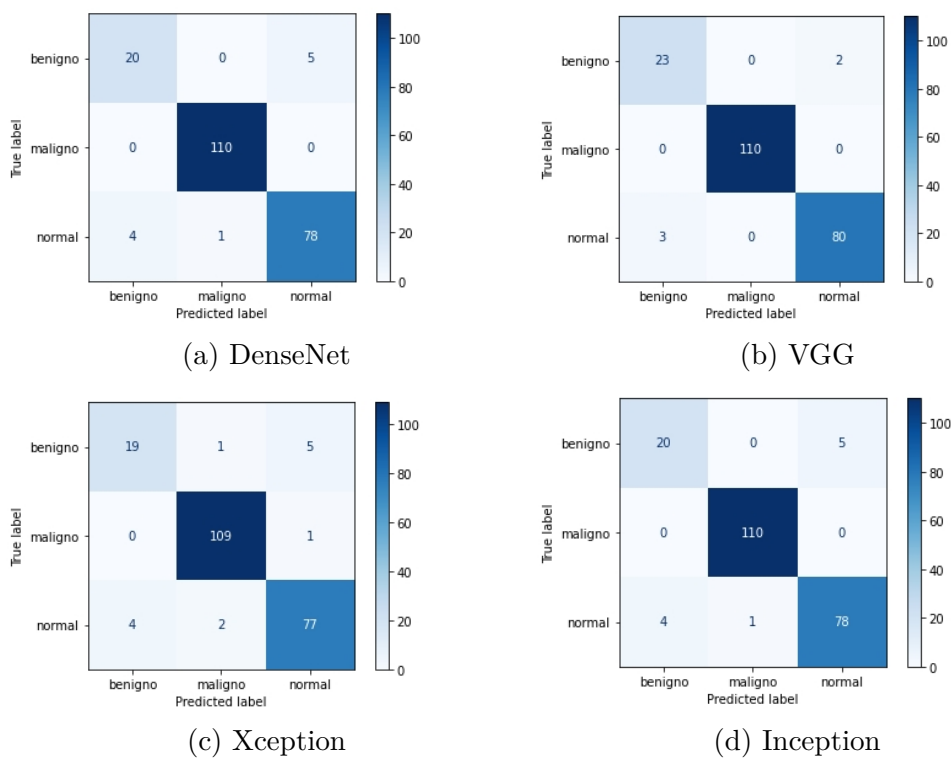


Figura 5.18: Matriz de confusão do modelo híbrido SVM sem balanceamento de classes.

### 5.3.2 Modelo híbrido com balanceamento de dados

Nesta subseção são testados os modelos híbridos abordados nas Subseções anteriores aplicando-se a eles métodos de aumento de dados com o objetivo de balanceamento de classes da mesma maneira que foram utilizados nos modelos de CNN completa: SMOTE e *Oversample*. O objetivo deste experimento é validar se haverá melhorias consideráveis nas métricas ao utilizar estes métodos, além da verificação da redução de classificações incorretas da classe de tumores benignos.

#### SMOTE

Como já citado no Capítulo 4, o método SMOTE gera imagens sintéticas a partir das amostras existentes no conjunto de dados. Neste experimento este método foi aplicado ao conjunto de treinamento e foram realizadas as execuções dos modelos aqui propostos.

A Tabela 5.6 demonstra as métricas obtidas para todos os modelos. O modelo híbrido utilizando a CNN Xception juntamente ao classificador de regressão linear alcançou o melhor desempenho obtendo um F1-score de 96.65%. Vale citar também que o modelo VGG utilizando os classificadores de regressão linear e SVM separadamente, alcançou a melhor acurácia e um F1-score de 96.53%. Assim, analisando os modelos híbridos aqui citados, entende-se que os modelos utilizando a VGG obtiveram melhor desempenho na tarefa de classificação em classes balanceadas pela técnica SMOTE.

Nota-se que os resultados alcançados são similares aos obtidos no modelo sem balanceamento de classes (Tabela 5.5). Porém, utilizando o método de balanceamento em questão, o modelo VGG combinado com os classificadores de Regressão Linear e SVM obtiveram os melhores resultados, enquanto anteriormente somente o classificador de Regressão Logística havia alcançado as melhores métricas.

As Figuras 5.19, 5.20, 5.21, 5.22 e 5.23 ilustram os resultados obtidos nas execuções dos modelos híbridos com classes balanceadas utilizando o SMOTE. Os modelos utilizando VGG e os classificadores de Regressão Logística e SVM demonstraram os melhores resultados, quando só houveram quatro classificações incorretas, sendo duas de classe benigna e duas de classe normal, como já era previsto a partir dos resultados encontrados na Tabela 5.6.

Tabela 5.6: Modelos híbridos com classificador com balanceamento de classes utilizando SMOTE.

| Classificador     | Modelo    | Acurácia      | $F$ -score    | Revocação     |
|-------------------|-----------|---------------|---------------|---------------|
| Regressão         | VGG       | <b>0.9816</b> | 0.9653        | <b>0.9653</b> |
|                   | Xception  | 0.9770        | <b>0.9665</b> | 0.9622        |
|                   | DenseNet  | 0.9678        | 0.9401        | 0.9439        |
|                   | Inception | 0.9724        | 0.9464        | 0.9386        |
| KNN               | VGG       | <b>0.9678</b> | <b>0.9365</b> | <b>0.9253</b> |
|                   | Xception  | 0.9541        | 0.9112        | 0.9049        |
|                   | DenseNet  | 0.9495        | 0.9032        | 0.8999        |
|                   | Inception | 0.9495        | 0.9089        | 0.9195        |
| Árvore de Decisão | VGG       | <b>0.8944</b> | <b>0.8549</b> | <b>0.8782</b> |
|                   | Xception  | 0.8394        | 0.7791        | 0.8016        |
|                   | DenseNet  | 0.8715        | 0.7982        | 0.8052        |
|                   | Inception | 0.8669        | 0.8140        | 0.8281        |
| SGDC              | VGG       | <b>0.9724</b> | 0.9506        | 0.9665        |
|                   | Xception  | 0.9678        | <b>0.9521</b> | <b>0.9728</b> |
|                   | DenseNet  | 0.9678        | 0.9445        | 0.9449        |
|                   | Inception | 0.9541        | 0.9176        | 0.9318        |
| SVM               | VGG       | <b>0.9816</b> | <b>0.9653</b> | <b>0.9653</b> |
|                   | Xception  | 0.9541        | 0.9261        | 0.9152        |
|                   | DenseNet  | 0.9633        | 0.9263        | 0.9119        |
|                   | Inception | 0.9587        | 0.9157        | 0.8986        |

