



Sistema de Detecção e Prevenção de Doenças Pulmonares

DIOGO FILIPE MEIRELES FORMOSINHO

Julho de 2023



Sistema de Detecção e Prevenção de Doenças Pulmonares

Diogo Filipe Meireles Formosinho

Aluno nº: 1210056

**Dissertação para obtenção do Grau de
Mestre em Engenharia de Inteligência Artificial**

Orientador: Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Júri:

Presidente:

Doutora Maria Goreti Carvalho Marreiros, Professora Coordenadora com Agregação do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Vogais:

Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Doutor Hugo Daniel Abreu Peixoto, Investigador Auxiliar da Universidade do Minho

Porto, julho 2023

Resumo

A análise de imagens médicas desempenha um papel fundamental no diagnóstico e tratamento de diversas condições de saúde. Entre as várias modalidades de imagens médicas, os raios X são amplamente utilizados devido à sua praticidade, baixo custo e capacidade de fornecer informações valiosas sobre o estado de saúde de um paciente. No entanto, a interpretação precisa e eficiente das radiografias requer conhecimento especializado e experiência clínica, o que pode ser limitado em determinados contextos.

Neste contexto, o desenvolvimento de modelos capazes de analisar e interpretar radiografias de forma automatizada e precisa tem se tornado uma área de pesquisa promissora. Estes modelos têm o potencial de auxiliar radiologistas e médicos no diagnóstico precoce e no tratamento de diversas doenças e condições médicas.

Existem algumas razões pelas quais o desenvolvimento de um modelo de análise automatizado de raios X é de grande relevância. A necessidade de resposta por serviços de saúde tem aumentado, e o número de radiografias a serem analisadas pode sobrecarregar os profissionais de saúde. Portanto, ter um modelo automatizado pode acelerar o processo de triagem e diagnóstico, permitindo um atendimento mais eficiente e reduzindo o tempo de espera dos pacientes.

Foi desenvolvido um modelo de análise automatizado que é capaz de identificar e diagnosticar a presença de problemas e anomalias nas imagens radiográficas. O modelo utiliza técnicas de *deep learning*, como redes neurais convolucionais, para extrair características relevantes das radiografias e realizar a classificação adequada. Na construção deste modelo também foi utilizado técnicas de pré processamento de imagem, para preparar todas as imagens que foram usadas no treino deste mesmo modelo, permitindo a melhor análise possível da mesma. No fim, o modelo foi capaz de reconhecer 14 doenças devido ao conjunto de dados usado no treino.

Apesar de reconhecer 14 padrões (doenças) diferentes, depois de uma avaliação do modelo, foi perceptível que o mesmo sente dificuldades em avaliar alguns destes padrões e retornar uma resposta válida, em contrapartida, tem uma boa *performance* na análise dos restantes casos principalmente quando a radiografia não apresenta problemas.

A aplicação desenvolvida pretende atender aos critérios de desempenho e precisão necessários para torná-la uma ferramenta útil na prática clínica, sempre considerando a importância de uma análise final por parte de um radiologista ou médico especializado. É fundamental realçar que o modelo não substituirá a especialização clínica, mas sim servirá como uma valiosa ferramenta complementar para a tomada de decisão médica, possibilitando assim uma análise mais ágil.

Palavras-chave: Análise de imagens médicas, Doenças, Redes Neurais Convolucionais, Classificação

Abstract

Medical imaging analysis plays a key role in the diagnosis and treatment of various health conditions. Among the various medical imaging modalities, X-rays are widely used due to their practicality, low cost and ability to provide valuable information about a patient's health status. However, accurate and efficient interpretation of x rays requires specialized knowledge and clinical experience, which may be limited in certain contexts.

In this context, the development of models capable of analyzing and interpreting x rays in an automated and accurate manner has become a promising area of research. These models have the potential to assist radiologists and medics with the early diagnosis and treatment of various diseases and medical conditions.

There are a few reasons why the development of an automated X-ray analysis model is relevant. Firstly, the need for fast response by healthcare services, the number of x rays to be analyzed can overwhelm healthcare professionals. Therefore, having an automated model can speed up the diagnostic process, increasing the efficient care and reducing the patient waiting time.

An automated analysis model was developed that can identify and diagnose the presence of problems and anomalies in x ray images. The model uses deep learning techniques, such as convolutional neural networks, to extract relevant features from the x rays and perform the appropriate classification. In the construction of this model, image pre-processing techniques were also used, to prepare all the images that were used in the training of this same model, allowing the best possible analysis. In the end, the model was able to recognize 14 diseases due to the dataset used in the training.

Despite recognizing 14 different patterns (diseases), after an evaluation of the model, it was perceptible that it has difficulties in evaluating some of these patterns and returning a valid answer, on the other hand, it has a good performance in the analysis of the remaining cases, especially when the x ray does not contain any problem.

The application developed intends to meet the performance and accuracy criteria necessary to make it a useful tool in clinical practice, always considering the importance of a final analysis by a radiologist or specialized medic. It is essential to highlight that the model will not replace clinical expertise, but will serve as a valuable complementary tool for medical decision making, thus enabling a more agile analysis.

Keywords: Medical Imaging Analysis, Diseases, Convolutional neuronal networks, Classification.

Agradecimentos

Em primeiro lugar, quero agradecer a toda a gente que de alguma forma esteve envolvida neste projeto e fez com que o mesmo fosse possível.

Ao meu orientador, Professor Doutor Constantino Martins, por toda a ajuda, apoio e disponibilidade prestada durante o desenvolvimento do projeto.

Ao Instituto Superior de Engenharia do Porto por me ter acolhido nestes dois últimos anos académicos e a todos os docentes que sempre mostraram muita disponibilidade.

Agradeço à minha família, e em especial aos meus pais, que sempre me acompanharam durante todo o meu percurso, por todo o apoio incondicional e sacrifícios que permitiram que eu alcançasse os meus objetivos académicos.

Por fim, mas não menos importante, gostaria de agradecer à minha namorada e amigos por todas as sugestões dadas e por toda a disponibilidade que sempre demonstraram.

Índice

1	Introdução	1
1.1	Problema	1
1.2	Objetivos	1
1.3	Motivação	2
1.4	Contributos	2
1.5	Metodologia de Investigação	3
1.6	Estrutura do Documento	5
2	Estado da Arte e Formalização Teórica	6
2.1	Metodologias de Pesquisa	6
2.1.1	Questões de Pesquisa e Critérios Definidos	7
2.1.2	Critérios de Pesquisa e Elegibilidade	8
2.1.3	Resultados da pesquisa	8
2.1.4	Questão de pesquisa nº1: Quais são os tipos de diagnóstico mais usados e fáceis para análise?	9
2.1.5	Questão de pesquisa nº2: Quais são as técnicas mais usadas nas soluções existentes e porquê?	9
2.1.6	Questão de pesquisa nº3: Quais são os resultados esperados das soluções e problemas encontrados?	10
2.2	Tipos de Imagens Médicas	10
2.2.1	Imagem de Raio X	11
2.2.2	Ressonância Magnética (MRI)	12
2.2.3	Imagens de Ultrassom	12
2.2.4	Tomografia Computorizada (TAC)	13
2.2.5	Medicina Nuclear	14
2.2.6	Tipo de Imagem Médica Usada	14
2.3	Técnicas e Métodos de Tratamento de Imagem Médica	15
2.3.1	Pré-Processamento e Segmentação de Imagem	16
2.4	Redes Neurais	18
2.5	Modelo Computacional	21
2.6	Contributos da IA na Medicina	23
2.6.1	Aplicações na área de Medicina	24
2.7	Soluções Existentes	25
2.8	Ética e Proteção	27
2.9	Resumo/Conclusão	28
3	Análise e Design	30
3.1	Análise de Requisitos	30
3.1.1	Atores Principais	30

3.1.2	Levantamento de Requisitos	30
3.2	Dataset	32
3.2.1	Fundamento e Motivação	32
3.2.2	Detalhes	32
3.2.3	Limitações	34
3.3	Método Proposto.....	35
3.4	Design	36
3.5	Resumo/Conclusão	36
4	Implementação	38
4.1	Preparação e Análise do Dataset	38
4.2	Ambiente de Desenvolvimento.....	39
4.3	Implementação do Modelo	39
4.3.1	Resultados do Treino	46
4.4	Implementação do Modelo (2º Versão).....	47
4.4.1	Resultados do Treino (2ª Versão)	48
4.5	Implementação da Interface Gráfica.....	48
4.5.1	Resultados da Interface Gráfica	56
4.6	Resumo/Conclusão	58
5	Avaliação e Experimentação	60
5.1	Avaliação e Comparação de Resultados	60
5.2	Experimentação	62
5.2.1	Procedimento do Modelo.....	62
5.2.2	Casos de Teste	64
5.3	Resumo/Conclusão	69
6	Conclusão.....	70
6.1	Objetivos Alcançados	71
6.2	Problemas Encontrados.....	71
6.3	Futuro Trabalho	72
6.3.1	Modelo	72
6.3.2	Interface Gráfica (UI).....	72

Lista de Figuras

Figura 1 - Etapas da metodologia Design Science Research	4
Figura 2 - Exemplo de uma imagem raio X	11
Figura 3 - Exemplo de uma ressonância magnética.....	12
Figura 4 - Exemplo de uma imagem ultrassom.....	13
Figura 5 - Exemplo de uma tomografia computadorizada	13
Figura 6 - Exemplo de uma imagem com suporte da medicina nuclear.....	14
Figura 7 - Exemplo de uma Rede Neuronal Artificial	21
Figura 8 - Performance e Resultados dos Sistemas	26
Figura 9 - Distribuição das categorias de doenças	34
Figura 10 - Método Proposto.....	35
Figura 11 - Desenho do processo da Interface Gráfica	36
Figura 12 - Resultados do treino na 1º epoch.....	46
Figura 13 - Resultados do treino na 50º epoch.....	46
Figura 14 - Resultados do treino na última epoch	46
Figura 15 - Análise do Modelo	47
Figura 16 - Análise do modelo (2ª Versão)	48
Figura 17 - Menu Inicial (UI).....	56
Figura 18 - Menu de Análise de Imagem (UI).....	57
Figura 19 - Menu de Análise de Imagem (UI).....	57
Figura 20 - Resultados da accuracy dos modelos	61
Figura 21 - Accuracy dos modelos pré treinados para algumas doenças	61
Figura 22 - Imagem para Avaliação	62
Figura 23 - Probabilidade de Cada Doença	62
Figura 24 - Ordem de doenças	63
Figura 25 - Resposta do Modelo	63
Figura 26 - Experimentação de Imagem (Menu Inicial)	64
Figura 27 - Experimentação de Imagem (Selecionar Imagem)	65
Figura 28 - Experimentação de Imagem (Processamento)	65
Figura 29 - Experimentação de Imagem (Resultado).....	66
Figura 30 - Experimentação de Imagem (Segundo Caso)	66
Figura 31 - Análise de Dataset (Menu Inicial)	67
Figura 32 - Análise de Dataset (Selecionar Ficheiro e Pasta de Imagens).....	68
Figura 33 - Análise de Dataset (Processamento)	68
Figura 34 - Análise de Dataset (Resultado)	69

Lista de Tabelas

Tabela 1 – Resultados de Treino	60
---------------------------------------	----

Acrónimos e Símbolos

AI/IA	Inteligência Artificial
ML	<i>Machine Learning</i>
DL	<i>Deep Learning</i>
DSR	<i>Design Science Research</i>
PRISMA	<i>Preferred Reporting Items for Systematic Reviews and Meta-Analyses</i>
TAC	Tomografia computadorizada
MRI	Ressonância magnética
RNA	Rede Neuronal Artificial
GAN	<i>Generative Adversarial Networks</i>
CNN	Rede Neuronal Convolutacional
IDE	Ambiente de Desenvolvimento Integrado
UI/GUI	Interface Gráfica
OMS	Organização Mundial de Saúde
SFTP	<i>Secured File Transfer Protocol</i>
IP	<i>Internet Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>

1 Introdução

Este capítulo apresenta uma breve introdução ao projeto que será desenvolvido. Inicialmente é descrito um pouco o problema em questão, de seguida apresenta os objetivos do trabalho e vantagens. São também expostas as motivações para o desenvolvimento do mesmo tal como os contributos de Inteligência Artificial para as outras áreas e acima de tudo para a área em questão, a saúde. O capítulo acaba com a descrição da metodologia usada para investigação e a respetiva estrutura do documento.

1.1 Problema

A gripe é a principal doença que mantém os hospitais preenchidos durante maioritariamente o ano todo, especialmente na altura do Inverno, assim como o covid, que nos dias de hoje desempenha um grande papel neste cenário bacteriano (*Portal Do INE, n.d.*). Ambas estas doenças atacam as vias respiratórias e os pulmões e podem provocar danos nas mesmas se não forem tratadas da maneira correta. Para além destas temos outras doenças no mesmo cenário como a bronquite, pneumonia entre outras. No último ano através do portal do Instituto Nacional de Estatística, consegue-se perceber que o covid foi a razão de 5,8% dos óbitos totais, para além disso, 9,1% dos óbitos foram devido a doenças do aparelho respiratório (*Portal Do INE, n.d.*).

1.2 Objetivos

O objetivo principal é implementar um sistema de prevenção que permite detetar infeções ou doenças pulmonares através da análise do raio x dos pulmões. Dessa forma permitirá aos profissionais de saúde ter uma melhor análise de cada caso sendo que o sistema servirá de suporte e ajudará na avaliação deste tipo de tratamentos.

O sistema para além do suporte nesta vertente irá ajudar os profissionais a perceber qual doença o paciente possa apresentar para que posteriormente estes mesmos profissionais definam qual será o acompanhamento mais adequado de cada caso para aumentar as probabilidades de o mesmo tratamento ser feito de forma correta e bem-sucedida.

A questão de investigação é a seguinte, **podem as técnicas de *deep learning* em imagens de raio x ajudar na deteção preventiva de doenças pulmonares e na tomada de decisão por parte dos profissionais de saúde?**

1.3 Motivação

Desenvolver um sistema de deteção na área da saúde é fundamental para ajudar na eficácia e eficiência do diagnóstico e tratamento de doenças. Com o avanço da tecnologia, é possível criar sistemas capazes de detetar precocemente sinais de doenças, o que aumenta as chances de cura e conseqüentemente melhora a qualidade de vida das pessoas. Além disso, a automatização dos processos de deteção pode libertar tempo e recursos aos profissionais de saúde, permitindo que estes se dediquem a outras tarefas importantes.

Outra motivação importante é a possibilidade de alcançar uma maior precisão e confiabilidade nos resultados obtidos. Sistemas de deteção baseados em inteligência artificial e aprendizagem automática são capazes de processar grandes volumes de dados e identificar padrões complexos.

Os sistemas de deteção podem ser utilizados para monitorizar a evolução de doenças crónicas e acompanhar a eficácia do tratamento, o que é essencial para garantir a qualidade e a segurança dos pacientes.

Em resumo, a motivação para desenvolver sistemas de deteção na área da saúde vem da necessidade de melhorar a qualidade e a eficácia do diagnóstico e tratamento de doenças, pois para além de alcançar uma maior precisão e confiabilidade nos resultados, é uma forma de contribuir para a saúde e bem-estar das pessoas.

1.4 Contributos

A área de deteção e prevenção contribui em todos os aspetos de várias maneiras, algumas destas são (*Efeitos Da Inteligência Artificial Na Medicina | APD, n.d.*):

- Melhoria da precisão: Sistemas de deteção e prevenção utilizam técnicas de *machine learning* para detetar comportamentos anómalos ou padrões que indicam uma ameaça ou problema. Isto pode ajudar a aumentar a precisão das deteções e reduzir os falsos positivos;
- Automatização: Sistemas de deteção e prevenção podem ser projetados para tomar medidas automatizadas para evitar que uma ameaça ou problema cause danos, como

bloquear automaticamente um endereço IP suspeito ou interromper uma transação fraudulenta;

- Análise de grandes conjuntos de dados: Sistemas de detecção e prevenção usam técnicas avançadas de *machine learning* para analisar grandes volumes de dados e identificar padrões e tendências emergentes, isto ajuda a perceber melhor os dados e como eles podem ser usados para melhorar a segurança;
- Aplicações em vários setores: A detecção e prevenção são importantes em vários setores, como ciber segurança, finanças, saúde e transporte. Isto ajuda a expandir e melhorar as aplicações de inteligência artificial e a torná-las mais acessíveis e úteis em vários campos;
- Melhoria da segurança: Sistemas de detecção e prevenção podem ser projetados para identificar e prevenir ameaças e problemas de segurança, como ataques cibernéticos, fraudes, e outras formas de má utilização de dados. Isto pode ajudar a proteger os sistemas e os dados dos utilizadores.

Neste projeto em questão será desenvolvido um sistema de detecção e prevenção como ferramenta de suporte para ajudar os profissionais de saúde na avaliação dos casos dos utentes para que haja uma melhor precisão na mesma.

O modelo desenvolvido tem de ser capaz de detetar anomalias em imagens raios x, posteriormente, classificando-as da forma correta com o nome da doença encontrada na análise. Para além disso, tem de apresentar uma mensagem caso através de algum tipo de índice, seja possível perceber que uma doença, apesar de não estar presente, possa ser um potencial problema futuro. Desta forma o sistema terá de contribuir para a análise de casos e ajudar como ferramenta de suporte os profissionais de saúde.

1.5 Metodologia de Investigação

Depois de uma revisão de literatura sobre metodologias de pesquisa, *Design Science Research* (DSR) foi identificada como a que melhor se adequa a este projeto. A metodologia DSR adequa-se a projetos relacionados com a saúde porque é uma abordagem interdisciplinar que permite a integração de conhecimentos de diferentes áreas, permite trabalhar com vários atores, e enfatiza a importância de avaliar as soluções desenvolvidas em situações reais. Isso garante que as soluções sejam relevantes e úteis (Horváth, 2007).

Design Science Research (DSR) é uma metodologia para conduzir pesquisas que é usada no campo de sistemas de informação e tecnologia. O objetivo do DSR é criar conhecimento projetando, desenvolvendo e avaliando novos produtos que resolvam problemas do mundo real (Pello, 2018).

A metodologia DSR normalmente segue várias etapas e para este projeto serão as seguintes (Zambalde et al., n.d.):

- Numa fase inicial, esta investigação dirige-se à análise da possibilidade da aplicação de um sistema de deteção na avaliação de radiografias;
- A segunda etapa compreende a definição dos objetivos da solução, assim como a abordagem do sistema de detenção;
- Na etapa três, a solução de software foi projetada e o design da mesma foi definida;
- A quarta etapa compreendeu as experimentações iniciais dos dados recolhidos e foram produzidos os primeiros resultados da análise;
- Na quinta etapa, o sistema foi testado, validando os resultados que foram produzidos.

É importante observar que a metodologia DSR é iterativa por natureza, com cada etapa sendo construída sobre a anterior. Isso significa que possa ser necessário voltar às etapas anteriores e fazer ajustes com base nos resultados das etapas posteriores (Horváth, 2007). Além disso, é comum usar vários métodos de pesquisa ao longo do processo de design, incluindo métodos quantitativos e qualitativos (Figura 1) (Pacheco Lacerda et al., n.d.).