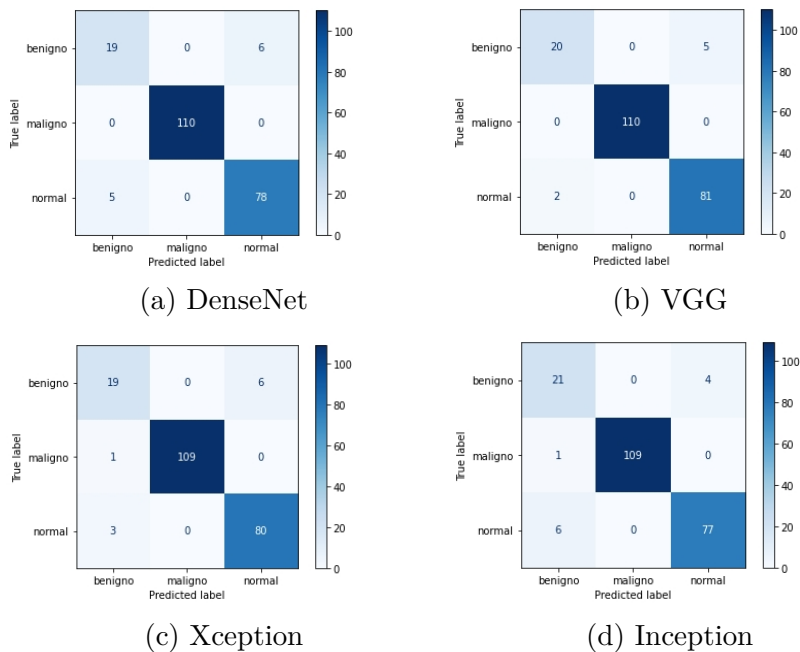


Figura 5.19: Matriz de confusão do modelo híbrido KNN com balanceamento de classes SMOTE.

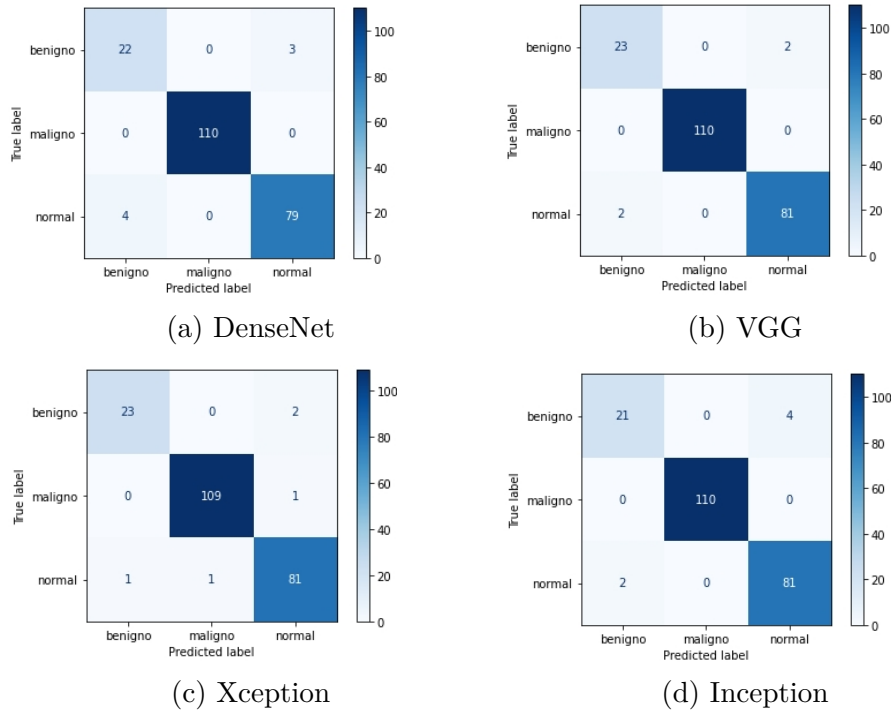


Figura 5.20: Matriz de confusão do modelo híbrido de Regressão Logística com balanceamento de classes SMOTE.