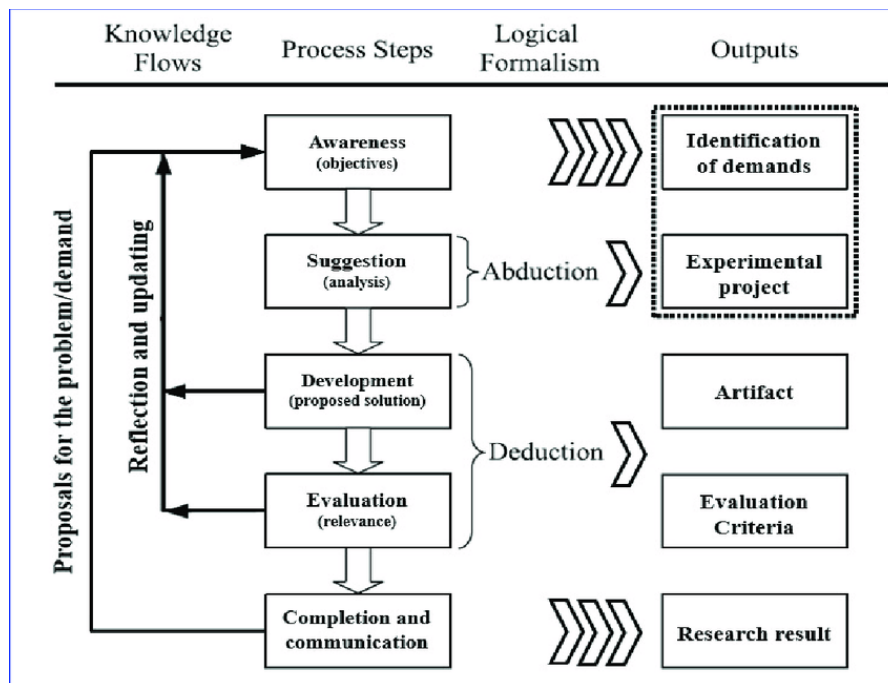


Figura 1 – Etapas da metodologia Design Science Research (Zambalde et al., n.d.)

1.6 Estrutura do Documento

Esta dissertação foi dividida em 6 capítulos, cujo conteúdo é descrito a seguir:

Capítulo 1 – Neste primeiro capítulo é apresentada uma descrição do contexto e o problema e, de seguida, apresentado os objetivos. É também desenvolvida um pouco a motivação e os contributos da área e do projeto e, por fim é apresentada a metodologia de investigação usada.

Capítulo 2 – No segundo capítulo é apresentado o estado da arte. Neste capítulo é descrito a metodologia de pesquisa que foi utilizada, é apresentado os conceitos, algoritmos e técnicas de soluções existentes, para além disso é apresentado soluções já existentes na área. Por fim temos um subcapítulo sobre ética e proteção de dados e a conclusão do estado da arte.

Capítulo 3 – No terceiro capítulo é feita a análise e *design* da solução. Neste capítulo é revisto o levantamento de requisitos para a construção do projeto em questão.

Capítulo 4 – Neste quarto capítulo é descrita a implementação do projeto e é feito uma explicação do código usado. É explicado todo o processo desde o início até ao fim da construção do sistema.

Capítulo 5 – No quinto capítulo é apresentada a avaliação e experimentação do sistema. Este é apresentado como um produto final e é feito alguns testes para comprovar que o mesmo é válido e atinge os resultados esperados. Também é feita a comparação do mesmo com outros modelos idênticos.

Capítulo 6 – Neste último capítulo é apresentada a conclusão da tese. É descrito um resumo de todo o processo desde o início ao fim do projeto, problemas encontrados e possíveis melhorias para futuras versões do mesmo.

2 Estado da Arte e Formalização Teórica

Este capítulo apresenta uma revisão da literatura sobre os temas fundamentais em que se baseia este trabalho de investigação.

Os problemas pulmonares distinguem-se por apresentar uma das maiores incidências e ter uma das mais taxas de mortalidade entre todos os outros tipos de problemas. Detetando e tratando estes problemas nos estágios iniciais fornece aos pacientes uma probabilidade baixa de futuras repercussões. Assim, esta revisão apresenta que métodos são usados para melhorar a eficiência do diagnóstico por parte destas soluções. Inicialmente é definida qual será a metodologia de pesquisa a ser usada, mencionando as questões e critérios de pesquisa para a obtenção de artigos. É apresentada os tipos de imagem médica e que técnicas e métodos existem no tratamento das mesmas. É abordado os diferentes tipos de redes neuronais e alguns exemplos de uso das mesmas, e ainda é definida qual será a usada para a solução. É retratado alguns contributos de IA na medicina e que soluções já existem neste âmbito. Por fim, é abordada a ética e a proteção de dados.

2.1 Metodologias de Pesquisa

O objetivo do *PRISMA* é garantir que as revisões sistemáticas e meta-análises sejam realizadas de maneira transparente e completa, de modo a permitir a avaliação crítica e a interpretação dos resultados pelos leitores (*Filho, 2020; Moher et al., 2010*).

Este projeto tem como objetivo utilizar tecnologias *AI/ML* para diagnosticar e detetar diferentes infeções através da análise de raios x, portanto, foi usado uma adaptação simplificada do método *PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses)* (*Moher et al., 2010*) para realizar as pesquisas e selecionar as informações mais relevantes para o projeto em questão.

O método *PRISMA* inclui os seguintes passos (*Rethlefsen & Page, 2022*):

- Definir claramente o objetivo da revisão e os critérios de elegibilidade para incluir estudos na revisão;
- Realizar uma pesquisa extensiva em várias bases de dados e fontes para identificar todos os estudos relevantes;
- Avaliar a qualidade dos estudos incluídos na revisão, usando os critérios estabelecidos anteriormente;
- Extrair os dados relevantes dos estudos incluídos na revisão de maneira consistente e sistemática;

- Analisar os dados extraídos para responder às perguntas da pesquisa estabelecidas no início da revisão;
- Avaliar os resultados da revisão de maneira clara e transparente, seguindo as diretrizes do *PRISMA* para garantir a qualidade e a confiabilidade da revisão.

O *PRISMA* é amplamente utilizado em revisões sistemáticas e meta-análises em várias áreas da ciência, incluindo a saúde, a psicologia e a economia. As diretrizes do *PRISMA* são atualizadas periodicamente para refletir o estado atual do conhecimento e os padrões de relatório mais recentes (Moher et al., 2010; Page et al., 2021).

2.1.1 Questões de Pesquisa e Critérios Definidos

Primeiramente foi definido o objetivo da pesquisa que passa por perceber o que existe atualmente na literatura que permite a análise de imagens para a detecção de doenças. Também é importante conhecer quais os tipos de diagnóstico e quais as técnicas usadas neste âmbito que permitem desenvolver a solução pretendida e, por fim, perceber se os resultados das soluções existentes podem ser considerados bons ou maus. Visto isto, foram então definidas as seguintes questões de pesquisa.

- > Quais são os tipos de diagnóstico mais usados e fáceis para análise?
- > Quais são as técnicas mais usadas nas soluções existentes e porquê?
- > Quais são os resultados esperados das soluções e problemas encontrados?

Depois de definidas as questões de pesquisa, foram definidas algumas fontes e repositórios de dados que serão utilizadas. Para este projeto foram considerados cinco repositórios:

- [IEEE Xplore](#): Consiste numa biblioteca digital e plataforma de pesquisa que contém uma vasta coleção de artigos técnicos, académicos e padrões relacionados a engenharias, ciências, tecnologias, entre outros. É mantido pela IEEE (*Institute of Electrical and Electronics Engineers*), uma organização profissional dedicada ao avanço da tecnologia;
- [PubMed](#): Trata-se de uma base de dados de acesso gratuito mantido pela *National Library of Medicine*. É especializado em publicações científicas nas áreas de medicina, biomedicina, saúde pública e campos relacionados;
- [SpringerLink](#): É uma plataforma online que oferece acesso a uma ampla gama de recursos académicos, incluindo livros, artigos e muito mais. É mantido pela *Springer*, conhecida pelas suas publicações de trabalhos científicos, técnicos e médicos;

- [ResearchGate](#): É uma plataforma online projetada especificamente para a comunidade acadêmica e científica e permite aos pesquisadores e cientistas compartilharem pesquisas e trabalhos.
- [BMC Medical Imaging](#): É uma das várias revistas publicadas pela BioMed Central, uma editora dedicada à divulgação de pesquisas científicas.

2.1.2 Critérios de Pesquisa e Elegibilidade

Para além disso, de forma a obter a máxima qualidade de toda a informação recolhida, foram definidos alguns critérios na pesquisa. Um destes critérios foi que, para um artigo ser válido na recolha de informação teria de ter no máximo 13 anos. Também foi usado o critério de que todos os artigos teriam de estar em inglês para uma melhor leitura da informação que o mesmo disponibilizava sendo que artigos duplicados eram automaticamente excluídos. Por fim, foram definidas algumas palavras-chaves na pesquisa de artigos como, *Artificial Neural Network*, *Convolutional Neural Network*, *Region Growing Segmentation* e *Lung problems detection*.

Inicialmente foi feito a validação de idioma do artigo, de forma a perceber se este encontrava-se na língua pretendida. Após este passo, foi verificado o ano do mesmo, para perceber se não tinha mais que 13 anos. Desta forma, era aprovado os artigos que passavam a fase de elegibilidade. Nesta fase, os critérios das palavras-chaves foram aplicados e foram excluídos os devidos artigos. Por fim, os que previamente cumpriram todos os requisitos, foram incluídos na avaliação caso não fossem duplicados.

2.1.3 Resultados da pesquisa

Depois da leitura de 49 artigos foram selecionados 16 que atendem a todos os critérios definidos anteriormente. Estes 16 artigos abordam os tipos de diagnósticos e as suas vantagens, que métodos e técnicas costumam ser aplicados nos modelos e por fim que resultados esperar neste tipo de soluções.

Foram então excluídos 33 artigos que não corresponderam a todos os critérios definidos e usados os repositórios acima indicados para obter os 16 artigos de forma a responder às questões de pesquisa.

2.1.4 Questão de pesquisa nº1: Quais são os tipos de diagnóstico mais usados e fáceis para análise?

Na primeira questão são analisados os tipos de diagnósticos mais usados, para identificar qual será o melhor para a solução. Esta questão pode ser respondida pelos seguintes artigos “*A Comparative Study of Medical Imaging Techniques*”, “*Modern Diagnostic Imaging Technique Applications and Risk Factors in the Medical Field: A Review*”, “*A survey of medical image classification techniques*”, “*Automated medical image segmentation techniques*” (Hussain et al., 2022; Kasban et al., 2015; Miranda et al., 2017; N. Sharma et al., 2010), todos os artigos abordam alguns tipos de diagnóstico de imagem e quais são as suas principais vantagens e áreas de uso.

Entre os tipos mais comuns estão os raios x, que permitem detetar fraturas ósseas e anomalias nos pulmões, a ressonância magnética, capaz de produzir imagens em diferentes planos e fornecer informações detalhadas sobre os órgãos, ultrassom, usado nas ecografias, tomografia computadorizada, que oferece imagens transversais, permitindo a visualização da estrutura interna de forma precisa e a medicina nuclear, que permite obter informações fisiológicas e moleculares (Hussain et al., 2022; Kasban et al., 2015; Miranda et al., 2017; N. Sharma et al., 2010).

2.1.5 Questão de pesquisa nº2: Quais são as técnicas mais usadas nas soluções existentes e porquê?

A segunda questão corresponde às técnicas e métodos usados para este tipo de análise, tendo em conta todo o processo necessário. Para responder a esta questão foi usado o seguinte livro e artigos, “*Data classification: Algorithms and applications*”, “*Weighing features of lung and heart regions for thoracic disease classification*”, “*Deep Learning in Medical Image Analysis*”, “*Neural Networks and Deep Learning*”, “*Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*”, “*Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges*”, “*A Survey on Segmentation Clustering techniques to detect brain tumor for MRI*”, “*Classification of plant leaf diseases using machine learning and image preprocessing techniques*”, “*Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019*”, “*A survey on image preprocessing techniques for diverse fields of medical imagery*” (Aggarwal, 2018; Aggarwal et al., 2014; Chan et al., 2020; Fang et al., 2021; Goos et al., n.d.; Hesamian et al., 2019; Rajeshwari & Shyamala, 2021; P. Sharma et al., 2020; Tiwari et al., 2020; Vasuki et al., 2017), todos estes explicam os diferentes tipos de redes neuronais e como cada uma destas funciona, para além disso abordam também alguns procedimentos no processamento de imagem.

Para o tipo de projeto em questão, as técnicas que costumam estar mais presentes são:

- Pré-processamento da imagem: Envolve a melhoria e qualidade da imagem através da eliminação do ruído e propriedades indesejadas. Pode incluir operações como, redimensionamento, filtragem, correção de contraste, entre outros;
- Segmentação: É o processo de dividir a imagem em regiões ou objetos. Pode ser feita por métodos como, *thresholding*, segmentação e crescimento de região;
- Classificação e Reconhecimento: Envolve a classificação das imagens com base nas características extraídas. Podem ser aplicados algoritmos como, árvores de decisão, vetores de suporte, redes neuronais, entre outros.

2.1.6 Questão de pesquisa nº3: Quais são os resultados esperados das soluções e problemas encontrados?

Por último são vistos os resultados esperados destas soluções e possíveis problemas durante a sua construção. Foram utilizados os artigos, *“Using artificial intelligence to read chest radiographs for tuberculosis detection: A multi-site evaluation of the diagnostic accuracy of three deep learning systems”*, *“AI-driven deep CNN approach for multi-label pathology classification using chest X-Rays”* (Albahli et al., 2021; Qin et al., 2019), para obter alguns resultados de modelos pré treinados na classificação e análise de imagens.

Foram encontrados alguns sistemas que servem de suporte na deteção da tuberculose, estes apresentaram uma boa análise e classificação, e mostraram resultados positivos na avaliação de radiografias. É ainda descrito a comparação entre os mesmos, para perceber qual desempenha da melhor forma este papel de investigação. Para além destes, foi ainda visto, alguns sistemas que fazem a deteção de múltiplas doenças, tornando-as em soluções de classificação múltipla.

2.2 Tipos de Imagens Médicas

A imagiologia médica, no contexto clínico, refere-se a técnicas e processos utilizados para a obtenção de imagens de um corpo para fins diagnósticos (Alexandre et al., 2011) e desempenha um papel fundamental nos cuidados de saúde atuais.

O diagnóstico de imagem permite aos médicos e outros profissionais procurar possíveis indicadores sobre uma condição de saúde e o tipo de imagens utilizadas vai sempre depender dos sintomas de cada paciente e da parte do corpo que está a ser examinada (*Diagnostic Imaging: MedlinePlus, n.d.; Hussain et al., 2022; Kasban et al., 2015*).

Estas podem incluir:

1. Imagem de Raio X;
2. Ressonância magnética (MRI);

3. Imagens de ultrassom;
4. Tomografia computadorizada (TAC);
5. Medicina Nuclear.

Nestas técnicas radiológicas mencionadas, a energia utilizada para gerar uma imagem terá de se mostrar capaz de penetrar nos tecidos. Na radiologia de diagnóstico, a energia do espectro eletromagnético encontra-se fora da região da luz visível. A radiação ionizante compreende modalidades como a radiografia, tomografia convencional e computadorizada. Outras regiões do espectro eletromagnético são utilizadas na imagiologia, nomeadamente radiação gama, para medicina nuclear e energia mecânica, na forma de ondas de som a altas frequências para a ecografia. À exceção da medicina nuclear, cujos agentes radioativos são injetados ou ingeridos pelo paciente, todas as outras técnicas exigem que a energia utilizada para penetrar nos tecidos interaja com esses mesmos tecidos (Alves et al., 2012).

2.2.1 Imagem de Raio X

Os raios x são uma forma de radiação eletromagnética que pode penetrar os tecidos moles e ossos, produzindo imagens do interior do corpo.

A imagem por raios x é a técnica imagiológica mais comum e amplamente utilizada na medicina e, apesar da observação mais usual passar pela observação das estruturas ósseas (Figura 2) (Hussain et al., 2022; X-Rays: MedlinePlus, n.d.), esta técnica também é utilizada para detetar fraturas, infeções, malformações e lesões tumorais e traumáticas e avaliar o aparelho cardiorrespiratório e génito-urinário (Alexandre et al., 2011).



Figura 2 – Exemplo de uma imagem raio X (O Que São Os Raios X? - Brasil Escola, n.d.)

2.2.2 Ressonância Magnética (MRI)

A ressonância magnética baseia-se nas propriedades magnéticas de um átomo pelo que consegue detetar, analisar e medir os sinais provenientes dos núcleos atômicos em resposta a ondas de rádio com a mesma frequência desses núcleos (Ferreira, 2009; Soares, 2016).

Desde modo consegue criar imagens detalhadas de órgãos e de tecidos dentro do corpo, através da utilização de poderosos campos magnéticos e ondas de rádio. É utilizada para diagnosticar e monitorizar doenças e condições tais como cancro, doenças cardíacas e perturbações neurológicas (Figura 3) (Hussain et al., 2022; MRI Scans: MedlinePlus, n.d.).

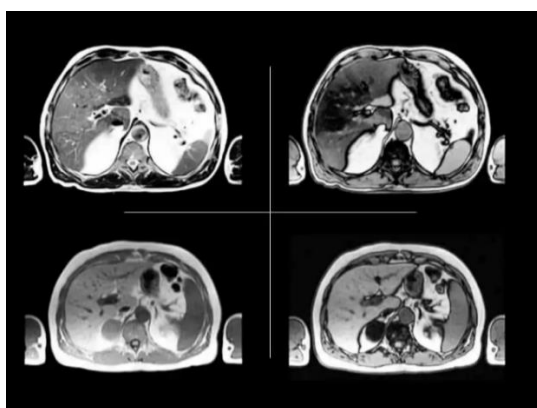


Figura 3 - Exemplo de uma ressonância magnética (*Ressonância Magnética (RMN): Como Funciona e Tipos de Exames, n.d.*)

2.2.3 Imagens de Ultrassom

A imagem por ultrassons utiliza ondas sonoras de alta frequência, com uma amplitude entre um e vinte mil megahertz (Ferreira, 2009; Soares, 2016), para produzir imagens de órgãos e de estruturas no interior do corpo. À medida que essas ondas atravessam o corpo, vão encontrando tecidos com distintas densidades e resistências que influem nos ecos reflexivos e que regressam à sonda de ultrassons com velocidades diferentes, sendo retransmitidos para um dispositivo que converte esses ecos em imagens (Alves et al., 2012). Ao medir essas ondas dos ecos é possível determinar a distância do objeto, o seu tamanho, a sua forma, a sua consistência e a sua uniformidade (Alexandre et al., 2011).

Esta técnica imagiológica é utilizada para detetar mudanças na morfologia, dimensões e estrutura dos órgãos (Alexandre et al., 2011). É principalmente usada para visualizar o feto durante a gravidez, mas também pode ser utilizada para diagnosticar várias condições médicas, tais como dores abdominais e doenças da vesícula biliar (Figura 4) (Hussain et al., 2022; Ultrasound: MedlinePlus Medical Test, n.d.).



Figura 4 - Exemplo de uma imagem ultrassom (*Você Sabe o Que é o Ultrassom Obstétrico?* - BedMed, n.d.)

2.2.4 Tomografia Computorizada (TAC)

A tomografia computadorizada tem sido a tecnologia na área da imagiologia médica que se mantém em constante transformação devido às grandes evoluções nas ciências da informação e tecnologias, o que tem possibilitado a realização de estudos no paciente com maior precisão (*De Magalhães & Faro, 2007*).

Caracteriza-se como uma técnica imagiológica que permite obter imagens transversais do corpo humano através de raios x e é utilizada para fornecer informações relevantes sobre a anatomia (*Soares, 2016*) e diagnosticar uma variedade de condições, incluindo tumores, derrames, doenças cardíacas, fraturas e alterações patológicas (Figura 5) (*CT Scans: MedlinePlus, n.d. Hussain et al., 2022*).

Fornece as informações através da multiplicidade de cortes imagiológicos finos da estrutura anatômica sem a sobreposição de outras estruturas, o que se torna muito vantajoso quando é necessário obter imagens de uma estrutura que esteja a sofrer sobreposição de outra (*De Magalhães & Faro, 2007; Soares, 2016*).



Figura 5 - Exemplo de uma tomografia computadorizada (*Tomografia Computadorizada (TC) Em Distúrbios Neurológicos - Distúrbios Neurológicos - Manuais MSD Edição Para Profissionais, n.d.*)

2.2.5 Medicina Nuclear

A medicina nuclear, ao inverso das restantes técnicas imagiológicas já aqui mencionadas avalia as alterações fisiológicas resultantes de mudanças bioquímicas a nível celular e sub-celular (Alves et al., 2012).

Esta técnica utiliza materiais radioativos que permitem medir a função tecidual bem como fornecer um marcador precoce da doença, através da medicação de alterações bioquímicas (Alves et al., 2012).

É utilizada para diagnosticar processos tumorais, arites, doenças cardíacas, entre outras, assim como para controlar a eficácia dos tratamentos (Figura 6) (Hussain et al., 2022; Nuclear Scans: MedlinePlus, n.d.).

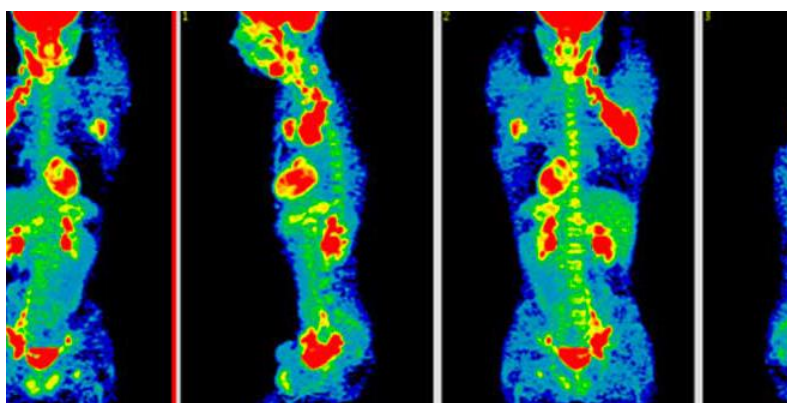


Figura 6 - Exemplo de uma imagem com suporte da medicina nuclear (*O Que é Um Estudo PET? - SPMN – Sociedade Portuguesa de Medicina Nuclear, n.d.*)

Estas são as técnicas de imagens médicas mais conhecidas, desta forma, foi feita uma abordagem a cada uma delas, sendo realizada uma explicação e as principais vantagens das mesmas.

2.2.6 Tipo de Imagem Médica Usada

Depois de analisadas as técnicas apresentadas, para o trabalho em questão irá ser usado a imagem de raio x para avaliar possíveis doenças num certo paciente, visto que a mesma apresenta algumas vantagens que a tornam a mais indicada (Kasban et al., 2015).

As vantagens no uso deste tipo de imagem são (*Advantages of X-Ray | Disadvantages of X-Ray, n.d; Kasban et al., 2015*):

- As imagens de raios x permitem diagnósticos rápidos e precisos;

- É relativamente barata em comparação com outras técnicas de imagem;
- É não invasiva e não requer qualquer preparação;
- Pode facilmente detetar fraturas nos ossos;
- Pode ser utilizado para detetar tumores e outras anomalias;
- Pode ser usado para diagnosticar e monitorizar certas condições médicas.

Tal como todas as técnicas de diagnóstico esta também apresenta algumas desvantagens (*Advantages of X-Ray | Disadvantages of X-Ray, n.d; Kasban et al., 2015*):

- A radiografia expõe o paciente à radiação;
- Só pode produzir imagens em duas dimensões, o que significa que não pode fornecer uma visão detalhada de certos órgãos ou estruturas;
- Não pode ser utilizado para detetar certas condições, tais como tumores nos pulmões ou outros tecidos moles.

Algumas aplicações médicas dos raios x são (*Kasban et al., 2015; X-Rays: Overview, Side Effects, Risks, and More, n.d.*):

- Podem ser usados para mostrar o movimento de órgãos, como o estômago, intestino e cólon;
- Também podem ser usados para estudar os vasos sanguíneos do coração e cérebro;
- Determinam o tipo e a extensão de uma fratura, detetam alterações patológicas nos pulmões, e usados para ver a estrutura do estômago e dos intestinos;
- Conseguem ainda medir o conteúdo mineral e a densidade óssea.

Estas imagens vão ser trabalhadas para gerar o melhor diagnóstico possível e resposta por parte do modelo.

2.3 Técnicas e Métodos de Tratamento de Imagem Médica

Para este projeto será definido um método proposto na Secção 3.3. Em seguida será explicado quais técnicas e métodos podem ser usados no tratamento de imagem médica.

Inicialmente serão analisadas técnicas de pré-processamento de imagem. Na prática, a melhoria da imagem inclui contraste, manipulação de intensidade, remoção de ruído, filtragem,

entre outros. De seguida será estudado qual rede neuronal artificial irá ser utilizada para determinar os resultados obtidos através da análise das imagens.

2.3.1 Pré-Processamento e Segmentação de Imagem

Foi abordado o pré-processamento de imagem e a forma como o mesmo potencia uma grande melhoria no resultado final da análise. É mencionado como funciona esta melhoria e que técnicas são usadas para tal. Ainda é referido alguns aspetos sobre a filtragem linear, como, o seu objetivo, e que vantagens traz para este processamento de imagem.

Para além do pré-processamento, foi descrita a segmentação de imagem, foram abordados alguns tópicos como objetivos e que técnicas existem neste âmbito, abordando ainda um deles, o crescimento de região.

2.3.1.1 Pré-Processamento

O pré-processamento de imagem é uma etapa essencial com o objetivo de melhorar a qualidade e a utilidade das imagens antes de realizar análises ou aplicar algoritmos específicos. Esta etapa envolve uma série de técnicas e procedimentos que visam corrigir imperfeições, remover ruídos indesejados, realçar características relevantes e facilitar a extração de informações importantes contidas nas mesmas (Jähne, n.d.).

As imagens são pré-processadas para ajustar as cores para uma melhor segmentação. O pré-processamento lida com algumas técnicas, entre as quais a filtragem linear, uma das importantes a ser realçada (Fang et al., 2021). É a técnica utilizada para melhorar a imagem de entrada. É um processo de aplicação de uma transformação linear a uma imagem, que normalmente é usada para suavizar ou realçar a imagem ou para melhorar certos recursos da mesma. A filtragem linear envolve a convolução da imagem com um *kernel*, que se trata de uma pequena matriz de pesos que determina a transformação aplicada à imagem. O *kernel* é aplicado à imagem deslizando sobre a mesma e calculando a soma ponderada dos valores dos píxeis sob o *kernel* em cada posição (Jähne, n.d.).

Por exemplo, um filtro de suavização simples pode ser criado usando um *kernel* com pesos iguais e normalizado, como:

1/9 1/9 1/9

1/9 1/9 1/9

1/9 1/9 1/9

Este *kernel* seria aplicado à imagem deslizando sobre a mesma e calculando o valor médio dos pixels sob o *kernel* em cada posição. Isto resultaria numa versão suavizada da imagem, com detalhes e ruídos reduzidos. Outros tipos de filtros lineares podem ser criados usando *kernels* diferentes com pesos diferentes, como filtros de detecção de borda ou filtros de nitidez. Estes filtros podem ser usados para melhorar certos recursos da imagem ou para extrair informações específicas da imagem (Bagherieh et al., 2014; Fang et al., 2021).

2.3.1.2 Segmentação de Imagem

A segmentação de imagem é o processo de dividir uma imagem em vários segmentos ou regiões em que cada um dos quais corresponde a um objeto ou plano de fundo diferente da mesma. O objetivo da segmentação de imagem é particionar uma imagem em regiões significativas que correspondem a diferentes objetos ou partes de objetos (Bagherieh et al., 2014).

Existem muitas técnicas para segmentação de imagens, incluindo o limiar, agrupamento e crescimento de região (Bagherieh et al., 2014; Rajeshwari & Shyamala, 2021). Limiar envolve dividir uma imagem em dois ou mais segmentos com base nos valores de intensidade dos pixels. O agrupamento envolve a agrupação de pixels em *clusters* com base nos seus recursos, como cor ou textura.

A segmentação de imagens é uma tarefa importante no processamento de imagens e na visão computacional, pois permite a análise e o reconhecimento de objetos individuais numa imagem. Também é frequentemente usado como uma etapa de pré-processamento para outras tarefas, como reconhecimento de objetos ou registo de imagens (Bagherieh et al., 2014).

A técnica de crescimento da região também apresenta as suas características, serve para segmentar áreas de interesse numa imagem e tem algumas vantagens, tais como (Region Growing - Wikipedia, n.d.; Region Segmentation Computação Visual e Multimédia Chap. 7- Region Segmentation, n.d.):

- Simplicidade: O crescimento da região é relativamente simples de implementar e entender em comparação com outras técnicas de segmentação;
- Velocidade: Pode ser relativamente rápido, especialmente para imagens grandes ou quando implementado em hardware paralelo;
- Robustez: Pode ser robusto a ruídos e pequenas variações na imagem, pois baseia-se em critérios locais e não em propriedades globais;
- Adaptabilidade: Pode ser facilmente customizado para diferentes tipos de imagens e aplicações ajustando os critérios usados para definir a região.

No entanto, também existem algumas desvantagens para o crescimento da região:

- Sensibilidade aos pontos de origem: a segmentação resultante pode ser sensível à escolha dos pontos de origem, o que pode afetar a precisão e integridade da segmentação;
- Limitado a regiões conectadas: geralmente é limitado à segmentação de regiões conectadas e pode não funcionar bem para separar objetos desconectados ou para lidar com mudanças topológicas, como fusão ou divisão;
- Determinação de limites: pode ser difícil determinar com precisão os limites das regiões, especialmente para regiões com formas complexas ou irregulares;
- Limitado a regiões homogêneas: geralmente é adequado para segmentar regiões com propriedades relativamente homogêneas e pode não funcionar bem para imagens com várias regiões distintas e com propriedades diferentes.

2.4 Redes Neurais

As redes neurais são estruturas interligadas de unidades de processamento, que conseguem aprender a partir de um conjunto de dados para o intuito de resolver diferentes tipos de problemas (Aggarwal, 2018; Aggarwal et al., 2014). Existem alguns tipos de redes neurais, como, redes *feedforward*, redes convolucionais, redes recorrentes, codificadores automáticos e redes generativas. Todas estas mencionadas serão descritas e será explicado para que serve cada uma delas.

2.4.1.1 Rede Neuronal Artificial

Uma rede neuronal artificial (RNA) é um modelo computacional inspirado na estrutura e função do cérebro (Fang et al., 2021). É formado por um grande número de unidades de processamento interligadas chamadas neurónios artificiais, que trabalham juntos para processar informações e resolver problemas (Aggarwal, 2018; Aggarwal et al., 2014).

Num nível alto, uma RNA recebe dados de entrada, processa-os através da rede de neurónios artificiais e produz uma saída. A saída pode ser uma previsão, uma decisão ou alguma outra forma de informação (Chan et al., 2020).

Cada neurónio artificial recebe entrada de outros neurónios e executa uma computação nessa entrada para produzir uma saída, que é então passada para outros neurónios ou para a camada

final da rede. A saída de cada neurónio é determinada pelo seus pesos e tendências, que são ajustados durante o processo de treino para otimizar o desempenho da rede (Aggarwal, 2018; Vasuki et al., 2017).

As RNAs são amplamente utilizadas numa variedade de aplicações, incluindo reconhecimento de imagem e fala, processamento de linguagem natural e até jogos. Provaram ser particularmente eficazes em tarefas que exigem reconhecimento de padrões e a capacidade de aprender com grandes quantidades de dados (Goos et al., n.d.; Hesamian et al., 2019).

Existem vários tipos de redes neuronais artificiais, e cada tipo tem as suas próprias características e é adequado para diferentes tarefas. De seguida são apresentadas algumas destas redes bem como as suas aplicações.

2.4.1.2 Redes Neuronais FeedForward

Redes neuronais *feedforward* são o tipo mais básico de RNAs, nas quais os dados de entrada fluem pela rede em uma única direção, da camada de entrada para a camada de saída, sem *loopback* (9 Types of Neural Networks: Applications, Pros, and Cons, n.d.; Types of Neural Networks and Definition of Neural Network, n.d.; Aggarwal et al., 2014).

Esta arquitetura simples permite a classificação e a regressão de dados. Alguns dos exemplos de aplicação são, reconhecimento de texto escrito à mão (Fernando & Uzai, 2018), deteção de fraudes em transferências financeiras (Prete & Dissertação, 2014) e diagnóstico baseado em imagens (Aggarwal, 2018).

2.4.1.3 Redes Neuronais Convolucionais

São um tipo de rede neuronal *feedforward* que são particularmente eficazes para tarefas de classificação e reconhecimento de imagens (Demir et al., 2019). Elas são projetadas para processar dados com uma topologia *grid-like*, como uma imagem, usando camadas convolucionais que aplicam um conjunto de filtros aos dados de entrada (Aggarwal, 2018; Aggarwal et al., 2014).

Estas redes são especialmente eficazes em tarefas de classificação e reconhecimento de padrões nas imagens. Além disso, também podem ser aplicadas em áreas como processamento de vídeo, diagnóstico médico (Apostolopoulos & Mpesiana, 2020) e carros autônomos (Couto, 2012).

2.4.1.4 Redes Neurais Recorrentes

Redes neuronais recorrentes são um tipo de RNAs que possuem ligações de *feedback*, que permitem processar dados sequenciais, como séries temporais ou linguagem natural (Aggarwal, 2018). Diferente das redes *feedforward*, estas têm a capacidade de manter informações de entradas anteriores e usá-las para processar entradas atuais (*9 Types of Neural Networks: Applications, Pros, and Cons, n.d.*).

São especialmente úteis em tarefas como, tradução automática (Caseli, 2017), previsão de séries temporais, reconhecimento de fala (Lima et al., n.d.), e legendas automáticas para imagens (*Vista Do Deep Learning Para Geração Automática de Legenda de Imagem, n.d.*).

2.4.1.5 Codificadores Automáticos

Codificadores automáticos são um tipo de rede neuronal *feedforward* usada para aprendizagem não supervisionada. Eles são projetados para reconstruir os dados de entrada e aprender através de uma representação compacta deles, chamada codificação, por meio de um processo de codificação-decodificação (Aggarwal et al., 2014).

Os codificadores automáticos podem ser usados na redução de dimensionalidade, onde dados de alta dimensão são comprimidos numa representação mais compacta, preservando as principais características dos mesmos. Também podem ser aplicados na detecção de anomalias, onde o modelo é treinado para reconstruir dados e identificar desvios significativos de padrões conhecidos (*Capítulo 58 - Introdução Aos Autoencoders - Deep Learning Book, n.d.*).

2.4.1.6 Redes Adversárias Generativas

Redes adversárias generativas são um tipo de RNA que consiste em duas redes. Uma rede geradora e uma rede discriminadora (Aggarwal et al., 2014). A rede geradora gera dados sintéticos que pretendem ser indistinguíveis dos dados reais, enquanto a rede discriminadora tenta distinguir os dados sintéticos dos reais. As duas redes são treinadas num processo contraditório, onde o gerador tenta produzir dados que o discriminador não consegue distinguir dos dados reais, e o discriminador tenta identificar corretamente os dados sintéticos (Aggarwal, 2018).

Estas redes são usadas para gerar imagens foto realistas, sintetizar áudios, traduzir imagens, e até mesmo para melhorar a resolução das mesmas (Leite, n.d.; *Capítulo 54 - Introdução Às Redes Adversárias Generativas (GANs - Generative Adversarial Networks) - Deep Learning Book, n.d.*).

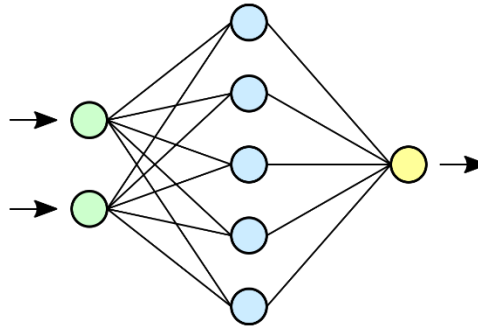


Figura 7 – Exemplo de uma Rede Neuronal Artificial (*Rede Neural Artificial – Wikipédia, a Enciclopédia Livre, n.d.*)

Com isto, foi possível perceber quais são os exemplos de aplicação para cada tipo de rede neuronal abordada. Cada uma destas possui diferentes características e aplicações em várias áreas, apresentando assim a diversidade e a utilidade das mesmas em diferentes problemas e domínios.

2.5 Modelo Computacional

Quando se trata de analisar e categorizar imagens com vários rótulos, usar uma Rede Neuronal Convolutiva (CNN) é uma escolha altamente eficaz, por isso, será a técnica usada neste projeto (*Image Processing Using CNN | Beginner's Guide to Image Processing, n.d.*). Aqui estão algumas razões pelas quais as CNNs são particularmente adequadas para esta tarefa em comparação com outros tipos de redes neurais (*Li et al., 2022*):

- Extração de recursos hierárquicos: As CNNs são projetadas para automaticamente aprender representações hierárquicas dos *pixels* no seu formato bruto. As camadas convolucionais de uma CNN extraem recursos da imagem, como arestas, texturas e formas, em diferentes níveis. Estas camadas capturam hierarquias espaciais, permitindo que a rede identifique padrões significativos nos dados da imagem;
- Invariância de tradução: As CNNs são inerentemente invariantes à tradução, o que significa que podem reconhecer padrões e recursos independentemente da sua localização específica numa imagem. Esta propriedade é essencial para a análise de imagens, pois os objetos podem aparecer em diferentes posições dentro da imagem;
- Parâmetros de peso partilhados: As CNNs utilizam parâmetros de peso compartilhados em diferentes localizações espaciais, de modo a permitir que a rede aprenda e generalize com eficiência os padrões espaciais. Ao compartilhar os pesos, as CNNs

podem capturar com eficácia os padrões locais em toda a imagem, reduzindo o número de parâmetros e melhorando a eficiência computacional;

- Agrupamento para redução de escala espacial: Camadas de agrupamento, normalmente usadas em *CNNs*, ajudam a reduzir as dimensões espaciais dos mapas de recursos, mantendo as informações importantes. Este processo de redução da resolução permite que a rede se concentre nos recursos mais salientes enquanto descarta detalhes espaciais irrelevantes. O agrupamento também fornece algum grau de invariância para pequenas variações espaciais, tornando a rede mais robusta para transformações de imagem;
- Capacidade de aprender representações complexas: As *CNNs* são capazes de aprender representações complexas e abstratas de imagens, permitindo-lhes capturar relacionamentos intrincados entre recursos e classes. Esta capacidade torna as *CNNs* altamente eficazes para tarefas de classificação de imagens com vários rótulos, pois aprendem a diferenciar vários conceitos visuais em simultâneo;
- Disponibilidade de modelos pré-treinados: As *CNNs* alcançaram um sucesso notável em tarefas de análise de imagens, o que levou à disponibilidade de vários modelos pré-treinados.

Alguns destes modelos disponíveis pré-treinados são (He et al., 2016; Huang et al., 2016):

- *DenseNet121*: É uma arquitetura de CNN proposta no artigo "*Densely Connected Convolutional Networks*" por Gao Huang, Zhuang Liu, Laurens van der Maaten e Kilian Q. Weinberger. Esta arquitetura destaca-se pela sua ligação densa entre camadas. Em vez de simplesmente construir as camadas umas após as outras, o *DenseNet* liga cada camada a todas as camadas subsequentes. Isto permite que as informações fluam livremente por toda a rede, promovendo a aprendizagem da mesma através de recursos partilhados e uma melhor utilização dos gradientes durante o treino (Szegedy et al., 2016). O "121" no nome refere-se ao número total de camadas na rede;
- *ResNet-152V2*: Trata-se de uma versão modificada da arquitetura *ResNet* originalmente proposta no artigo "*Identity Mappings in Deep Residual Networks*" por Kaiming He, Xiangyu Zhang, Shaoqing Ren e Jian Sun. A *ResNet* (Rede Residual) introduziu o conceito de ligações residuais, permitindo que o modelo aprenda as diferenças residuais entre as camadas em vez de aprender diretamente as representações. A versão *ResNet-152V2* é uma variante mais profunda da *ResNet*, com 152 camadas. Esta arquitetura profunda permite a extração de recursos mais complexos e uma melhor representação dos dados, resultando num melhor desempenho em várias tarefas de visão computacional (Szegedy et al., 2016);
- *InceptionResNetV2*: É uma arquitetura de CNN que combina os conceitos das redes *Inception* e *ResNet*. Foi proposto no artigo "*Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*" por Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke e Alexander A. Alemi. Esta arquitetura utiliza blocos *Inception*, que são responsáveis por extrair recursos em diferentes escalas e com diferentes filtros convolucionais em conjunto com ligações residuais da *ResNet*, que ajudam a mitigar o

problema de desvanecimento de gradientes em redes mais profundas (Szegedy et al., 2016). Esta combinação resulta numa rede capaz de extrair recursos complexos, permitindo um desempenho muito bom em tarefas de visão computacional.