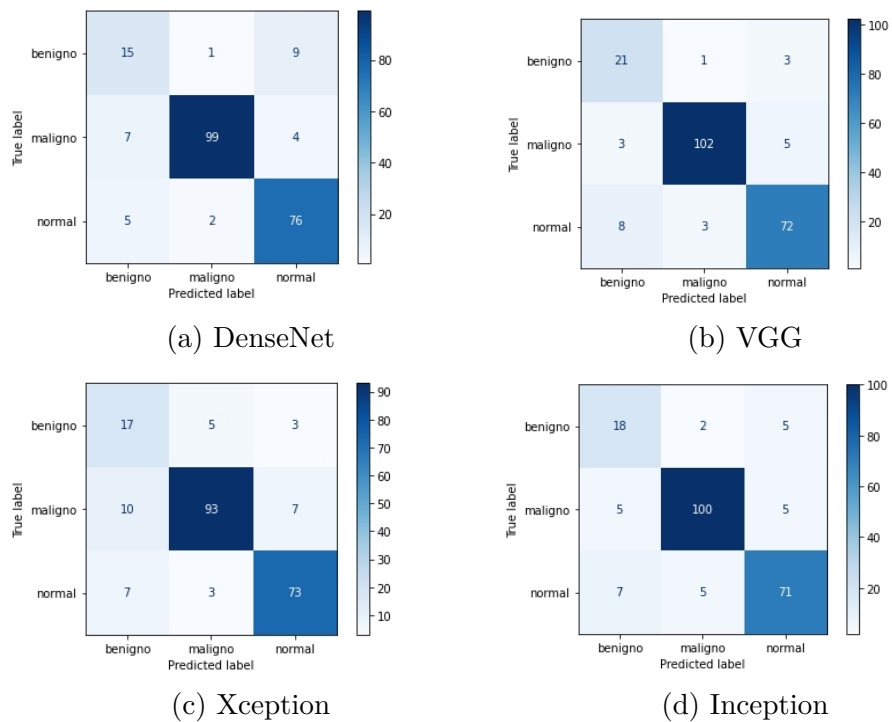


Figura 5.21: Matriz de confusão do modelo híbrido de Árvore de Decisão com balanceamento de classes SMOTE.

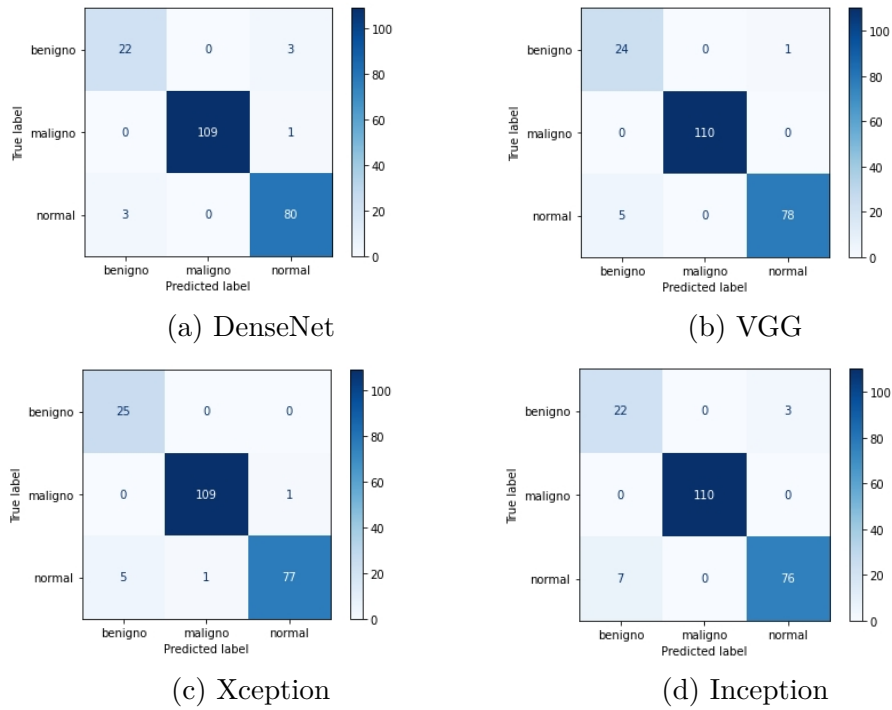


Figura 5.22: Matriz de confusão do modelo híbrido SGDC com balanceamento de classes SMOTE.

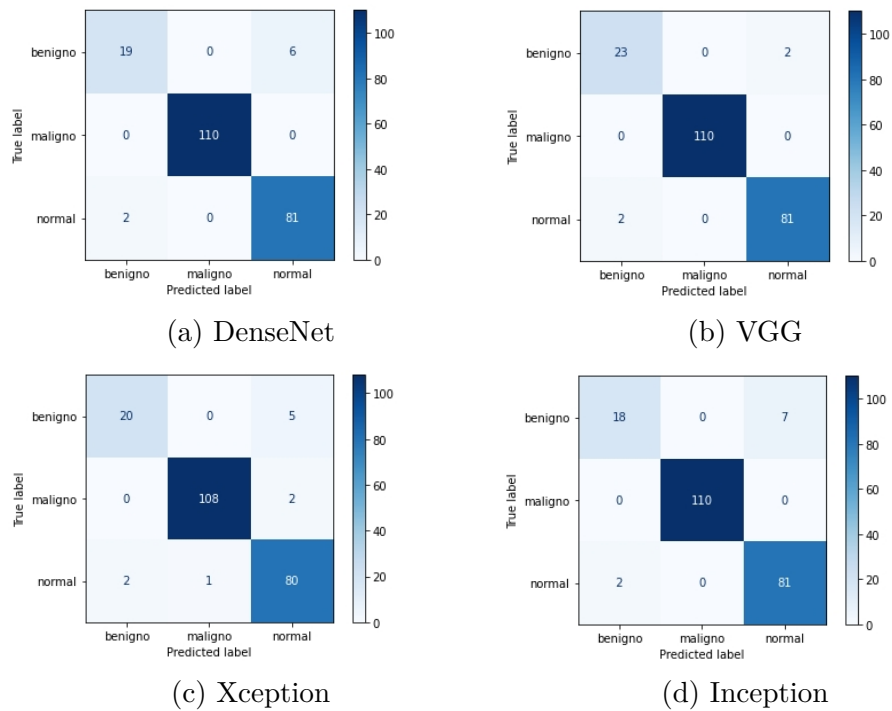


Figura 5.23: Matriz de confusão do modelo híbrido SVM com balanceamento de classes SMOTE.

## Oversample

Objetivando comparar o desempenho obtido quando utilizados os métodos de aumento de dados com objetivo de balanceamento de classes, também foi utilizado o *Oversample*, método citado no capítulo de metodologia. Este método somente replica amostras já existentes nos conjuntos de dados, o que poderá levar os modelos a um sobre-ajuste para determinadas classes.

Na Tabela 5.7 é possível verificar os resultados obtidos após a execução de todos os modelos. Nota-se que a acurácia manteve-se igual à dos experimentos anteriores onde foram utilizados modelos híbridos. No entanto, este modelo apresentou bons resultados para a acurácia para três classificadores, um a mais que o método de balanceamento SMOTE. Ademais, o modelo utilizando a CNN Xception e o classificador de Regressão Logística alcançou os maiores valores das acurácias de F-score e revocação, o que gerou o melhor resultado de classificação da classe de tumores benignos. Por fim, analisando os resultados obtidos pelos modelos de CNN quando utilizado o classificador de árvore aleatória, é evidente que estes modelos tiveram maior dificuldade de classificação, obtendo assim métricas abaixo do esperado.

As Figuras 5.24, 5.25, 5.26, 5.27 demonstram os resultados obtidos em todos os modelos balanceados com o método *Oversample*. Desta vez, a CNN VGG com classificador de Regressão Logística, KNN e SVM apresentou os melhores resultados. Além disso, vale citar os resultados obtidos pela CNN Xception utilizando o classificador de Regressão Logística, no qual este obteve o maior acerto para a classe de tumores benignos, havendo somente duas classificações incorretas.

## Síntese dos experimentos

Assim, tendo realizado todas as execuções dos experimentos propostos nesta seção, ficou claro que a utilização de métodos de balanceamento de classes não melhorou os resultados para os modelos que já haviam demonstrado um bom desempenho, uma vez que as métricas dos melhores modelos foram similares. No entanto, percebe-se que, utilizando os métodos SMOTE e *Oversample* para balancear os modelos, mais de um modelo híbrido alcançou as melhores métricas. A Tabela 5.8 apresenta um resumo das melhores métricas obtidas nos experimentos com modelos híbridos.



Tabela 5.7: Modelos híbridos com classificador com balanceamento de classes utilizando Oversample.

| Classificador     | Modelo    | Acurácia      | $F$ -score    | Revocação     |
|-------------------|-----------|---------------|---------------|---------------|
| Regressão         | VGG       | <b>0.9816</b> | 0.9653        | 0.9653        |
|                   | Xception  | 0.9770        | <b>0.9720</b> | <b>0.9715</b> |
|                   | DenseNet  | 0.9724        | 0.9479        | 0.9479        |
|                   | Inception | 0.9724        | 0.9524        | 0.9479        |
| KNN               | VGG       | <b>0.9816</b> | <b>0.9653</b> | 0.9653        |
|                   | Xception  | 0.9541        | 0.9203        | 0.9421        |
|                   | DenseNet  | 0.9724        | 0.9506        | <b>0.9665</b> |
|                   | Inception | 0.9587        | 0.9274        | 0.9462        |
| Árvore de Decisão | VGG       | <b>0.9311</b> | 0.8758        | 0.8578        |
|                   | Xception  | 0.8807        | 0.8231        | 0.8362        |
|                   | DenseNet  | <b>0.9311</b> | <b>0.8832</b> | <b>0.8662</b> |
|                   | Inception | 0.8486        | 0.7805        | 0.7718        |
| SGDC              | VGG       | <b>0.9770</b> | <b>0.9577</b> | <b>0.9622</b> |
|                   | Xception  | 0.9678        | 0.9489        | 0.9449        |
|                   | DenseNet  | 0.9633        | 0.9311        | 0.9315        |
|                   | Inception | 0.9495        | 0.9135        | 0.9205        |
| SVM               | VGG       | <b>0.9816</b> | <b>0.9653</b> | <b>0.9653</b> |
|                   | Xception  | 0.9633        | 0.9453        | 0.9409        |
|                   | DenseNet  | 0.9633        | 0.9329        | 0.9212        |
|                   | Inception | 0.9678        | 0.9428        | 0.9346        |

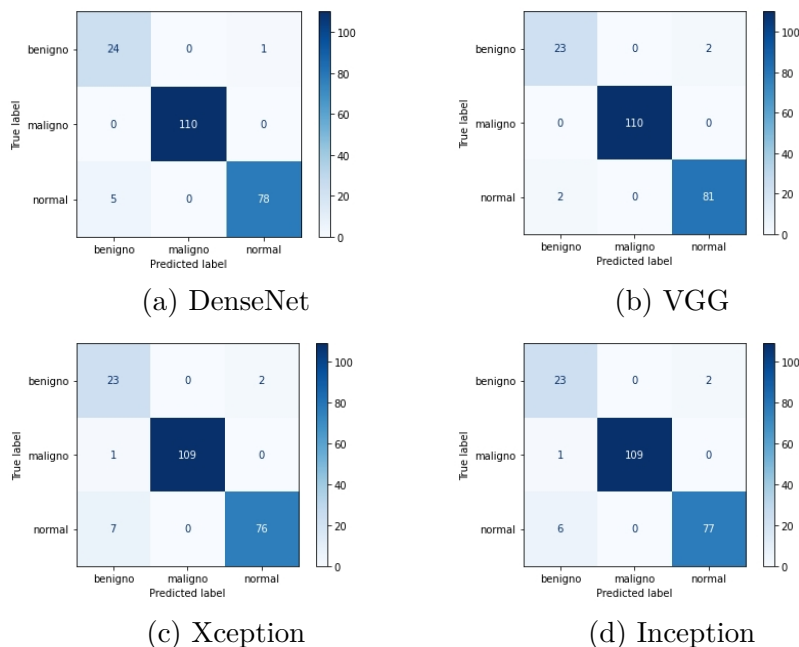
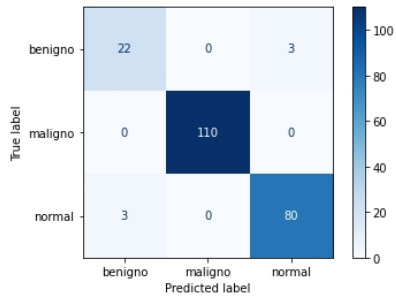
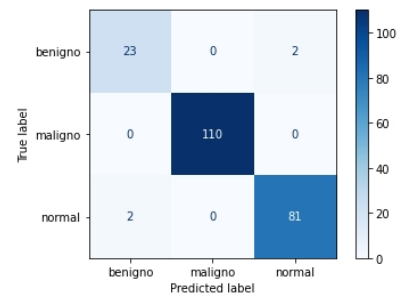