Em resumo, as CNNs destacam-se a analisar imagens e categorizá-las em vários rótulos devido à sua capacidade de aprender de uma forma hierárquica (Yadav & Jadhav, 2019). Além disso, a disponibilidade de modelos pré-treinados aumenta ainda mais o desempenho e a eficiência das CNNs em tarefas de classificação de imagens com múltiplos rótulos (Albahli et al., 2021; Sarvamangala & Kulkarni, 2022).

2.6 Contributos da IA na Medicina

A IA já provou a sua utilidade na comunidade científica e médica porque tem vantagens e benefícios como os seguintes (Briganti & Le Moine, 2020; *Efeitos Da Inteligência Artificial Na Medicina | APD, n.d.*):

- Diagnóstico e planeamento de tratamento: a IA pode ser usada para analisar os dados do paciente, incluindo histórico médico, resultados de exames e estudos de imagem, para ajudar a diagnosticar doenças e identificar as opções de tratamento mais eficazes;
- Desenvolvimento de medicamentos: a IA pode ser usada para analisar grandes quantidades de dados de ensaios clínicos e outras fontes para identificar novos medicamentos e prever a sua eficácia;
- Apoio à decisão clínica: a IA pode ser usada para fornecer recomendações em tempo real a médicos e outros profissionais de saúde, com base nas evidências médicas e práticas recomendadas mais recentes;
- Medicina personalizada: a IA pode ser usada para analisar os dados genéticos e outros dados de saúde de um indivíduo para adaptar os planos de tratamento às suas necessidades específicas;
- Gestão da saúde da população: a IA pode ser usada para analisar dados de grandes populações para identificar tendências e padrões que possam informar o desenvolvimento de políticas e intervenções de saúde pública;
- Monitoramento do paciente: a IA pode ser usada para monitorizar continuamente os pacientes, alertando os profissionais de saúde sobre possíveis problemas e ajudando a prevenir complicações.

2.6.1 Aplicações na área de Medicina

A IA pode ser aplicada em praticamente todas as áreas da medicina, uma vez que se trata de uma ferramenta de apoio ao profissional de saúde que complementa e aprimora o seu trabalho. Para ilustrar as vantagens mencionadas que já estão presentes neste campo, há alguns exemplos de aplicações que a IA tem na medicina, nomeadamente no seguinte (*Inteligência Artificial Na Medicina: Aplicações e Benefícios, n.d.*).

1. Na análise de imagem médica

Um dos benefícios mais importantes da IA é a velocidade com que pode processar grandes quantidades de dados. Algoritmos de IA podem ser treinados para identificar características específicas em imagens, como lesões, tumores, anomalias ou variações que por vezes possam ser difíceis de detetar a olho nu. Isto ajuda os médicos e radiologistas no diagnóstico e na interpretação de exames, como ressonâncias magnéticas, tomografias computadorizadas e raios x (*Briganti & Le Moine, 2020*).

2. Nos diagnósticos médicos

Esta é outra das aplicações mais interessantes, pois os dados podem ser analisados para fazer diagnósticos precisos e precoces. A IA tem a capacidade de analisar dados clínicos e informações genéticas de pacientes, e podem ser construídos algoritmos capazes de identificar padrões e correlações que podem não ser óbvios para os médicos (*Briganti & Le Moine, 2020*). Isto pode auxiliar na deteção e identificação de fatores de risco e na previsão de resultados clínicos.

3. Nos tratamentos farmacológicos

No campo da farmacologia, a IA também oferece inúmeras vantagens. Com a análise de grandes conjuntos de dados genéticos, moleculares e clínicos, os algoritmos de IA podem identificar possíveis alvos terapêuticos, projetar novas moléculas e acelerar o processo de descoberta de medicamentos (*Briganti & Le Moine, 2020*). Desta forma, facilita o trabalho dos pesquisadores no momento de analisar sequências genéticas para encontrar vacinas ou soluções adequadas para diferentes problemas.

4. Na genética

Um dos campos que, no futuro, irá beneficiar mais com a aplicação da IA é a genética (*Briganti & Le Moine, 2020*). Atualmente já existem ferramentas que o demonstram, como é o caso da aplicação móvel que, através de um sistema de reconhecimento facial, permite detetar doenças raras e genéticas (*Inteligência Artificial Na Medicina: Como Ela Pode Ser Usada? - Blog Da IPEMED, n.d.*).

Para além destes existem outros exemplos de aplicação em que a IA tem um efeito muito positivo na medicina, como por exemplo, a cirurgia assistida, e o controlo de pacientes no que toca aos seus sinais vitais, níveis de glicose e outros padrões (*Inteligência Artificial Na Medicina: Aplicações e Benefícios, n.d.*).

2.7 Soluções Existentes

É quase impossível falar sobre o futuro da medicina sem mencionar que a inteligência artificial (IA) ganhou força na área da saúde na última década, em parte devido aos avanços nas redes neuronais de *deep learning*. Redes neuronais como mencionado anteriormente, são um conjunto de algoritmos organizados em nós e camadas que mimetizam as funções cognitivas humanas e são projetadas para que consigam reconhecer padrões (*Li et al., 2022*).

A OMS não fez uma recomendação sobre o uso de sistemas de leitura automatizada devido à atual falta de evidências (*Qin et al., 2019*). Para preencher a lacuna de evidências, foi comparado o desempenho de três aplicativos DL diferentes na deteção de tuberculose confirmada bacteriologicamente por radiologistas experientes na deteção da mesma, usando conjuntos de dados de dois países (*Putha et al., 2018; Qin et al., 2019*).

Três sistemas DL com versão estável foram incluídos neste estudo (*Putha et al., 2018; Qin et al., 2019*). CAD4TB (versão 6), qXR (versão 2) desenvolvido pela Qure.ai (Índia) e Lunit INSIGHT (versão 4.7.2) desenvolvido por Lunit (Coreia do Sul). Foram usadas as últimas versões disponíveis dos três sistemas DL nesta avaliação. A versão 6 do CAD4TB difere das versões anteriores porque usa DL. Tanto o CAD4TB como o Lunit conseguem ler apenas o formato DICOM (*Digital Imaging and Communications in Medicine*), enquanto o qXR pode analisar radiografias digitais armazenadas em PNG e JPEG. O CAD4TB deteta anormalidades específicas da tuberculose e gera pontuações contínuas de anormalidade variando de 0 a 100. Quanto maior a pontuação da anomalia, maior a probabilidade da presença de tuberculose (*Qin et al., 2019*). As versões atuais do qXR16 e Lunit17 detetam várias anormalidades pulmonares discretas, como calcificação, cavitação, opacidades, entre outras (*Qin et al., 2019*). Ambos os sistemas apresentam os resultados para a tuberculose e as anormalidades clínicas específicas em binário (“Sim” / “Não”) usando um *score* de anormalidade limiar. As pontuações de anormalidade para Lunit e qXR variam de 0 a 100%. Todos os três sistemas DL podem gerar mapas de calor mostrando anormalidades (*Qin et al., 2019*).

Foi feita uma avaliação dos três sistemas DL na avaliação da precisão e diagnóstico usando imagens de raio x do Nepal e Camarões (*Qin et al., 2019*). Neste conjunto estavam presentes os dados de adultos (com 15 anos ou mais) com sintomas sugestivos de tuberculose (tosse há mais de 2 semanas, febre, suores noturnos e perda de peso) e que tinham sido consecutivamente

inscritos no ambulatório pulmonar do Instituto Koirala de Ciências da Saúde (BPKIHS) no leste do Nepal entre 28 de junho e 24 de dezembro de 2015 e no Laboratório de Referência de Tuberculose Bamenda e no Hospital Regional de Bamenda em Camarões entre 9 de setembro de 2015 e 15 de abril de 2016. Cada participante do estudo efetuou um raio x posteriormente (Putha et al., 2018; Qin et al., 2019). Todos os sistemas mostraram um grande nível de performance na análise destas radiografias, retornando classificações e respostas acertadas, sendo uma mais valia para os profissionais de saúde.

Em ambos os locais, cada raio x foi classificado como “anormal” se qualquer anomalia pulmonar fosse detetada pelos radiologistas (Qin et al., 2019), independentemente de a mesma ser específica de tuberculose, ativa ou antiga.

No Nepal, cada radiografia foi lida duas vezes por dois grupos de radiologistas (Qin et al., 2019). A primeira leitura foi feita por um professor de radiologia com um doutoramento em radiologia na BPKIHS com 21 anos de experiência e a segunda leitura foi feita por um grupo de radiologistas juniores em rotação no BPKIHS, todos com 3 a 5 anos de experiência.

Nos Camarões, cada radiografia foi lida pela primeira vez por um radiologista de campo com 9 anos de experiência em radiologia. Independentemente dos resultados do radiologista de campo, todas as radiografias eram enviadas anonimamente para uma empresa de tele radiologia remota, chamada *Teleradiology Solutions*, que foi credenciada em 2005 pela *Joint Commission* (Qin et al., 2019). Enquanto a leitura humana foi feita prospectivamente, os três sistemas DL pontuaram as imagens retrospectivamente. As imagens foram transferidas para o Lunit e qXR para a sua leitura através de *Secured File Transfer Protocol (SFTP)*, e para Delft através de transferência em *cloud*. Todas as leituras de máquina foram realizadas de forma independente.

No final foi validada a precisão e performance dos radiologistas e dos três sistemas de DL (Figura 8).

	Human Readers			Deep Learning Systems								
	Accuracy	Sensitivity (95%CI)	Specificity (95% CI)	CAD4TB (v6)			Lunit (v4.7.2)			qXR (v2)		
				Accuracy	Sensitivity (95% CI)	Specificity (95% CI)	Accuracy	Sensitivity (95% CI)	Specificity (95%CI)	Accuracy	Sensitivity (95% CI)	Specificity (95% CI)
Nepal												
Senior Radiologist	0.57	0.96 (0.89-0.99)	0.48 (0.43-0.53)	0.74	0.96 (0.89-0.99)	0.69 (0.64-0.73)	0.67	0.96 (0.89-0.99)	0.6 (0.55-0.65)	0.7	0.97* (0.91-0.99)	0.65 (0.6-0.69)
Junior Radiologist & Residents	0.72	0.87 (0.79-0.93)	0.69 (0.64-0.73)	0.77	0.87 (0.79-0.93)	0.75 (0.71-0.79)	0.85	0.87 (0.79-0.93)	0.78 (0.73-0.82)	0.69	0.87 (0.79-0.93)	0.81 (0.76-0.84)
Cameroon												
Radiologist	0.74	0.8 (0.52-0.96)	0.74 (0.71-0.78)	0.9	0.8 (0.52-0.96)	0.9 (0.87-0.92)	0.94	0.8 (0.52-0.96)	0.94 (0.92-0.96)	0.94	0.8 (0.52-0.96)	0.95 (0.93-0.96)
Teleradiology Company	0.74	0.8 (0.52-0.96)	0.74 (0.71-0.77)	0.9	0.8 (0.52-0.96)	0.9 (0.87-0.92)	0.94	0.8 (0.52-0.96)	0.94 (0.92-0.96)	0.94	0.8 (0.52-0.96)	0.95 (0.93-0.96)

Figura 8 - Performance e Resultados dos Sistemas (Qin et al., 2019)

O Lunit e o qXR tiveram um desempenho tão bom quanto o CAD4TB em diferentes métricas de análise e todos os três sistemas de DL superaram as expectativas na identificação de pessoas com tuberculose confirmada bacteriologicamente por profissionais de saúde (Qin et al., 2019). Todos os três produtos tiveram um desempenho semelhante e podem ser usados para reduzir o número de testes subsequentes, na análise destes casos, fornecendo um melhor custo-benefício.

O princípio destes modelos é que o desempenho melhora com a exposição destes sistemas a mais exemplos de raio x. Foi concluído que estas tecnologias, portanto, têm o potencial de aumentar a capacidade e melhorar o diagnóstico e tratamento geral da tuberculose, especialmente em locais com escassez de radiologistas (Qin et al., 2019).

Estas foram algumas das soluções já existentes neste ramo, que foram estudadas e testadas para o âmbito de doenças infecciosas. Além disso, as mesmas foram usadas num ambiente profissional e mostraram-se muito capazes no apoio à decisão médica através de boas avaliações e classificações de radiografias. Podemos então concluir, que estes tipos de soluções são vantajosos neste cenário, visto que, conseguem dar uma resposta acertada e rápida de análise.

2.8 Ética e Proteção

A ética e a proteção de dados são aspetos fundamentais na medicina. A confidencialidade dos pacientes é uma parte crucial da relação médico-paciente e é importante garantir que a privacidade dos pacientes seja protegida. Isso inclui garantir que os dados médicos dos pacientes sejam armazenados de forma segura e acessíveis apenas para aqueles com autorização apropriada (*Ética Na Medicina: Importância, Interpretação e Novo Código Ética Médica, n.d.*).

Para um sistema de análise de imagem médica é importante garantir que os pacientes deem o seu consentimento antes de qualquer procedimento médico ou levantamento de dados. Isso inclui explicar claramente os riscos e benefícios deste procedimento ou levantamento de dados, bem como as alternativas disponíveis (*Aspectos Éticos Nos Sistemas de Informação de Saúde - Medicina S/A, n.d.*).

Outra questão ética importante na medicina é a pesquisa médica. É crucial garantir que os pacientes envolvidos em estudos clínicos deem o seu consentimento informado e antecipado e que sejam protegidos contra quaisquer danos desnecessários. Além disso, é importante garantir que os resultados dos estudos sejam divulgados de forma transparente e que os benefícios da pesquisa sejam compartilhados com a comunidade médica e a sociedade (*Nunes, n.d.*).

Em resumo, a ética e a proteção de dados são fundamentais na medicina e é fundamental garantir que os pacientes sejam tratados com respeito e dignidade em qualquer tipo de procedimento.

2.9 Resumo/Conclusão

Neste capítulo o estado de arte foi desenvolvido. Foi mencionado inicialmente a metodologia de pesquisa *PRISMA*, onde foram apresentados os conceitos e princípios da mesma, ainda neste âmbito foi feita a identificação do processo de pesquisa e definidas as questões, repositórios e critérios de seleção. De seguida, foram explicadas algumas técnicas de diagnóstico na análise de imagens onde foram descritas as imagens de Raio x, ressonâncias magnéticas, imagens de ultrassom, tomografias computadorizadas e medicina nuclear e ainda apresentado um exemplo de cada. Depois foi abordado um pouco mais a fundo as imagens de Raio x visto que serão as usadas pelo sistema e foram apresentadas algumas vantagens e desvantagens das mesmas. De seguida são explicadas as técnicas e métodos existentes para o desenvolvimento deste sistema e são descritas as seguintes:

- Pré Processamento: Nesta etapa é descrito em que consiste o pré processamento e como o mesmo funciona;
- Segmentação de Imagem: Na segunda etapa é caracterizado a segmentação de imagem e respetivas técnicas e são apresentadas algumas vantagens e desvantagens do crescimento de região;
- Rede Neuronal: Como terceira etapa temos as redes neuronais, onde foram apresentados alguns tipos das mesmas e as suas especificações. Por fim ainda foi explicada qual rede será usada no desenvolvimento do sistema.

É exposto também alguns contributos de inteligência artificial na área da medicina e especificado algumas aplicações nesta área.

Ainda no estado da arte foram apresentadas três soluções existentes, que efetuam a análise de radiografias para deteção de problemas infecciosos, foi exposto um estudo com estes 3 sistemas comparando os resultados obtidos entre eles e a precisão dos mesmos com a análise de experientes radiologistas na deteção da tuberculose. Por fim foi descrita a ética e a proteção de dados para a obtenção dos mesmos.

Neste projeto será desenvolvida uma solução que faz a avaliação e análise de raios x. Para esta solução será criada uma rede neuronal convolucional que classifica uma imagem, depois de efetuar a análise da mesma, de forma a obter o resultado e a respetiva precisão desta classificação.

3 Análise e Design

Este capítulo apresenta uma revisão da análise dos requisitos e a arquitetura da interface para o utilizador.

O capítulo inicia com a explicação dos atores principais da solução, de seguida, é realizado o levantamento de requisitos onde será explicado de uma forma breve todas as condições necessárias para o desenvolvimento. Foi acrescentado um subcapítulo 3.2 onde será abordado algumas informações relevantes sobre o *dataset* escolhido, visto que é a base fundamental para este projeto. De seguida, é explicado o método proposto e como o modelo será construído, por fim, é apresentado como será o processo na interface gráfica, neste caso no produto final.

3.1 Análise de Requisitos

Nesta secção serão apresentados e identificados os atores principais do sistema, e em seguida será efetuado o levantamento de requisitos para esta solução.

3.1.1 Atores Principais

Para esta solução foram identificados dois atores principais:

- Sistema – entidade encarregue de efetuar a análise da radiografia e apresentar as conclusões no final do processo.
- Utilizador – entidade que irá utilizar esta aplicação, que neste caso será um radiologista ou um profissional de saúde. O mesmo fornecerá ao sistema as radiografias a serem analisadas por este.

3.1.2 Levantamento de Requisitos

O levantamento de requisitos é dividido em 3 fases:

- Preparação
- Análise
- Resultados

3.1.2.1 Preparação

No início teremos de preparar o sistema para receber radiografias e produzir as conclusões necessárias que é o objetivo principal do mesmo. Para isso será necessário um *dataset* modelo constituído por várias colunas com informações relevantes para treinar o modelo e melhorar a sua análise e precisão. Este *dataset* terá de conter informações relacionadas com os pacientes, como a idade, o sexo, a radiografia da pessoa e outros tipos de dados relevantes para a análise. No final da preparação, o sistema terá de ser capaz de identificar alguns tipos de doenças infecciosas e apresentar uma precisão na análise das mesmas, para além disso, terá também de ser capaz de identificar possíveis radiografias que apresentem sintomas iniciais de algumas destas doenças e alertar para um possível problema no futuro.

3.1.2.2 Análise

Depois da conclusão da preparação o sistema estará pronto a ser usado. Para o mesmo apresentar resultados precisa de alguém, neste caso o utilizador que será um profissional de saúde, que forneça os dados a serem analisados, neste caso os do paciente e a respetiva radiografia. Depois de fornecidos estes dados, o sistema irá analisar os mesmos.

3.1.2.3 Resultados

Quando a solução acabar a análise dos dados, apresenta os resultados obtidos. Nestes resultados são apresentados os problemas do paciente que foi analisado, caso este apresente sinais de algum. Pode também ser apresentado um aviso de um potencial futuro problema relacionado com qualquer doença infecciosa.

De uma forma geral, para o sistema ser funcional é necessário:

- **Desenvolvimento e Treino:** Fornecer todos os dados necessários para o treino da solução para que esta se torne a mais autónoma e precisa possível;
- **Análise e Resultados:** Depois de concluído o desenvolvimento e treino, o sistema está pronto a ser usado pelos profissionais de saúde. Para que este produza resultados estes mesmos profissionais terão de fornecer informações ao sistema relacionados com os pacientes.

3.2 Dataset

Para este projeto foi usado o *National Institutes of Health Chest X-Ray Dataset (NIH Chest X-ray)*. Este *dataset* encontra-se disponível no *kaggle* e é aquele cujo conjunto de dados mais se adequa a este projeto (*NIH Chest X-Rays | Kaggle, n.d.*).