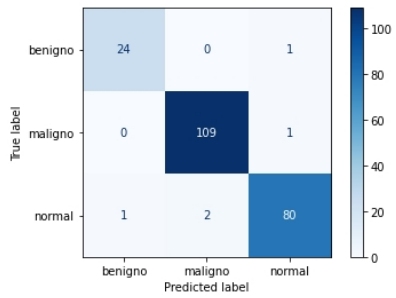
Figura 5.24: Matriz de confusão do modelo híbrido KNN com balanceamento de classes Oversample.



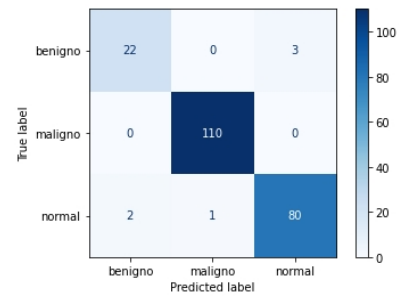
(a) DenseNet



(b) VGG

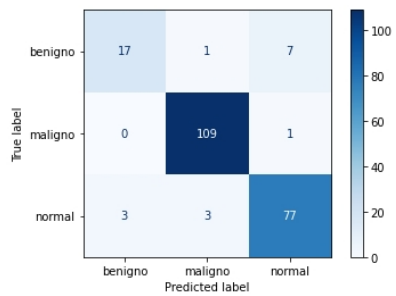


(c) Xception

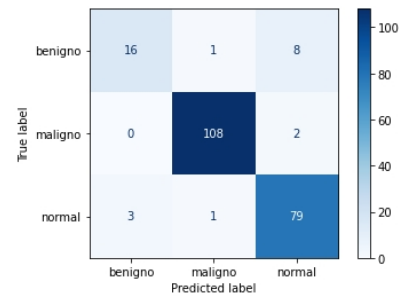


(d) Inception

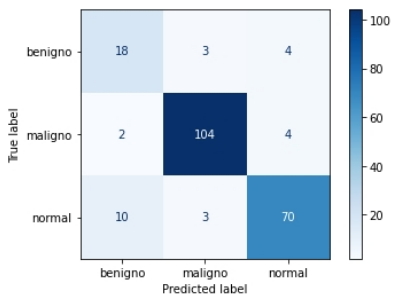
Figura 5.25: Matriz de confusão do modelo híbrido de Regressão Logística com balanceamento de classes Oversample.



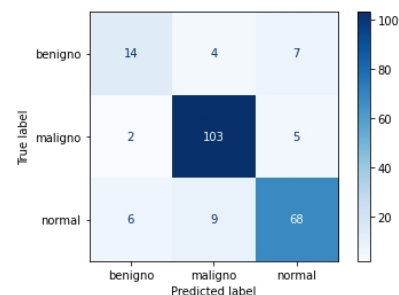
(a) DenseNet



(b) VGG



(c) Xception



(d) Inception

Figura 5.26: Matriz de confusão do modelo híbrido de Árvore de Decisão com balanceamento de classes Oversample.

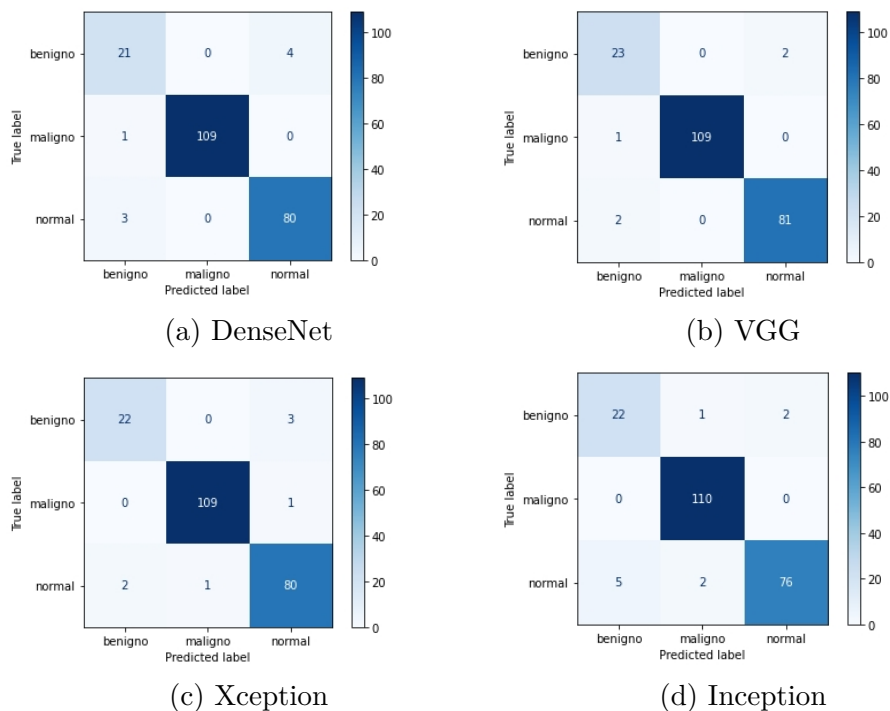


Figura 5.27: Matriz de confusão do modelo híbrido SGDC com balanceamento de classes Oversample.