3.2.1 Fundamento e Motivação

O raio x é um dos exames mais frequentes e económicos de radiografia médica. No entanto, este diagnóstico clínico pode ser desafiador e acredita-se que por vezes possa ser mais difícil do que o diagnóstico por meio de tomografia computadorizada (TAC). Alguns trabalhos promissores foram relatados, especialmente, trabalhos recentes relacionados com *deep learning* na classificação da tuberculose. Para obter uma deteção assistida por computador clinicamente relevante é necessário radiografias e os respetivos dados o que é difícil, quando apenas uns poucos milhares de imagens estão disponíveis e ainda não existe *datasets* muito completos. Com isto, o desempenho das redes neuronais para o reconhecimento de doenças torácicas é severamente limitado pela disponibilidade de apenas 4143 imagens de vista frontal (anterior maior conjunto de dados de radiografia torácica publicamente disponível) (*NIH Chest X-Rays | Kaggle, n.d.*).

Neste conjunto de dados, é fornecida uma versão melhorada (com mais 6 categorias de doenças e também mais imagens) que é aproximadamente mais 27 vezes o número de imagens de radiografia torácica frontal no anterior maior conjunto de dados. Foi extraído da base de dados do *National Institutes of Health Clinical Center* e consiste em ~ 60% de todas as radiografias do hospital (*NIH Chest X-Rays | Kaggle, n.d.*). O conjunto de dados é significativamente mais representativo no que toca às distribuições reais da população de pacientes e desafios no diagnóstico clínico.

3.2.2 Detalhes

O conjunto de dados compreende 112.120 imagens de radiografia frontal de 30.805 pacientes com as 14 patologias de doenças (onde cada imagem pode ter mais do que uma) (Figura 9), extraídos dos relatórios radiológicos associados usando o processamento de linguagem natural.

Estas catorze patologias torácicas incluem (*Nahiduzzaman et al., 2023*):

- *Atelectasis*: É uma condição em que uma ou mais áreas dos pulmões colapsam ou não se inflam corretamente. Os doentes podem sentir falta de ar, dores no peito e tosse. Pode ocorrer devido à obstrução das vias respiratórias, ao cancro do pulmão ou a outras doenças pulmonares (*Atelectasis - Symptoms and Causes - Mayo Clinic, n.d.; Fang et al., 2021*);
- *Cardiomegaly*: Consiste numa doença em que o coração aumenta para além do seu tamanho normal. Pode ser causada por vários fatores, incluindo pressão arterial

elevada, doença valvular ou insuficiência cardíaca. Os sintomas podem compreender falta de ar, dor no peito e fadiga (Fang et al., 2021);

- *Effusion*: É a acumulação de líquido em excesso no espaço pleural, que é o espaço entre os pulmões e a parede torácica. Pode ser causada por muitos fatores, incluindo insuficiência cardíaca, pneumonia e cancro do pulmão. Os sintomas podem incluir dor no peito, tosse e falta de ar (Fang et al., 2021);
- *Infiltration*: Trata-se de uma condição em que os espaços de ar nos pulmões são preenchidos com substâncias como sangue, pus ou células. Pode ser causada por vários fatores, incluindo infeção, inflamação ou cancro. Os sintomas podem incluir tosse, falta de ar e dores no peito (Fang et al., 2021);
- *Mass*: refere-se a um crescimento anormal ou nódulo nos pulmões. Pode ser benigna ou maligna e pode ser causada pelo tabagismo, pela exposição a toxinas ou por fatores genéticos. Os sintomas podem não estar presentes nas fases iniciais (Fang et al., 2021);
- *Nodule*: é um pequeno crescimento nos pulmões que pode ser benigno ou maligno. É frequentemente detetada durante radiografias ou tomografias computadorizadas. Tal como na patologia anterior os sintomas podem não estar presentes nas fases iniciais (Fang et al., 2021);
- *Pneumonia*: é uma infeção pulmonar causada por bactérias, vírus ou fungos. Os sintomas podem incluir tosse, febre, falta de ar e dores no peito e pode ser tratada com antibióticos ou medicamentos antivirais (Fang et al., 2021);
- *Pneumothorax*: ocorre quando há uma fuga de ar para o espaço entre os pulmões e a parede torácica, provocando o colapso do pulmão. Pode ser causada por uma lesão, doença pulmonar ou cirurgia. Os sintomas podem incluir dor súbita no peito e falta de ar (Fang et al., 2021);
- *Consolidation*: refere-se ao preenchimento dos espaços de ar nos pulmões com líquido, pus ou outras substâncias. Pode ser causada por pneumonia, tuberculose ou outras infeções pulmonares e os sintomas podem incluir tosse, febre e falta de ar (Fang et al., 2021);
- *Edema*: é a acumulação de líquido nos pulmões, que pode ser causada por insuficiência cardíaca, doença renal ou doença pulmonar. Os sintomas podem incluir falta de ar, tosse e dor no peito (Fang et al., 2021);
- *Emphysema*: é uma doença pulmonar crónica em que os sacos de ar dos pulmões são danificados, causando falta de ar e aperto no peito. É frequentemente causada pelo tabagismo ou pela exposição a poluentes atmosféricos (Fang et al., 2021);
- *Fibrosis*: é uma doença em que o tecido pulmonar se torna espesso e cicatrizado, dificultando a respiração. Pode ser causada por muitos fatores, incluindo a exposição a toxinas, radiação ou doenças autoimunes (Fang et al., 2021);
- *Pleural_Thickening*: ocorre quando o revestimento dos pulmões fica espesso, causando dor no peito e dificuldade em respirar. Pode ser causada por exposição ao amianto, infeção ou outras doenças pulmonares (Fang et al., 2021);
- *Hernia*: é uma condição em que parte de um órgão ou tecido sobressai através de um ponto fraco no músculo ou no tecido circundante. No caso de uma hérnia diafragmática, parte do estômago ou outros órgãos podem sobressair através do diafragma para a

cavidade torácica. Os sintomas podem incluir dor no peito, dificuldade em respirar e azia (Fang et al., 2021).

Espera-se que as doenças atribuídas no *dataset* tenham uma precisão > 90%, isto é, espera-se que > 90% da informação do *dataset* esteja correta.

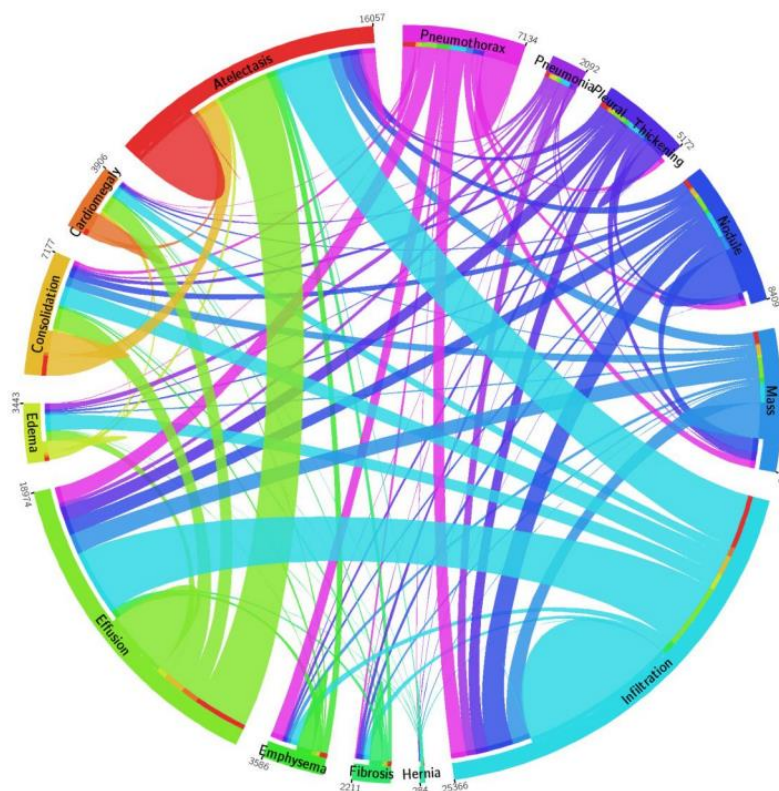


Figura 9 - Distribuição das categorias de doenças (Chest X-Ray Multi-Classification | Kaggle, n.d.; NIH Chest X-Rays | Kaggle, n.d.)

3.2.3 Limitações

Como dito anteriormente os rótulos das imagens são extraídos por linguagem natural, então pode haver alguns rótulos errados, mas estima-se que a precisão da informação seja > 90%. Trata-se de um *dataset* com uma quantidade grande de doenças e uma imensa quantidade de combinações entre elas, o que torna o treino bastante difícil, sendo que uma das poucas soluções encontradas com este *dataset* apresenta uma *accuracy* de ~0.15 (NIH Chest X-Ray Multi-Classification | Kaggle, n.d.). Para dificultar ainda mais o processo de treino, o *dataset* apresenta uma discrepância muito grande no que toca aos rótulos sendo que o *No Finding* (sem

doenças detetadas) corresponde a 54% do mesmo. Por fim e não menos importante, não se espera que os relatórios de radiologia sejam divulgados publicamente, por isso nem todos os dados contidos neste conjunto podem estar corretos.

Em geral, e como já dito previamente, o *dataset* é o mais acertado para o projeto em questão visto que o grande objetivo da solução é a detecção de várias doenças e não apenas de uma ou duas, como acontece na maioria dos sistemas de análise de radiografias.

3.3 Método Proposto

Para o desenvolvimento e construção do modelo foi seguido o seguinte método (Figura 10).

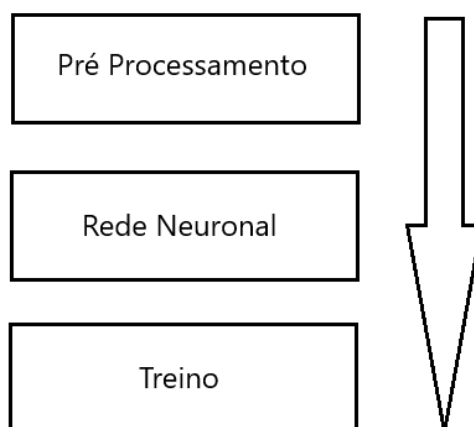


Figura 10 – Método Proposto

Inicialmente foi feito o pré processamento, foi usada a filtragem linear para melhorar a qualidade das imagens do conjunto de dados, cada uma destas foi filtrada para melhorar certos recursos da mesma. Como já mencionado anteriormente, esta técnica, envolve a convolução da imagem com um *kernel*. Este é aplicado à imagem deslizando sobre a mesma e calculando a soma ponderada dos valores dos pixels sob o *kernel* em cada posição (Bagherieh et al., 2014).

De seguida, foi desenvolvida uma rede neuronal convolucional, este tipo de redes mostraram uma boa *performance* em analisar imagens e produzir classificações, as camadas da rede serão explicadas de uma forma detalhada no capítulo 4 de implementação.

Por fim, a rede neuronal desenvolvida será treinada com o *dataset* escolhido, o *National Institutes of Health Chest X-Ray Dataset (NIH Chest X-ray)*, as 112.120 imagens de radiografia frontal de 30.805 pacientes serão pré processadas e a rede neuronal irá avaliá-las para produzir uma classificação, durante este treino será criado um modelo com a capacidade de analisar imagens e produzir avaliações acertadas para as mesmas.

3.4 Design

Para o projeto em questão será projetado um design (figura 11) que permita aos utilizadores do sistema selecionar imagens que pretendam que o sistema analise. Após as mesmas serem selecionadas, o sistema irá calcular a probabilidade para cada doença da respetiva imagem e irá processar os resultados para retornar as informações relevantes (doenças e possíveis doenças futuras).

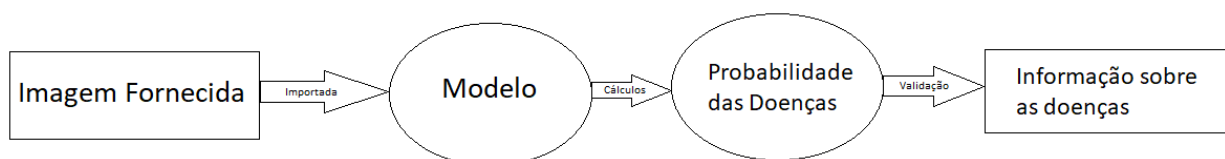


Figura 11 - Desenho do processo da Interface Gráfica

3.5 Resumo/Conclusão

Neste capítulo foi feita a análise geral e *design* da solução. Inicialmente foram descritos os atores principais e de seguida o levantamento de requisitos. Foram explicadas as 3 fases do levantamento de requisitos e como cada uma é aplicada no projeto. De seguida, foi mencionado algumas informações sobre o conjunto de dados escolhido, expondo alguns dos seus detalhes e limitações. Foi apresentado como será construído o modelo e que técnicas serão usadas no seu desenvolvimento, acabando o capítulo com o *design* do produto final. De uma forma geral, é apresentado um *overview* de como será o processo de desenvolvimento, quais serão as etapas a ser seguidas e como será o produto final.

4 Implementação

Este capítulo apresenta uma revisão do código desenvolvido para o presente projeto. Inclui o desenvolvimento do modelo e da interface gráfica, assim como apresenta alguns resultados obtidos com estes.

O capítulo começa por explicar qual a análise e a preparação feita ao *dataset* para ser utilizada posteriormente no treino do modelo. Em seguida é descrito o código desenvolvido para o treino do modelo, assim como o modelo de classificação que foi construído e o pré processamento das imagens. Para além disso, também são demonstrados os resultados obtidos com o modelo treinado. Importante salientar que uma segunda versão deste modelo foi implementada sem a preparação do *dataset* para comparação de resultados. Por fim, explica-se como foi construído a interface que será mostrada aos utilizadores.

4.1 Preparação e Análise do Dataset

Como dito anteriormente, o *dataset* escolhido para este projeto foi o *National Institutes of Health Chest X-Ray Dataset (NIH Chest X-ray)*. Consiste num conjunto de dados que contém 112.120 imagens de radiografia frontal de 30.805 pacientes com catorze patologias de doenças diferentes. Devido à sua complexidade por causa de todas as combinações que existem e da discrepância da quantidade de imagens para cada uma destas patologias, torna a aprendizagem do modelo mais difícil, por isso foi feita uma análise preparação do *dataset* antes do mesmo ser usado para o treino.

Depois desta análise e preparação, foram removidas todas as imagens que pertenciam ao rótulo de *No Finding* (sem doenças detetadas), com isto o *dataset* passou a ter 51759 imagens, mas desta forma foi garantido o máximo de equilíbrio entre os restantes rótulos para evitar que o modelo fique tendencioso. Como serão apenas removidas estas imagens e não serão efetuadas qualquer outro tipo de alterações, foram implementadas condições no código que permitiram fazer este filtro não efetuando qualquer alteração direta no conjunto de dados.

4.2 Ambiente de Desenvolvimento

O *PyCharm* é um ambiente de desenvolvimento integrado (IDE) para Python (*What Is PyCharm? Features, Advantages & Disadvantages, n.d.*). Desenvolvido pela *JetBrains*, o *PyCharm* fornece uma interface de fácil utilização, juntamente com uma vasta gama de funcionalidades que tornam a codificação mais fácil e mais eficiente. Algumas das suas funcionalidades incluem a navegação no código, a conclusão do código, a depuração e a reestruturação.

O *PyCharm* também tem um terminal integrado, que permite que os desenvolvedores executem scripts e comandos de *Python* diretamente do IDE. Além disso, o *PyCharm* suporta uma variedade de *frameworks*, como *Django*, *Flask* e *Pyramid*, tornando-o uma escolha ideal para o desenvolvimento web (*What Is PyCharm? Features, Advantages & Disadvantages, n.d.*).

Um dos aspetos mais positivos sobre o *PyCharm* é a sua capacidade de integração com sistemas de controlo de versões como o *Git*, facilitando aos programadores a gestão do seu código (*What Is PyCharm? Features, Advantages & Disadvantages, n.d.*). O *PyCharm* também possui um executor de testes integrado, que permite aos programadores escrever e executar testes sem sair do IDE.

No geral, o *PyCharm* é um IDE altamente recomendado para desenvolvedores de *Python*. Os seus recursos avançados, configurações personalizáveis e recursos de depuração tornam-no numa ferramenta essencial.

4.3 Implementação do Modelo

Inicialmente foram importadas as bibliotecas necessárias para o desenvolvimento do projeto.

```
import pandas as pd

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense

from keras.preprocessing.image import ImageDataGenerator
```

Primeiro foi importada a biblioteca *pandas*, utilizada na manipulação e análise de dados.

De seguida, foram importados o *Sequential*, *Conv2D*, *MaxPooling2D*, *Dropout*, *Flatten* e *Dense* do *Keras*, classes que normalmente são usadas no desenvolvimento de redes convolucionais (*CNNs*).

Por fim, também foi importado o *ImageDataGenerator* que permite aplicar algumas técnicas nas imagens como rotação, redimensionamento e inversão.

```
metadata = pd.read_csv('../Dataset/Data_Entry_2017_1.csv')  
image_dir = '../Dataset/images/'
```

Neste bloco de código as informações do *dataset* são atribuídas à variável *metadata*, e em seguida, é definido o caminho para as imagens de treino.

```
metadata = metadata[['Image Index', 'Finding Labels']]  
metadata['Finding Labels'] = metadata['Finding Labels'].str.strip()  
metadatafilter = metadata[metadata['Finding Labels'] != 'No Finding']
```

Na primeira linha o *metadata* é filtrado para apresentar apenas as colunas de *Image Index* e *Finding Labels*. Logo de seguida, são removidos os espaços em branco da coluna de *Finding Labels*. Por fim é criado o *metadatafilter* que contém os dados do *metadata* filtrados, de modo a que apenas sejam apresentadas as imagens onde as *Finding Labels* sejam diferentes de *No Finding*. Desta forma o modelo será apenas treinado com os restantes rótulos como o pretendido.

```
labels=metadatafilter["Finding  
Labels"].str.get_dummies(sep='|').columns.tolist()  
metadatafilter['Encoded Labels'] = metadatafilter["Finding  
Labels"].apply(lambda x: [1 if label in x else 0 for label in labels])  
encoded_labels = pd.DataFrame(metadatafilter['Encoded Labels'].to_list(),  
columns=labels)
```

Foi criada uma lista *labels* a partir da coluna *Finding Labels* do *metadatafilter*, a função que é aplicada converte os valores presentes nas colunas para um valor binário tendo em conta que a separação usada é a |.

Na linha seguinte é acrescentada a coluna *Encoded Labels* ao *metadatafilter*, é aplicada a função *apply* que retorna o valor 0 caso a doença não esteja presente na imagem e 1 caso isso se verifique, aplica isto para todas as imagens presentes no *dataset*, retornando no fim para cada uma, um *array* de zeros e uns com as respetivas identificações.

Por último estes *arrays* são guardados na *encoded_labels*.

```
metadatafilter = pd.concat([metadatafilter, encoded_labels], axis=1)

train_metadata = metadatafilter[['Image Index'] + labels]
```

Neste bloco de código é feita a concatenação do *metadatafilter* com o *encoded_labels* e retorna o resultado no *metadatafilter*, este *metadatafilter* serve apenas como uma variável de controle pois será usada para perceber se o *encoding* foi feito de forma correta.

De seguida, foi definido o *train_metadata*, este contém os dados que serão usados no treino do modelo, onde se encontra presente a coluna *Image Index* do *metadatafilter* e as *labels* que correspondem aos rótulos.

```
datagen = ImageDataGenerator(
    rescale=1./255,
    featurewise_center=True,
    featurewise_std_normalization=True,
    preprocessing_function=tf.image.rgb_to_grayscale
)
```

Este código define uma instância da classe *ImageDataGenerator* da biblioteca *Keras*, normalmente usada para aumentar a qualidade de dados através do pré-processamento.

Analisando os parâmetros usados:

- *rescale=1./255*: É utilizado para redimensionar os valores dos pixels das imagens. Neste caso, ele divide os valores do pixel por 255 para normalizá-los dentro do intervalo de 0 a 1. Esta etapa de redimensionamento é frequentemente executada para garantir que os valores dos pixels estejam num intervalo adequado para o treino de redes neuronais;
- *featurewise_center=True*: Indica que se deve subtrair a média do conjunto de dados de cada característica (pixel) nas imagens. Ao defini-lo como *True*, o gerador calculará a média de todo o conjunto de dados e subtrai-la-á em cada imagem. Esta etapa ajuda a centralizar os dados em torno de zero, o que pode melhorar a convergência do modelo durante o treino;
- *featurewise_std_normalization=True*: Determina se cada característica (pixel) nas imagens deve ser dividida pelo seu desvio padrão. Quando definido como *True*, o gerador calcula o desvio padrão do conjunto de dados e divide os pixels de cada imagem por esse valor. Este passo ajuda a normalizar os dados e a colocá-los numa escala semelhante, ajudando o processo de aprendizagem;
- *preprocessing_function=tf.image.rgb_to_grayscale*: Permite especificar uma função personalizada a aplicar a cada imagem antes de quaisquer outros passos de pré-processamento. Neste caso, a função *tf.image.rgb_to_grayscale* da biblioteca *TensorFlow* é usada. Esta função converte as imagens RGB (*Red, Green, Blue*) para

tons de cinzento, o que pode ser útil para reduzir a complexidade dos dados de entrada e o modelo concentrar-se na informação de intensidade em vez da cor.

Ao criar uma instância de *ImageDataGenerator* com estas definições, o código configura um gerador que irá pré-processar e aumentar as imagens utilizando o redimensionamento, a centragem, a normalização e a conversão para tons de cinzentos. Este gerador pode então ser usado para gerar imagens pré-processadas que serão usadas para treino.

```
train_generator = datagen.flow_from_dataframe(  
    dataframe=train_metadata,  
    directory=image_dir,  
    x_col='Image Index',  
    y_col=labels,  
    batch_size=32,  
    shuffle=True,  
    class_mode='raw',  
    target_size=(256, 256)  
)
```

Este código cria um gerador para dados de treino ao utilizar a instância *datagen* do *ImageDataGenerator*. Ele gera as imagens pré-processadas e os seus respetivos rótulos a partir de um *DataFrame*, neste caso o *train_metadata*, criado anteriormente. Os parâmetros utilizados no método *flow_from_dataframe* foram os seguintes:

- *dataframe=train_metadata*: Este parâmetro especifica o *DataFrame* que contém os meta dados e rótulos das imagens que serão usados para o treino;
- *directory=image_dir*: Indica o diretório onde as imagens estão localizadas. É normalmente utilizado quando as imagens são armazenadas em ficheiros separados em vez de estarem diretamente presentes no *dataset*, neste caso o diretório é o *image_dir* que foi definido inicialmente;
- *x_col='Image Index'*: Este parâmetro especifica o nome da coluna no *DataFrame* (*train_metadata*) que contém os nomes ou caminhos dos ficheiros de imagem. Neste caso, é definido como *'Image Index'*, indicando que a coluna *'Image Index'* no *train_metadata* contém os nomes dos ficheiros de imagem;
- *y_col=labels*: O parâmetro *y_col* especifica a coluna no *DataFrame* que contém os rótulos de cada imagem. Neste caso os nossos rótulos são as *labels*;
- *batch_size=32*: O parâmetro *batch_size* determina o número de imagens e rótulos a serem incluídos em cada lote durante o treino. Neste caso, cada lote irá conter 32 amostras;
- *shuffle=True*: O parâmetro *shuffle* indica se a ordem das imagens e os seus rótulos correspondentes deve ser baralhado em cada época. Definindo-o como verdadeiro garante que os dados são baralhados aleatoriamente, o que pode ajudar a melhorar o processo de aprendizagem;

- *class_mode='raw'*: O parâmetro *class_mode* especifica o tipo de rótulos a gerar. Definindo-o como *'raw'* significa que os rótulos serão devolvidos como uma matriz, mantendo o seu formato original sem qualquer pré-processamento ou codificação adicional;
- *target_size=(256, 256)*: O parâmetro *target_size* define o tamanho para o qual as imagens de entrada serão redimensionadas. Neste caso, as imagens serão redimensionadas para uma forma quadrada de 256x256 pixels.

Ao chamar o *flow_from_dataframe* com estes parâmetros, é criado o *train_generator*, que pode ser usado para gerar lotes de imagens pré-processadas e os seus rótulos correspondentes que, posteriormente, serão usados para treinar um modelo.

```

classification_model = Sequential()

classification_model.add(Conv2D(32, (3, 3), activation='relu',
input_shape=(256, 256, 3)))

classification_model.add(MaxPooling2D(pool_size=(2, 2)))

classification_model.add(Conv2D(64, (3, 3), activation='relu'))

classification_model.add(MaxPooling2D(pool_size=(2, 2)))

classification_model.add(Conv2D(128, (3, 3), activation='relu'))

classification_model.add(MaxPooling2D(pool_size=(2, 2)))

classification_model.add(Flatten())

classification_model.add(Dense(512, activation='relu'))

classification_model.add(Dropout(0.5))

classification_model.add(Dense(len(labels), activation='sigmoid'))

```

Neste bloco de código é definido o nosso modelo de classificação utilizando a classe *sequential model* da biblioteca *Keras*. Em vez de um modelo pré feito, foi construído um, que era o objetivo principal, a ideia é criar um modelo personalizado que consiga identificar com a maior precisão possível as diferentes doenças (rótulos) das imagens analisadas. Foi um modelo criado através de algum trabalho de investigação para perceber qual seria a melhor abordagem para o tipo de problema que o modelo vai enfrentar. Passando à análise da arquitetura do modelo e das suas camadas:

- *Sequential*: Na primeira linha é criada uma instância de um *sequential model* que permite posteriormente adicionar camadas a este mesmo modelo;
- *Conv2D*: Esta camada executa uma operação de convolução 2D nos dados de entrada e aplica um conjunto de filtros para extrair recursos da imagem de entrada. O 32 indica o número de filtros e (3, 3) representa o tamanho do *kernel* do filtro;

- MaxPooling2D: Esta camada realiza o agrupamento máximo 2D, que reduz a resolução da representação de entrada, de forma a obter o valor máximo em cada janela de agrupamento. Ajuda na redução das dimensões espaciais e na extração de características dominantes;
- Conv2D: Outra camada convolucional com 64 filtros e (3, 3) tamanho do *kernel*;
- MaxPooling2D: Outra camada de *pooling* máximo;
- Conv2D: Uma terceira camada convolucional com 128 filtros e (3, 3) tamanho do *kernel*;
- MaxPooling2D: Uma terceira camada de *pooling* máximo;
- *Flatten*: Esta camada faz a nivelção dos mapas de recursos 2D num vetor 1D e prepara os dados para as camadas subseqüentes totalmente ligadas;
- Dense: Uma camada totalmente ligada com 512 neurónios. Executa uma transformação linear nos dados de entrada usando pesos e tendências e aplica a função de ativação *ReLU* para introduzir a não linearidade;
- *Dropout*: Esta camada ajuda a evitar o *overfitting* definindo aleatoriamente uma fração (0,5) das unidades de entrada como 0 durante o treino e atua como uma técnica de regularização;
- Dense: A camada final totalmente ligada com neurónios produz as previsões de saída para cada classe utilizando a função de ativação *sigmoide*. Os valores de saída estão entre 0 e 1, representando a probabilidade de a entrada pertencer a cada classe.

```
def signal_handler(signal, frame):
    combined_model.save('./Modelos/UnfinishedTrainedModel.h5') # Replace
    with the desired path and filename
    print('Model saved.')
    sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)
```

Foi criada uma função para guardar o modelo caso o treino fosse interrompido a meio, isto é, caso o treino fosse cancelado a qualquer altura, o modelo treinado até ao momento era guardado. Isto serve para haver um maior controlo durante o treino caso a performance do mesmo comece a decrescer em vez de crescer. Neste caso o modelo será guardado na pasta de Modelos com o nome *UnfinishedTrainedModel.h5*.

```
classification_model.compile(loss='binary_crossentropy',
    optimizer='adam', metrics=['accuracy'])
```

Foi utilizado o *compile* da biblioteca *Keras*. A função de compilação é usada para definir a configuração do modelo para o treino, especificando a função de perda, o otimizador e as métricas de avaliação:

- *loss='binary_crossentropy'*: o parâmetro *loss* determina a função de perda a ser usada durante o treino. Neste caso, é escolhida a função de perda de entropia cruzada binária. A entropia cruzada binária é normalmente utilizada em problemas de classificação binária, em que cada amostra pertence a uma de duas classes. Mede a dissimilaridade entre os rótulos verdadeiros e as probabilidades previstas;
- *optimizer='adam'*: o parâmetro otimizador especifica o algoritmo de otimização utilizado para atualizar os pesos do modelo durante o treino. Aqui é escolhido o otimizador 'adam', um otimizador popular que combina ideias do algoritmo de gradiente adaptativo (*AdaGrad*) e da propagação do quadrado médio (*RMSProp*). Adapta a taxa de aprendizagem para cada parâmetro, levando a atualizações eficientes e eficazes;
- *metrics=['accuracy']*: o parâmetro *metrics* especifica as métricas de avaliação utilizadas para monitorizar o desempenho do modelo durante o treino. Neste caso, foi utilizada a métrica 'accuracy', de forma a medir a proporção de amostras classificadas corretamente.

Esta etapa prepara o modelo para treino, especificando como ele aprenderá com os dados e como o seu desempenho será avaliado durante o mesmo.

```
combined_model.fit(train_generator, epochs=100)
combined_model.save('../Modelos/TrainedModel.h5')
```

Na primeira linha de código é usado o método *fit* para treinar o modelo. O *train_generator* é definido como os dados de treino e o parâmetro *epochs* é igualado a 100, isto é, durante o treino o modelo irá passar por este conjunto de dados 100 vezes.

Por fim, o modelo é guardado com o nome *TrainedModel* na pasta de Modelos.

4.3.1 Resultados do Treino

Durante o treino foi visto um crescimento de *accuracy* e uma diminuição de *loss* em aproximadamente 95% das *epochs*, o modelo cresceu de forma constante, tendo uma *accuracy* de 0.2234 e um *loss* de 1.1998 no final da primeira época (Figura 12) (*epoch*).

```
Epoch 1/100  
1618/1618 [=====] - 2560s 2s/step - loss: 1.1998 - accuracy: 0.2234
```

Figura 12 - Resultados do treino na 1ª epoch

Na época 50, a *accuracy* já tinha um valor de 0.4479 e contava com uma *loss* de 0.1940, um valor bem diferente da época 1 (Figura 13).

```
Epoch 50/100  
1618/1618 [=====] - 2524s 2s/step - loss: 0.1940 - accuracy: 0.4479
```

Figura 13 - Resultados do treino na 50ª epoch

Por fim o modelo na época 100, obteve uma *accuracy* de 0.5002 e uma *loss* de 0.1632 (Figura 14).

```
Epoch 100/100  
1618/1618 [=====] - 2546s 2s/step - loss: 0.1632 - accuracy: 0.5002
```

Figura 14 — Resultados do treino na última epoch

Pode-se concluir que o treino foi positivo visto que o modelo manteve uma constante aprendizagem, apesar de mesma ter sido mais baixa nas últimas épocas como o expectável, o modelo não deixou de melhorar. Comparando ao modelo referenciado no capítulo 3.2.3, podemos perceber que houve uma melhoria significativa no treino em comparação a essa solução já existente.

Depois de concluído este treino, o modelo foi avaliado num *dataset* específico para testes onde obteve os seguintes resultados (Figura 15).

```
Classification Model:  
Accuracy: 0.86  
Precision: 0.86  
Recall: 0.50  
F1-Score: 0.36
```

Figura 15 - Análise do Modelo

Em resumo, o modelo possui uma boa precisão apesar dos valores de *Recall* e *F1-Score* serem baixos, o que significa que o modelo tem algumas dificuldades em identificar a doença correta em alguns casos. Isto deve-se ao facto de alguns rótulos (doenças) não estarem presentes tão frequentemente no *dataset* usado para treino como os restantes.

4.4 Implementação do Modelo (2ª Versão)

Foi produzida uma segunda versão do código para comparação, nesta segunda e última versão foram incluídos todos os dados do *dataset*, abrangendo também aqueles cujo rótulo era *No Finding*. Estes tinham sido removidos no primeiro treino para haver um maior equilíbrio entre estes rótulos visto que mais de 50% pertenciam a classe de *No Finding*.

Em termos de código, manteve-se o mesmo sendo esta a única alteração.

```
metadatafilter = metadata[metadata['Finding Labels'] != 'No Finding']
```

Foi removida esta linha que estava presente no primeiro código e os *metadatafilter* que estavam presentes inicialmente passaram a chamar-se *metadata* pois não houve a filtragem dos rótulos e foram usadas todas as imagens.

Depois disto o modelo foi treinado com todo o conjunto de dados.

4.4.1 Resultados do Treino (2ª Versão)

Tal como esperado apesar de os resultados serem positivos, notou-se um grande desequilíbrio do modelo visto que o mesmo tinha dificuldades em identificar os rótulos que correspondiam a doenças. Apesar da *accuracy* ser boa (Figura 16), o modelo falhava nesta deteção tornando o mesmo inconsistente na análise de imagens.

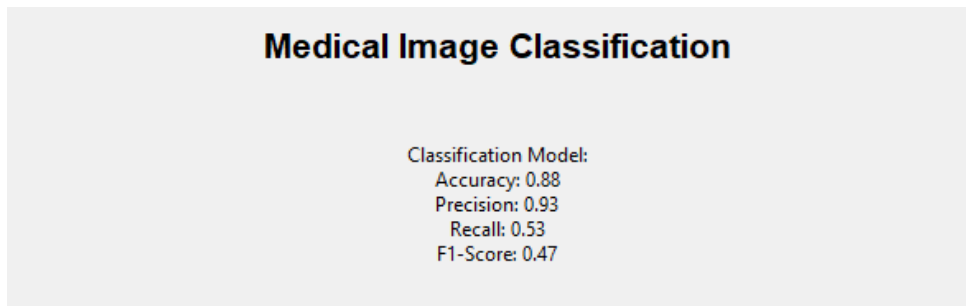


Figura 16 – Análise do modelo (2ª Versão)

Apesar de ter um bom nível de classificação quando uma imagem não continha doenças, o mesmo tinha bastante dificuldade em prever caso esta tivesse.

4.5 Implementação da Interface Gráfica

```
import tkinter as tk
from tkinter import filedialog, messagebox
from PIL import Image, ImageTk
from keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score
import cv2
```

Tal como na implementação anterior, o primeiro passo foi importar as bibliotecas necessárias para o desenvolvimento desta interface, a maior parte delas já explicadas anteriormente. A biblioteca que permite fazer esta interface é a *tkinter*.

Da *tkinter* foi importado o *filedialog* e a *messagebox* que permite a interação do utilizador com o upload de imagens e as mensagens de aviso.

A *Image* e o *ImageTK* permitem mostrar imagens na interface, através da compatibilidade entre a PIL e a *Tkinter*.

O *load_model* serve para importar um modelo treinado.

A biblioteca *numpy*, fornece suporte para grandes matrizes multidimensionais e várias funções matemáticas.

Através da biblioteca *sklearn.metrics* são importadas as diferentes métricas que permitem avaliar o desempenho do modelo.

Por último, o *cv2* é usado para tarefas de visão computacional, fornece várias funções e algoritmos para o processamento de imagem.

```
root = tk.Tk()
root.title("Medical Image Classification")
root.geometry("600x700")
root.resizable(False, False)
```

Este código configura a janela principal de uma interface gráfica de utilizador (GUI) utilizando a biblioteca *Tkinter*, previamente importada.

Na primeira linha foi criada uma instância da classe *Tk*, que representa a janela principal ou a janela raiz da aplicação GUI. É atribuída à variável *root*.

Na linha seguinte, foi definido o título da janela como "*Medical Image Classification*". O título será apresentado na parte superior da janela.

Logo de seguida foi definido o tamanho inicial da janela que será 600 píxéis de largura e 700 píxéis de altura. As dimensões são especificadas como uma *string* no formato "*widthxheight*".

Na última linha deste bloco é definida a propriedade redimensionável da janela. Ao passar *False* como argumento, ela desativa a capacidade de redimensionar a janela horizontalmente e verticalmente. O tamanho da janela permanece fixo durante a utilização da *interface*.

Ao executar este código, uma janela GUI é criada com as propriedades de título, tamanho e redimensionamento. Esta serve de ecrã principal para a interface de utilizador.

```

title_label = tk.Label(root, text="Medical Image Classification",
font=("Helvetica", 16, "bold"))
title_label.pack(pady=10)
single_image_button = tk.Button(root, text="Upload Image",
command=upload_single_image, padx=10, pady=10)
dataset_button = tk.Button(root, text="Upload Dataset",
command=upload_dataset, padx=10, pady=10)
return_button = tk.Button(root, text="Return to Main Menu",
command=return_to_main_menu, padx=10, pady=10)
img_label = tk.Label(root)
img_label.pack()
imageMM_path = "../Dataset/imgMM.png"
title_image = Image.open(imageMM_path)
title_image = title_image.resize((200, 200))
title_image = ImageTk.PhotoImage(title_image)
title_image_label = tk.Label(root, image=title_image)
title_image_label.pack()
result_label = tk.Label(root, text="", padx=10, pady=10)
result_label.pack()
info_label = tk.Label(root, text="", padx=10, pady=10)
info_label.pack()
process_button = tk.Button(root, text="Process", command=process_data,
padx=10, pady=10)
single_image_button_uploadagain = tk.Button(root, text="Upload Another
Image", command=upload_single_image, padx=10, pady=10)
dataset_button_uploadagain = tk.Button(root, text="Upload Another Dataset",
command=upload_dataset, padx=10, pady=10)
switch_menu("main_menu")
center_window(root)
root.mainloop()

```

Neste bloco de código são criados todos os elementos que serão mostrados na interface. Estes elementos incluem botões, *labels* e até uma imagem que será a imagem de marca desta GUI. Ainda é definido o menu inicial e é feita a chamada de algumas funções.

```

disease_names = ['Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema',
'Effusion', 'Emphysema', 'Fibrosis', 'Hernia', 'Infiltration', 'Mass',
'Nodule', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax']

classification_model = load_model('../Modelos/ClassificationModel.h5')

```

Foi criada a *disease_names* que contém todas as doenças que o modelo pode detetar e de seguida, o modelo treinado foi importado através do *load_model*.

```

current_menu = "main_menu"

img_path = ""

csv_file_path = ""

metadata = None

```

Foram definidas algumas variáveis iniciais e atribuídos os seus respetivos valores.

```

def center_window(window):
    window.update_idletasks()
    width = window.winfo_width()
    height = window.winfo_height()
    x = (window.winfo_screenwidth() // 2) - (width // 2)
    y = (window.winfo_screenheight() // 2) - (height // 2)
    window.geometry(f"{width}x{height}+{x}+{y}")

```

Foi produzida uma função para centrar a interface gráfica no ecrã, desta forma, sempre que a aplicação é usada será aberta no centro do ecrã.

```

def switch_menu(menu_name):
    global current_menu
    current_menu = menu_name
    if menu_name == "main_menu":
        title_image_label.pack()
        single_image_button.pack(padx=10, pady=10)
        dataset_button.pack(padx=10, pady=10)
        return_button.pack_forget()
        dataset_button_uploadagain.pack_forget()
        single_image_button_uploadagain.pack_forget()
        process_button.pack_forget()
    elif menu_name == "single_image_menu":
        dataset_button.pack_forget()
        single_image_button.pack_forget()
        process_button.pack(padx=10, pady=10)
        single_image_button_uploadagain.pack(padx=10, pady=10)
        return_button.pack(padx=10, pady=10)
        title_image_label.pack_forget()
    elif menu_name == "dataset_menu":
        result_label.pack()
        process_button.pack(padx=10, pady=10)
        dataset_button_uploadagain.pack(padx=10, pady=10)
        single_image_button.pack_forget()
        dataset_button.pack_forget()
        title_image_label.pack_forget()
        return_button.pack(padx=10, pady=10)