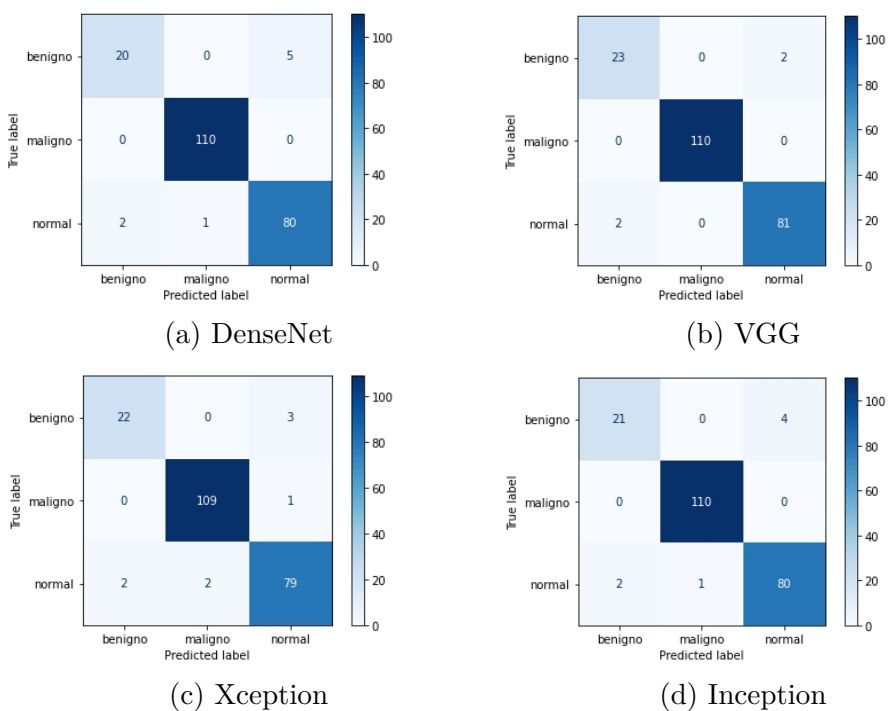


Figura 5.28: Matriz de confusão do modelo híbrido SVM com balanceamento de classes Oversample.

Tabela 5.8: Resumo de melhores métricas obtidas nos experimentos com modelos híbridos.

| Experimento       | Modelo   | Classificador | Acurácia      | <i>F-score</i> | Revocação     |
|-------------------|----------|---------------|---------------|----------------|---------------|
| Sem balanceamento | VGG      | Regressão     | <b>0.9816</b> | 0.9653         | 0.9653        |
| <i>Oversample</i> | VGG      | Regressão     | <b>0.9816</b> | 0.9653         | 0.9653        |
| <i>Oversample</i> | VGG      | SVM           | <b>0.9816</b> | 0.9653         | 0.9653        |
| <i>Oversample</i> | VGG      | KNN           | <b>0.9816</b> | 0.9653         | 0.9653        |
| <i>Oversample</i> | Xception | KNN           | 0.9770        | <b>0.9720</b>  | <b>0.9715</b> |
| SMOTE             | VGG      | Regressão     | <b>0.9816</b> | 0.9653         | 0.9653        |
| SMOTE             | VGG      | SVM           | <b>0.9816</b> | 0.9653         | 0.9653        |

# Capítulo 6

## Conclusão e Trabalhos Futuros

Neste trabalho foram realizadas comparações entre diversos modelos para a tarefa de classificação de múltiplas classes em bases de dados desbalanceadas de tomografias computadorizadas do pulmão. Foram utilizados modelos de CNN e modelos híbridos que são compostos de CNNs para extrair características e junto a elas foram utilizados classificadores de aprendizado de máquina tradicionais.

No primeiro experimento definido no Capítulo 4 quatro modelos de CNN foram utilizados e destes a CNN VGG19 com balanceamento de dados utilizando *Oversample* alcançou os melhores resultados, tendo como F1-score 95.37%. Além disso, verificando a matriz de confusão para este modelo, pode-se concluir que o mesmo teve bom desempenho e um baixo número de classificações errôneas comparado aos outros modelos. Por outro lado, as CNNs Xception e DenseNet obtiveram os piores resultados, em que alcançou-se 88.99% na métrica F1-score, resultado esse que reflete em uma maior taxa de classificações errôneas quando comparado à CNN VGG19. Por fim, foi analisado um experimento onde as CNNs com balanceamento de dados foram executadas e, durante o período de treinamento, foi utilizado aumento de dados com o objetivo de variar as amostras de imagens do conjunto de treinamento. Notou-se que o tempo de treinamento por época aumentou de cinco para quinze segundos e, além disso, houve uma piora nos resultados obtidos, uma vez que no melhor caso obteve-se um F1-score de 84.23%.

No segundo experimento proposto, os quatro modelos de CNN foram utilizados juntamente aos métodos tradicionais de classificação. Inicialmente notou-se que, mesmo com classes desbalanceadas, o modelo híbrido de VGG com Regressão Logística apresentou um bom desempenho, alcançando um F1-score de 95.53%, resultado esse que se manteve ao utilizar métodos de balanceamento de classes. Utilizando os métodos SMOTE e *Oversample* a VGG Xception indicou os melhores resultados durante a análise comparativa, no qual se alcançou um F1-score de 96.65% e 97.2% respectivamente.

Os modelos aqui apresentados demonstraram bons desempenhos na tarefa de classificação. No entanto, por conta da baixa quantidade de imagens, a utilização de validação cruzada [44] poderá trazer um maior desempenho e confiabilidade para os modelos.

Para trabalhos futuros a utilização de bases de dados contendo um maior número de imagens de tomografias computadorizadas permitiria a validação do bom desempenho dos modelos utilizando as CNNs VGG e Xception. Além disso, seria possível a utilização de pesos, nos casos onde houveram um maior erro de classificação em determinadas classes, permitindo assim a definição de um maior peso para as classes que menos são classificadas corretamente. Além disso, poderia ser proposto um experimento onde seria feito um ensemble, combinando os modelos VGG19 e modelos híbridos da VGG19 e Xception com classificador de Regressão Logística e todos utilizando balanceamento de classes *Oversample*.

Por fim, objetivando a proposta de um modelo capaz de detectar e classificar tumores malignos, para trabalhos futuros, os modelos aqui citados poderiam ser generalizados, para permitir que, a partir deles, se tornasse possível a classificação das tomografias computadorizadas e em caso de tumor maligno, a classificação entre os tipos: adenocarcinoma, carcinoma de células escamosas e carcinoma de grandes células.

# Referências

- [1] American Cancer Society: *Lung cancer*. <https://www.cancer.org/cancer/lung-cancer>. 1
- [2] Instituto Nacional de Câncer: *Câncer de pulmão*. <https://www.gov.br/inca/pt-br/assuntos/cancer/tipos/pulmao>. 1
- [3] O’Shea, Keiron e Ryan Nash: *An Introduction to Convolutional Neural Networks*. CoRR, abs/1511.08458, 2015. <http://arxiv.org/abs/1511.08458>. 2
- [4] Zhang, Chao, Xing Sun, Kang Dang, Ke Li, Xiao-wei Guo, Jia Chang, Zong-qiao Yu, Fei-yue Huang, Yun-sheng Wu, Zhu Liang, Zai-yi Liu, Xue-gong Zhang, Xing-lin Gao, Shao-hong Huang, Jie Qin, Wei-neng Feng, Tao Zhou, Yan-bin Zhang, Wei-jun Fang, Ming-fang Zhao, Xue-ning Yang, Qing Zhou, Yi-long Wu e Wen-zhao Zhong: *Toward an expert level of lung cancer detection and classification using a deep convolutional neural network*. *The Oncologist*, 24(9):1159–1165, abril 2019, ISSN 1083-7159. <https://doi.org/10.1634/theoncologist.2018-0908>. 2, 24, 27
- [5] Toğaçar, Mesut, Burhan Ergen e Zafer Cömert: *Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks*. *Biocybernetics and Biomedical Engineering*, 40(1):23–39, 2020, ISSN 0208-5216. <https://www.sciencedirect.com/science/article/pii/S0208521619304759>. 2, 25, 27
- [6] Polat, Huseyin e Homay Danaei Mehr: *Classification of Pulmonary CT Images by Using Hybrid 3D-Deep Convolutional Neural Network Architecture*. *Applied Sciences*, 9(5), 2019, ISSN 2076-3417. <https://www.mdpi.com/2076-3417/9/5/940>. 2, 26, 27
- [7] National Institute of Biomedical Imaging and Bioengineering: *Computed Tomography (CT)*. <https://www.nibib.nih.gov/science-education/science-topics/computed-tomography-ct>. Acessado em: 19-11-2022. 4
- [8] Than, Joel C.M., Luca Saba, Norliza M. Noor, Omar M. Rijal, Rosminah M. Kassim, Ashari Yunus, Harman S. Suri, Michele Porcu e Jasjit S. Suri: *Lung disease stratification using amalgamation of Riesz and Gabor transforms in machine learning framework*. *Computers in Biology and Medicine*, 89:197–211, 2017, ISSN 0010-4825. <https://www.sciencedirect.com/science/article/pii/S0010482517302731>. 5
- [9] Mildenerger, Peter, Marco Eichelberg e Eric Martin: *Introduction to the DICOM standard*. *European Radiology*, 12(4):920–927, Apr 2002, ISSN 1432-1084. <https://doi.org/10.1007/s003300101100>. 5

- [10] Carmo, J. S: *Aprendizagem - Contribuições da psicologia*. São Carlos: Universidade Federal de São Carlos, 2012. 6
- [11] Franco, C. R: *Inteligência Artificial*. Londrina: Editora e distribuidora educacional, 1ª edição, 2014. 6
- [12] Russell, Stuart e Peter Norvig: *Artificial Intelligence: A Modern Approach* . 4ª edição, 2019. 7, 9, 10, 11
- [13] Vapnik, Vladimir N.: *Statistical Learning Theory*. Wiley-Interscience, 1998. 8
- [14] Bottou, Léon: *Stochastic Gradient Descent Tricks*, páginas 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, ISBN 978-3-642-35289-8. [https://doi.org/10.1007/978-3-642-35289-8\\_25](https://doi.org/10.1007/978-3-642-35289-8_25). 9
- [15] Yacim, Joseph e Douw Boshoff: *Impact of Artificial Neural Networks Training Algorithms on Accurate Prediction of Property Values*. Journal of Real Estate Research, 40:375–418, novembro 2018. 11
- [16] Haykin, S.: *Redes Neurais: Princípios e Prática*. Bookman Editora, ISBN 9788577800865. 12
- [17] Hubel, D. H. e T. N. Wisel: *Brain and Visual Perception: The Story of a 25-Year Collaboration*. Oxford University Press, 2004, ISBN 9780198039167. 12
- [18] Aggarwal, Charu C.: *Neural Networks and Deep Learning*. Springer, Cham, 2018, ISBN 978-3-319-94462-3. 12, 13, 14, 19
- [19] Ghosh, Anirudha, A. Sufian, Farhana Sultana, Amlan Chakrabarti e Debashis De: *Fundamental Concepts of Convolutional Neural Network*, páginas 519–567. janeiro 2020, ISBN 978-3-030-32643-2. 12, 13, 14, 15, 16
- [20] O’Shea, Keiron e Ryan Nash: *An Introduction to Convolutional Neural Networks*, 2015. <https://arxiv.org/abs/1511.08458>. 13
- [21] Radu, Mihai Daniel, Ilona Madalina Costea e Valentin Alexandru Stan: *Automatic Traffic Sign Recognition Artificial Intelligence - Deep Learning Algorithm*. Em *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, páginas 1–4, 2020. 14
- [22] Leixian, Shen, Qingyun Zhang, Guoxu Cao e He Xu: *Fall Detection System Based on Deep Learning and Image Processing in Cloud Environment*, páginas 590–598. janeiro 2019, ISBN 978-3-319-93658-1. 15
- [23] DeepAI: *What is Max Pooling?* <https://deepai.org/machine-learning-glossary-and-terms/max-pooling>. Acesso em: 03-12-2022. 15
- [24] Simonyan, Karen e Andrew Zisserman: *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2014. <https://arxiv.org/abs/1409.1556>. 16, 35