```

Pelo bloco de código acima pode-se perceber que foi criada uma função que controla os menus da interface. Esta interface será constituída por 3 menus, sendo eles o menu inicial, o menu onde é feita a análise das imagens e, por último, um terceiro menu onde serão mostrados os

resultados da avaliação do modelo através de *datasets* de teste. Também é efetuado o controlo dos botões para cada menu visto que são diferentes para cada um.

```
def upload_single_image():
    global img_path
    img_path = filedialog.askopenfilename()
    if img_path:
        switch_menu("single_image_menu")
        display_image()
```

A função abre um *filedialog* que permite ao utilizador seleccionar uma imagem para análise e depois desta ser seleccionada, executa a função *display_image* e muda para o menu *single_image_menu*.

```
def display_image():
    img = Image.open(img_path)
    img.thumbnail((256, 256))
    img = ImageTk.PhotoImage(img)
    img_label.configure(image=img)
    img_label.image = img
```

Caso seja escolhida uma foto válida na função *upload_single_image*, no menu *single_image_menu* é mostrada a foto seleccionada para o utilizador perceber se escolheu a foto correcta.

```

def upload_dataset():
    global csv_file_path
    global dir_path
    global imageCount
    global info_text
    csv_file_path = filedialog.askopenfilename(title="Select CSV File",
filetypes=[("CSV Files", "*.csv")])
    global metadata
    metadata = pd.read_csv(csv_file_path)
    if "Image Index" not in metadata.columns:
        messagebox.showerror("Error", "Invalid CSV file format! Missing
'Image Index' column.")
        return
    if "Finding Label" not in metadata.columns:
        messagebox.showerror("Error", "Invalid CSV file format! Missing
'Finding Label' column.")
        return
    dir_path = filedialog.askdirectory(title="Select Dataset Directory")
    count_metadata = pd.read_csv(csv_file_path)
    imageCount = len(count_metadata)
    info_text = f"Number of images found in the Dataset: {imageCount}"
    info_label.configure(text=info_text)
    if csv_file_path and dir_path:
        switch_menu("dataset_menu")

```

Caso seja selecionado o botão *dataset_button* é executada a função acima. Esta função, tal como a *upload_single_image*, abre um *filedialog* para o utilizador selecionar o *dataset* de teste para o modelo e a respetiva pasta onde estão as imagens que serão usadas por este *dataset*. De seguida é armazenada toda a informação do *dataset* no *metadata*. A função ainda avalia se no *dataset* selecionado existem as colunas *Image Index* e *Finding Label* que contêm o nome das imagens e respetivos rótulos, sendo que estas colunas podem mudar de *dataset* para *dataset*. Atribui a mensagem a apresentar na *info_text* com a quantidade de imagens encontradas no *dataset* selecionado e, por fim, caso tudo seja válido muda para o menu *dataset_menu*.

```

def return_to_main_menu():
    switch_menu("main_menu")
    img_label.configure(image="")
    result_label.configure(text="")
    info_label.configure(text="")

```

Foi concebida uma função que retorna ao menu inicial e limpa todas as *labels* da interface, esta é chamada pelo botão *return_button*.

```

def process_data():
    if current_menu == "single_image_menu":
        process_single_image()
    elif current_menu == "dataset_menu":
        process_dataset()

```

O botão *process_button* chama a função acima. Esta função é responsável por validar em que menu o utilizador se encontra e chamar a respetiva função. Neste caso, se o utilizador se encontrar no menu *single_image_menu* a função chamada será a *process_single_image* e caso seja o *dataset_menu* será chamada a *process_dataset*.

```

def process_single_image():
    img_array = np.array(Image.open(img_path).resize((256, 256)))
    img_array = cv2.cvtColor(img_array, cv2.COLOR_GRAY2RGB)
    img_array = img_array.astype('float32') / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    classification_prediction = classification_model.predict(img_array)
    threshold = 0.75
    threshold_prevention = 0.45
    predicted_labels = np.flatnonzero(np.where(classification_prediction >
threshold, 1, 0))
    prevention_labels = np.flatnonzero(np.where((classification_prediction
> thresholdprevention, 1, 0))
    predicted_diseases = [disease_names[label] for label in
predicted_labels]
    preventions_diseases = [disease_names[label] for label in
prevention_labels]
    if not predicted_diseases:
        predicted_diseases = ['No Finding']
    if not preventions_diseases:
        preventions_diseases = ['No Finding']
    result_text = f"Problems: {predicted_diseases}\n Potential Problems:
{preventions_diseases}"
    result_label.configure(text=result_text)

```

Caso o botão *process_button* seja selecionado e o menu presente seja o *single_image_menu* então a função acima é executada. Inicialmente é feito o tratamento da imagem que foi selecionada para análise, de seguida, o modelo importado irá prever qual é a probabilidade de cada doença na imagem selecionada. São definidos 2 *thresholds*, um para doenças detetadas e outro para potenciais. Nestes *thresholds* são definidos os valores para os quais o modelo irá considerar se uma doença está presente ou não, estes devem ser avaliados e alterados caso necessário e devem ser revistos caso o modelo altere para obter o máximo de previsões certas possíveis. Por fim, através destes *thresholds* será verificado se há alguma probabilidade maior que os valores definidos, e caso exista, retorna o nome da doença que está presente na *disease_names*. Caso não haja retorna *No Finding*.

```

def process_dataset():
    if dir_path:
        test_metadata = pd.read_csv(csv_file_path)
        test_image_dir = dir_path
        test_metadata = test_metadata[['Image Index', 'Finding Label']]
        test_metadata['Finding Label'] = test_metadata['Finding
Label'].str.strip()
        test_metadata['Encoded Labels'] = test_metadata["Finding
Label"].apply(lambda x: [1 if label in x else 0 for label in labels])
        encoded_labels = pd.DataFrame(test_metadata['Encoded
Labels'].to_list(), columns=labels)
        test_metadata = pd.concat([test_metadata, encoded_labels], axis=1)
        test_metadata = test_metadata[['Image Index'] + labels]
        test_labels = np.array(test_metadata[labels].values)
        test_datagen = ImageDataGenerator(rescale=1. / 255)
        test_generator = test_datagen.flow_from_dataframe(
            dataframe=test_metadata,
            directory=test_image_dir,
            x_col='Image Index',
            y_col=labels,
            class_mode='raw', # No class mode
            batch_size=32,
            shuffle=False, # No shuffling for evaluation
            target_size=(256, 256),
            color_mode='rgb'
        )
        classification_predictions =
classification_model.predict(test_generator)
        classification_labels_pred = (classification_predictions >
0.50).astype(int)
        classification_accuracy = np.mean(np.equal(test_labels,
classification_labels_pred))
        classification_precision = precision_score(test_labels,
classification_labels_pred, average='macro', zero_division=1)
        classification_recall = recall_score(test_labels,
classification_labels_pred, average='macro', zero_division=1)
        classification_f1 = f1_score(test_labels,
classification_labels_pred, average='macro', zero_division=1)
        print(f"Classification Model:\nAccuracy:
{classification_accuracy:.2f}\nPrecision:
{classification_precision:.2f}\nRecall: {classification_recall:.2f}\nF1-
Score: {classification_f1:.2f}")
        result_text = f"Classification Model:\nAccuracy:
{classification_accuracy:.2f}\nPrecision:
{classification_precision:.2f}\nRecall: {classification_recall:.2f}\nF1-
Score: {classification_f1:.2f}"
        result_label.configure(text=result_text)
        info_label.configure(text="")

```

Caso o botão *process_buton* seja selecionado e o menu presente seja o *dataset_menu* então a função acima é executada. Esta função começa por ler os dados do *dataset* importados anteriormente e prepara o *metadata* que será usado no teste. Para isto faz a transformação das *Finding Label* em zeros e uns como acontece no treino do modelo. Para além disso cria um *test_datagen* através do *ImageDataGenerator* onde efetua o pré processamento de imagem e

gera uma instância de teste onde define as informações necessárias do mesmo. É efetuada a previsão do modelo para as imagens do *dataset* e, por fim, são retornados os resultados de *accuracy*, *precision*, *recall* e *f1 score*.

4.5.1 Resultados da Interface Gráfica

Com este código foi desenvolvido uma interface gráfica capaz de interagir com o utilizador e que permite analisar imagens para previsão de doenças, para além disso, permite ainda escolher um *dataset* e avaliar se o modelo é o mais adequado para a tarefa ou não.

Como dito anteriormente a interface contém 3 menus:

- Menu Inicial (Figura 17)

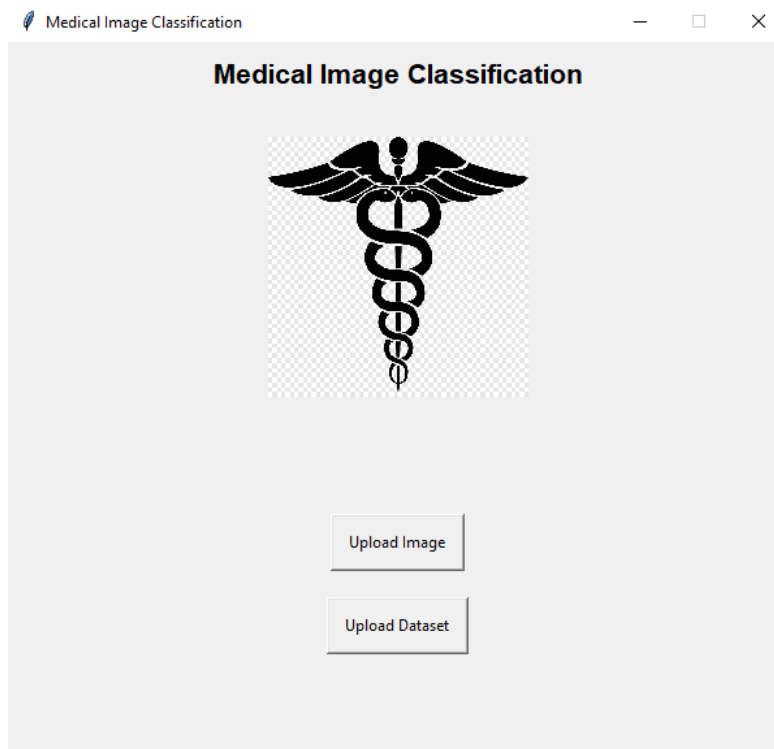


Figura 17 - Menu Inicial (UI)

- Menu de Análise de Imagem (Figura 18)

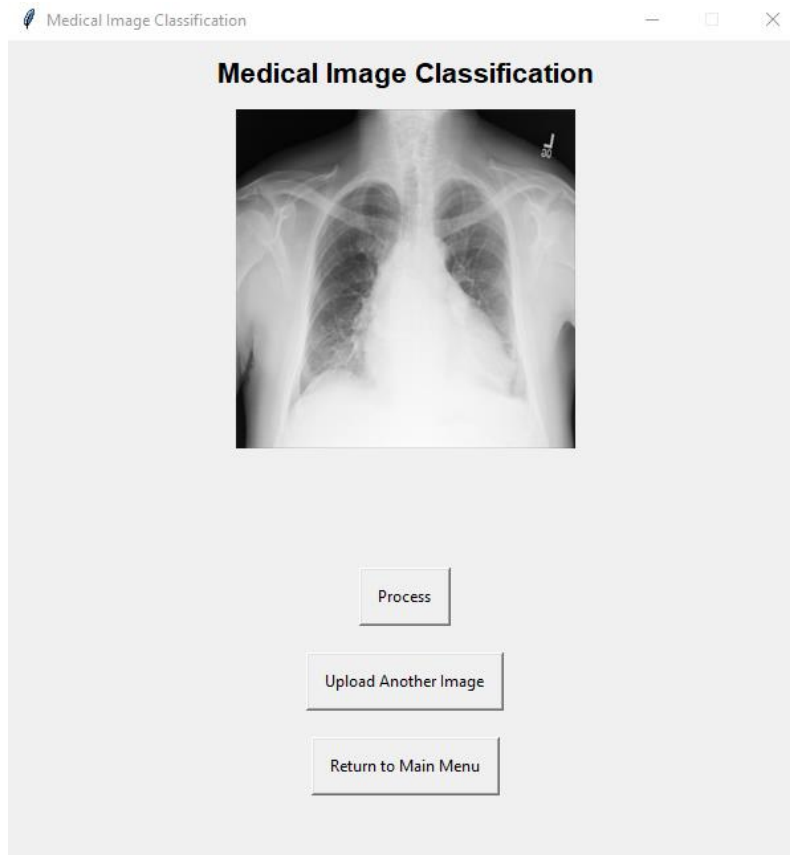


Figura 18 - Menu de Análise de Imagem (UI)

- Menu de Análise de Modelo (Figura 19)

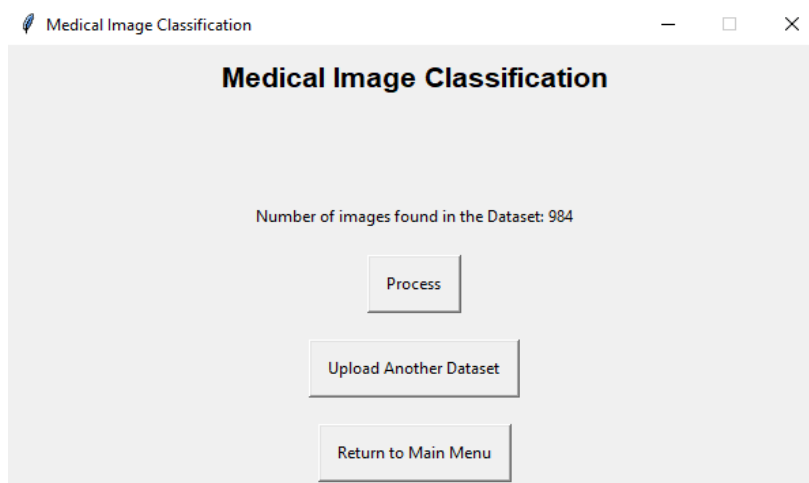


Figura 19 - Menu de Análise de Imagem (UI)

4.6 Resumo/Conclusão

Em suma, este capítulo apresenta todo o trabalho feito na implementação do modelo e da aplicação, mostra também os resultados deste desenvolvimento, principalmente na questão do modelo que consiste na base da solução.

Apesar do modelo criado não ser sempre preciso nas análises, o que faz com que este não seja o ideal em alguns casos, tendo em conta os objetivos definidos inicialmente, podemos concluir que estes foram concretizados, visto que foi criado um modelo capaz de reconhecer diferentes doenças e uma aplicação apta para interagir com o utilizador que usará este modelo criado.

A grande vantagem deste novo modelo é a sua capacidade de diferenciar bastantes doenças, tornando-a uma solução um pouco diferente das que existem atualmente. Este modelo, como referido anteriormente, em algumas destas doenças apresenta alguma inconsistência na análise e reconhecimento da mesma, por isso, poderiam ser aplicadas algumas melhorias que serão descritas mais a frente nesta dissertação.

5 Avaliação e Experimentação

Este capítulo apresenta uma avaliação e comparação de resultados de modelos já existentes com o modelo criado e detalha a forma como este analisa as imagens e retorna os respectivos resultados. Também é feita a experimentação, onde será mostrado o modelo na aplicação (interface gráfica) criada a analisar algumas destas imagens.

O capítulo começa por avaliar alguns modelos em comparação com esta solução, explica como o modelo faz o processamento de imagem e retorna os valores dos resultados. Posteriormente ainda é mencionado como este valida se deve apresentar alguma das doenças no resultado final. Para concluir são feitos alguns casos de teste para perceber o funcionamento do mesmo.

5.1 Avaliação e Comparação de Resultados

Para a avaliação da solução, foram investigados três modelos de redes neurais convolucionais profundas (*DenseNet121*, *InceptionResNetV2* e *ResNet152V2*) que foram treinados pelo mesmo conjunto de dados (*Albahli et al., 2021*). Os resultados de treino destes 4 modelos, acrescentando o modelo desenvolvido, foram os seguintes (Tabela 1).

Modelos	<i>Train_loss</i>	<i>Train_accuracy</i>	Épocas
DenseNet121	0.2596	0.4238	40
InceptionResNetV2	0.2446	0.4547	40
ResNet152V2	0.2827	0.3686	40
Modelo Desenvolvido	0.1632	0.5002	100

Tabela 1 – Resultados de Treino

Em geral os hiper parâmetros usados para o treino nestes modelos foram os mesmos usados no treino do modelo construído para a aplicação. Com isto, podemos concluir que o treino do modelo construído que apresentou uma *accuracy* de 0.5002 e uma *loss* de 0.1632, foi melhor

do que nestes modelos referidos acima, tendo em conta que este mesmo treino foi feito por 100 épocas e não 40 como os anteriores.

No que toca aos testes, o modelo desenvolvido obteve melhores resultados do que aqueles que já existem (Figura 20), sendo um dos grandes pontos positivos nesta solução.

Models	Train_loss	Train_accuracy	Val_loss	Val_accuracy
DenseNet121	0.2596	0.4238	0.2645	0.4043
InceptionResNetV2	0.2446	0.4547	0.2641	0.4102
ResNet152V2	0.2827	0.3686	0.2821	0.3760

Figura 20 - Resultados da accuracy dos modelos (Albahli et al., 2021)

Se comparamos os resultados da Figura 20 com os da Figura 11, confirmamos que o modelo desenvolvido apresentou, de facto, melhores resultados. Todos os modelos descritos anteriormente mostraram margem de melhoria pelo que poderiam ser aperfeiçoados com o devido treino para obter melhores resultados, assim como o desenvolvido.

Como já mencionado anteriormente, apesar da precisão do modelo construído ser boa, o mesmo sente dificuldades em analisar determinados tipos de doenças como *pneumothorax*, *edema* e *emphysema* (Albahli et al., 2021), assim como os modelos utilizados para comparação (Figura 21). Em contrapartida apresenta bons resultados na classificação de outras doenças e até mesmo na classificação de imagens que não apresentam qualquer tipo de problema. O mesmo foi visto nestes modelos comparativos, onde foi detetado baixíssimas precisões para algumas doenças.

Edema	DenseNet121	4.49%	4.39%	Emphysema	DenseNet121	4.79%	3.59%	Pneumothorax	DenseNet121	8.89%	8.84%
	InceptionResNetV2	4.10%	4.67%		InceptionResNetV2	4.49%	4.04%		InceptionResNetV2	9.77%	9.83%

Figura 21 – Accuracy dos modelos pré treinados para algumas doenças (Albahli et al., 2021)

Visto que o modelo concebido é capaz de prever vários problemas, pode-se concluir que o desempenho é positivo, sendo que foi melhor comparativamente às soluções encontradas onde foram utilizados o mesmo conjunto de dados, para além disso, é de realçar que a maior parte das soluções encontradas nesta vertente apenas preveem um problema, o que as torna soluções mais simples. Desta forma temos um modelo um pouco mais completo nesse aspeto com a vantagem de identificar outros tipos de doenças mesmo que às vezes não seja o objetivo inicial da análise.

Para concluir, os três modelos referenciados possuem um desempenho mais baixo nas suas funções do que a solução que foi criada, para além disso, a solução desenvolvida apresenta uma boa análise e um bom potencial para futuras melhorias tornando-a assim uma boa resposta ao problema e objetivo inicial.

5.2 Experimentação

5.2.1 Procedimento do Modelo

Para ser efetuado o procedimento de análise por parte do modelo é necessária uma imagem válida de um raio x importada pelo utilizador da aplicação.