- [25] Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg e Li Fei-Fei: *ImageNet Large Scale Visual Recognition Challenge*. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015. 16, 18
- [26] Ferguson, Max, Ronay ak, Yung Tsun Lee e Kincho Law: *Automatic localization of casting defects with convolutional neural networks*. páginas 1726–1735, dezembro 2017. 17
- [27] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke e Andrew Rabinovich: *Going Deeper with Convolutions*, 2014. <https://arxiv.org/abs/1409.4842>. 17, 35
- [28] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens e Zbigniew Wojna: *Rethinking the Inception Architecture for Computer Vision*. Em *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 2818–2826, 2016. 17
- [29] Huang, Gao, Zhuang Liu, Laurens van der Maaten e Kilian Q. Weinberger: *Densely Connected Convolutional Networks*, 2016. <https://arxiv.org/abs/1608.06993>. 17, 18, 35
- [30] Chollet, François: *Xception: Deep Learning with Depthwise Separable Convolutions*, 2016. <https://arxiv.org/abs/1610.02357>. 18, 35
- [31] Weiss, Karl, Taghi M. Khoshgoftaar e DingDing Wang: *A survey of transfer learning*. Journal of Big Data, 3(1):9, May 2016, ISSN 2196-1115. <https://doi.org/10.1186/s40537-016-0043-6>. 18
- [32] Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall e W Philip Kegelmeyer: *SMOTE: synthetic minority over-sampling technique*. Journal of artificial intelligence research, 16:321–357, 2002. 19, 20, 32, 38
- [33] Lakshmanaprabu, S.K., Sachi Nandan Mohanty, Shankar K., Arunkumar N. e Gustavo Ramirez: *Optimal deep learning model for classification of lung cancer on CT images*. Future Generation Computer Systems, 92:374–382, 2019, ISSN 0167-739X. <https://www.sciencedirect.com/science/article/pii/S0167739X18317011>. 23, 27
- [34] Song, QingZeng, Lei Zhao, XingKe Luo e XueChen Dou: *Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images*. Journal of Healthcare Engineering, 2017. <https://doi.org/10.1155/2017/8314740>. 23, 27
- [35] Schwyzer, M, Ferraro D A, Muehlematter U J, Curioni Fontecedro A, Huellner MW, von Schulthess GK, Kaufmann PA, Burger IA e Messerli M: *Automated detection of lung cancer at ultralow dose PET/CT by deep neural networks - Initial results*. Lung Cancer, 2018. 24, 27

- [36] Shakeel, P. Mohamed, M.A. Burhanuddin e Mohamad Ishak Desa: *Lung cancer detection from CT image using improved profuse clustering and deep learning instantaneously trained neural networks*. Measurement, 145:702–712, 2019. <https://www.sciencedirect.com/science/article/pii/S0263224119304439>. 24, 27
- [37] Xiao, Zhitao, Bowen Liu, Lei Geng, Fang Zhang e Yanbei Liu: *Segmentation of Lung Nodules Using Improved 3D-UNet Neural Network*. Symmetry, 12(11), 2020. <https://www.mdpi.com/2073-8994/12/11/1787>. 24, 27
- [38] Jung, Hwejin, Bumssoo Kim, Inyeop Lee, Junhyun Lee e Jaewoo Kang: *Classification of lung nodules in CT scans using three-dimensional deep convolutional neural networks with a checkpoint ensemble method*. BMC Medical Imaging, 18, 2018. <https://doi.org/10.1186/s12880-018-0286-0>. 25, 27
- [39] Jakimovski, Goran e Danco Davcev: *Using Double Convolution Neural Network for Lung Cancer Stage Detection*. Applied Sciences, 9(3), 2019, ISSN 2076-3417. <https://www.mdpi.com/2076-3417/9/3/427>. 25, 27
- [40] Khan, Muhammad Attique, Venkatesan Rajinikanth, Suresh Chandra Satapathy, David Taniar, Jnyana Ranjan Mohanty, Usman Tariq e Robertas Damaševičius: *VGG19 Network Assisted Joint Segmentation and Classification of Lung Nodules in CT Images*. Diagnostics, 11(12), 2021, ISSN 2075-4418. <https://www.mdpi.com/2075-4418/11/12/2208>. 26, 27
- [41] Alyasriy, Hamdalla e Muayed AL-Huseiny: *The IQ-OTHNCCD lung cancer dataset*. Mendeley Data, 2020. <https://www.kaggle.com/hamdallak/the-iqothnccd-lung-cancer-dataset/metadata>. 29
- [42] Deng, Jia, Wei Dong, Richard Socher, Li Jia Li, Kai Li e Li Fei-Fei: *Imagenet: A large-scale hierarchical image database*. Em *2009 IEEE conference on computer vision and pattern recognition*, páginas 248–255. Ieee, 2009. 31
- [43] Lin, Tsung Yi, Priya Goyal, Ross Girshick, Kaiming He e Piotr Dollár: *Focal Loss for Dense Object Detection*, 2017. <https://arxiv.org/abs/1708.02002>. 36
- [44] Wong, Tzu Tsung: *Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation*. Pattern Recognition, 48(9):2839–2846, 2015, ISSN 0031-3203. <https://www.sciencedirect.com/science/article/pii/S0031320315000989>. 65