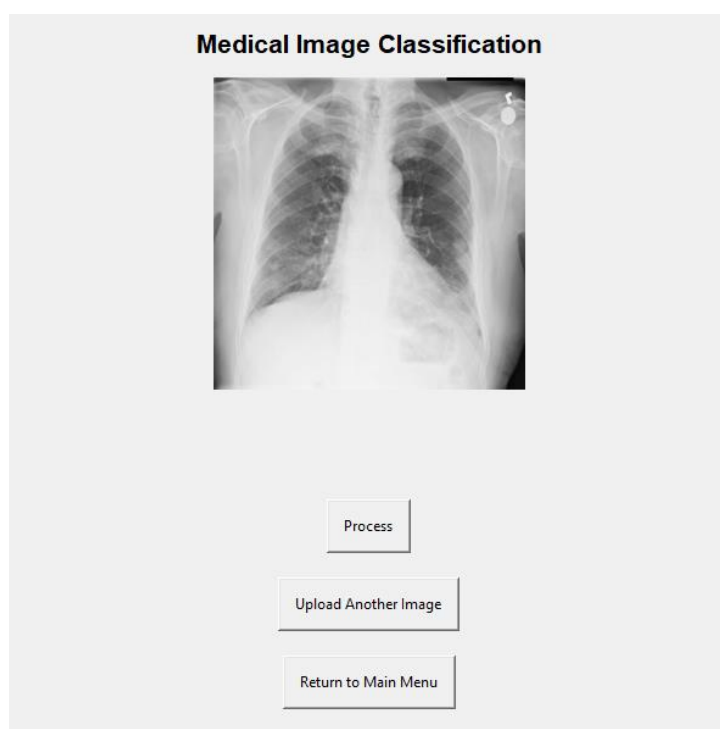


Figura 22 - Imagem para Avaliação

Depois de importada uma imagem, através do botão que faz o processamento, a mesma será analisada (Figura 22). De seguida, o modelo irá calcular qual é a probabilidade que cada doença tem de estar presente no raio x analisado.

```
Probability of : [[0. 0. 0.369 0. 0. 0. 0. 0. 0.001 0. 0. 0.
0. 0. ]]
```

Figura 23 - Probabilidade de Cada Doença

O modelo retorna um *array* com as probabilidades de cada doença, sendo a ordem a seguinte (Figura 23).

```
disease_names = ['Atelectasis', 'Cardiomegaly', 'Consolidation', 'Edema', 'Effusion', 'Emphysema', 'Fibrosis',  
'Hernia', 'Infiltration', 'Mass', 'Nodule', 'Pleural_Thickening', 'Pneumonia', 'Pneumothorax']
```

Figura 24 – Ordem de doenças

Como se pode observar nas duas imagens Figura 23 e Figura 24, o modelo detetou que havia uma probabilidade de 0.369 do raio x conter *Consolidation* e uma probabilidade de 0.001 de conter *Infiltration*. Apesar de terem sido detetadas probabilidades acima de 0.00 para algumas doenças, as mesmas não são suficientemente altas para o modelo considerar que realmente existe esse problema no raio x ou até que exista algum potencial problema futuro. Desta forma é retornada uma resposta do mesmo com essa informação (Figura 25).

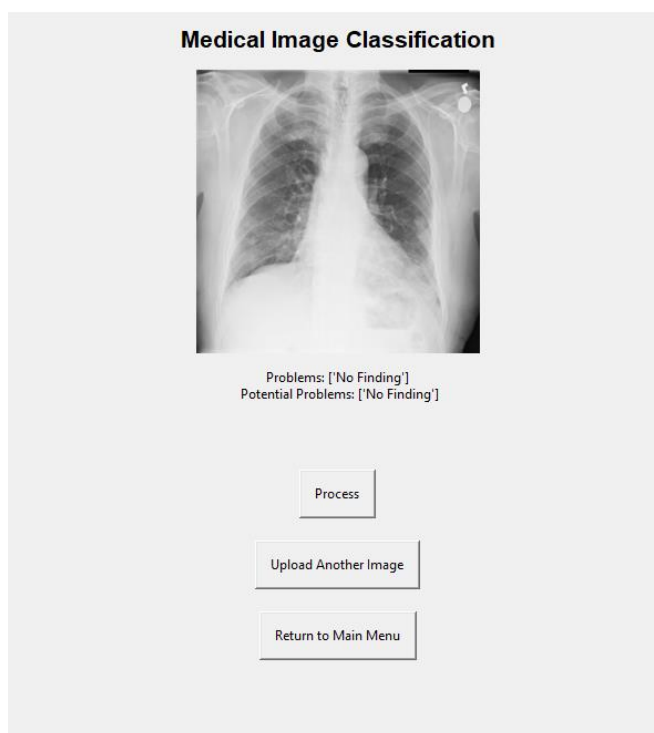


Figura 25 - Resposta do Modelo

5.2.2 Casos de Teste

Foram testadas as duas opções da aplicação criada, a primeira que analisa as imagens importadas e a segunda que analisa o modelo através de um *dataset*.

5.2.2.1 Importação de Imagem

Para testar a primeira opção foi utilizada uma imagem que não continha qualquer tipo de problema de saúde. Para isso, como já mencionado anteriormente, foi realizada a importação desta mesma imagem e, de seguida, a avaliação do modelo através do seguinte procedimento:

1º: No menu inicial foi selecionada a opção *Upload Image* (Figura 26).

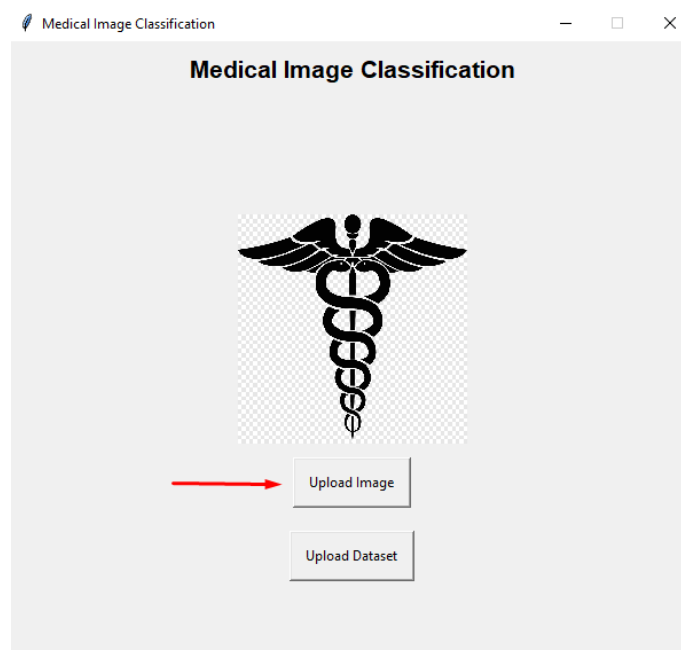


Figura 26 - Experimentação de Imagem (Menu Inicial)

2º: De seguida, selecionamos a imagem que pretendemos que seja analisada (Figura 27).

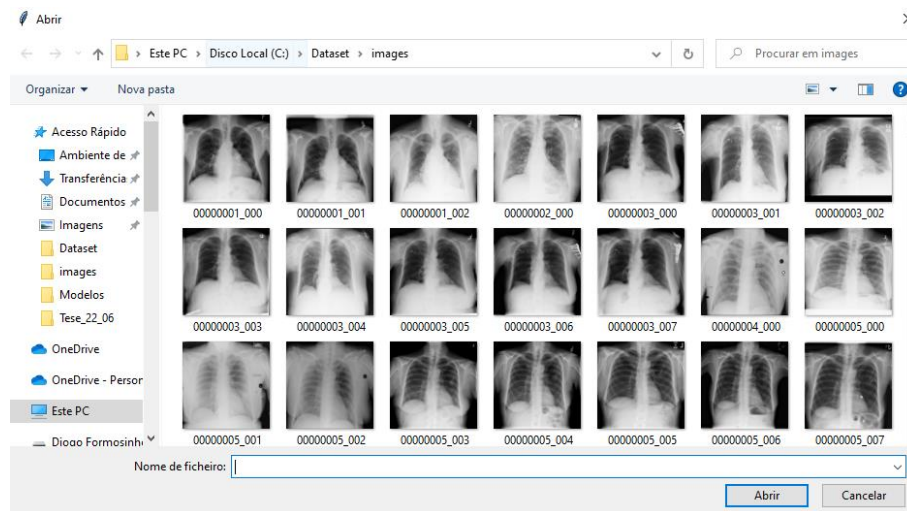


Figura 27 - Experimentação de Imagem (Selecionar Imagem)

3º: Depois de importada, podemos selecionar o botão *Process* para que a mesma seja analisada (Figura 28).

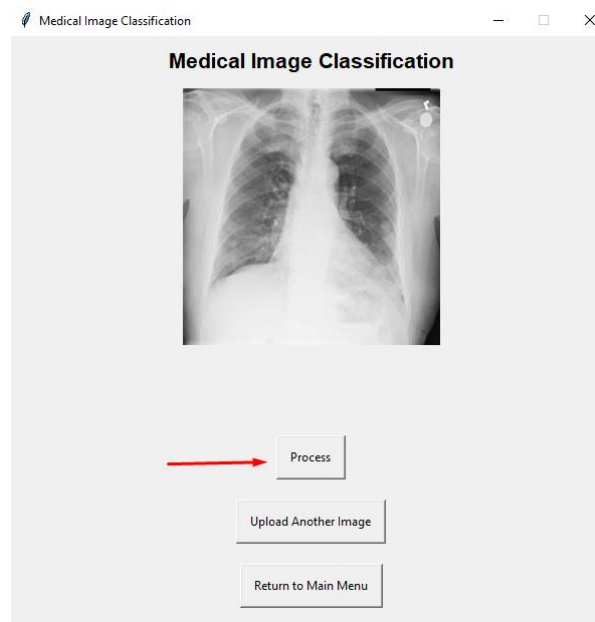


Figura 28 - Experimentação de Imagem (Processamento)

4º: Por último, é apresentado o resultado (Figura 29).

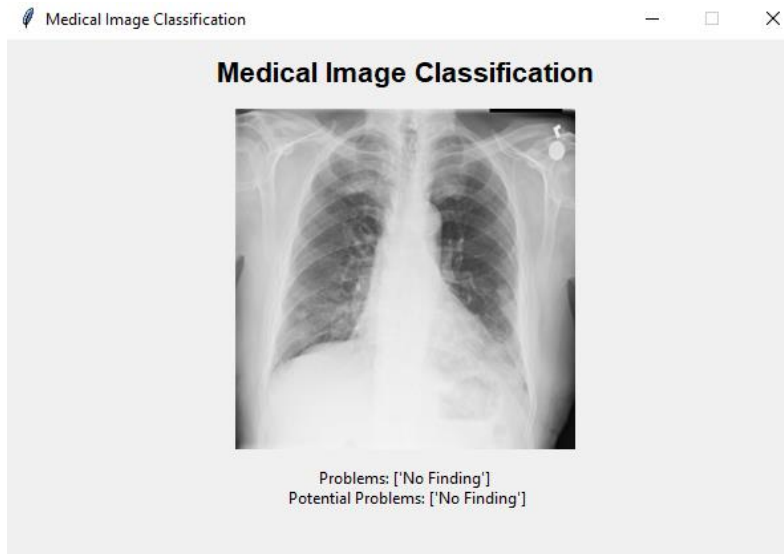


Figura 29 - Experimentação de Imagem (Resultado)

Como se pode observar, o modelo retornou uma resposta acertada para o caso analisado visto que a mesma é *No Finding*.

Foi feita uma segunda validação desta opção onde foi selecionada uma imagem que continha o problema *Consolidation*, depois dos passos demonstrados anteriormente foi retornado o seguinte resultado.

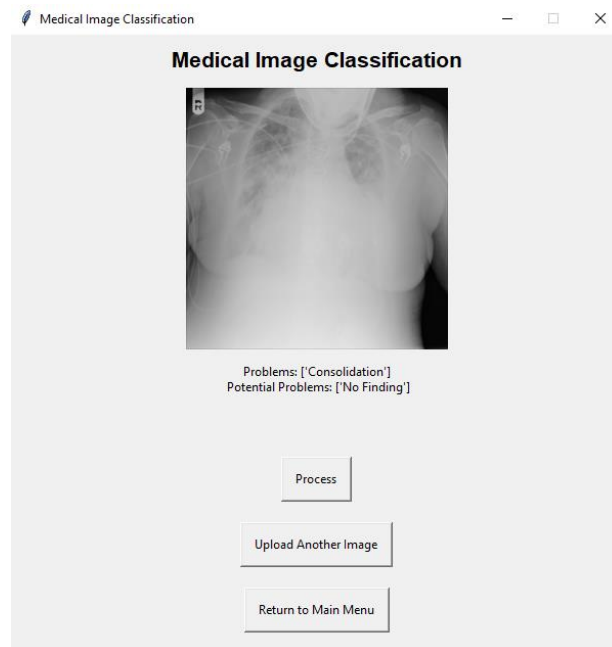


Figura 30 - Experimentação de Imagem (Segundo Caso)

Mais uma vez, como mostra a Figura 30, o modelo foi assertivo na sua análise retornando o resultado esperado.

5.2.2.2 Análise de Dataset

Para a demonstração da segunda opção da aplicação foi usado um *dataset* de testes para perceber a *performance* do modelo. Neste *dataset* encontravam-se presentes alguns rótulos que o *dataset* de treino continha com as suas respetivas imagens associadas. Para isso foi realizado o seguinte procedimento:

1º: No menu inicial foi selecionada a opção *Upload Dataset* (Figura 31).

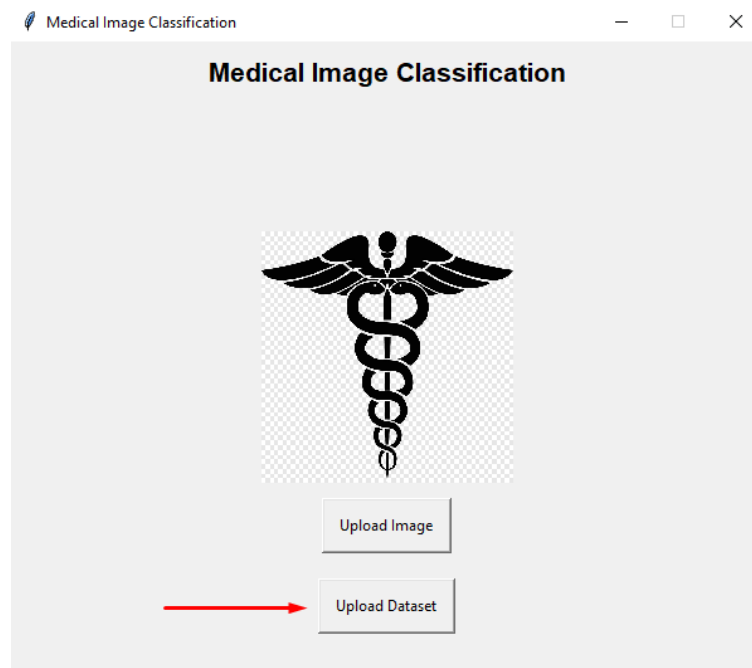


Figura 31 – Análise de Dataset (Menu Inicial)

2º: De seguida, é seleccionado o ficheiro correspondente ao *dataset* e a pasta que contém as imagens para análise (Figura 32).

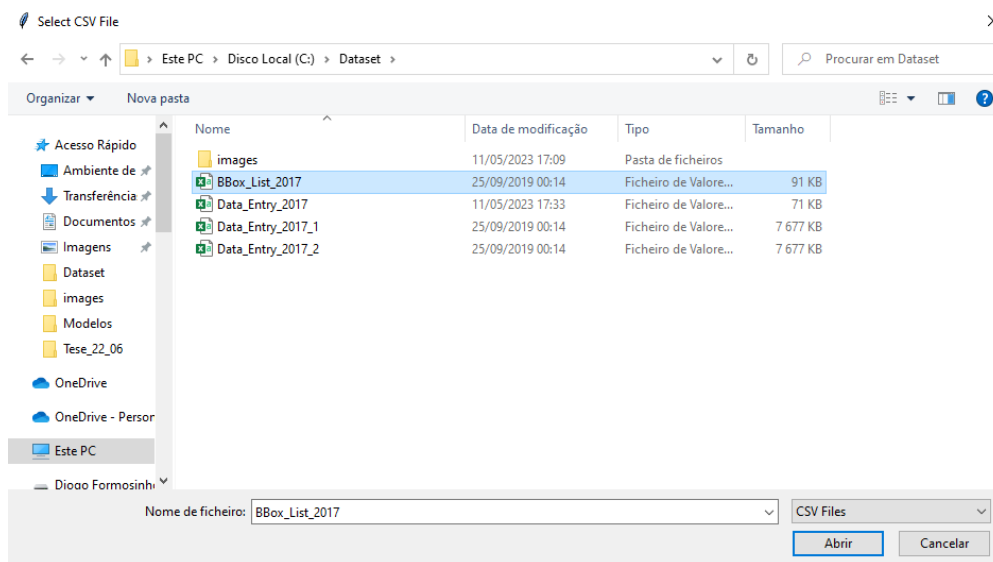


Figura 32 - Análise de Dataset (Selecionar Ficheiro e Pasta de Imagens)

3º: Caso o ficheiro seleccionado seja válido, será mostrado uma mensagem com a quantidade de imagens encontradas no *dataset* que serão usadas na análise. Depois do botão *Process* ser seleccionado o procedimento começa (Figura 33).

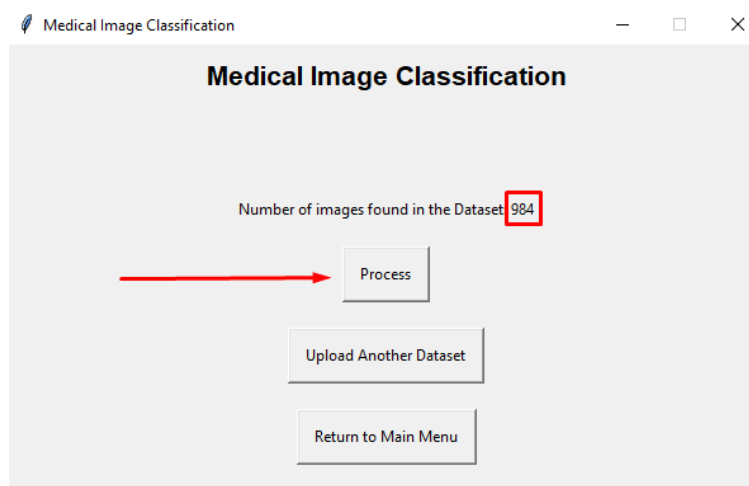


Figura 33 - Análise de Dataset (Processamento)

4º: Por último, é apresentado o resultado.

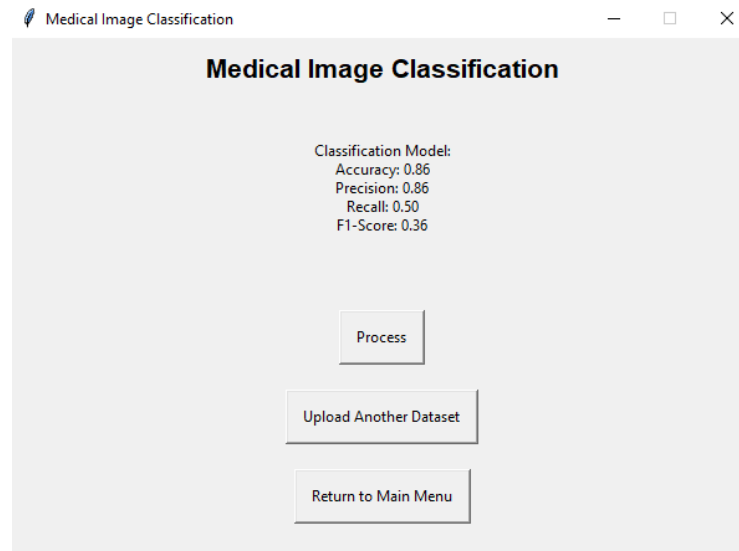


Figura 34 - Análise de Dataset (Resultado)

Através destes valores (Figura 34), conseguimos perceber se o modelo tem uma boa ou má *performance* para este conjunto de dados selecionado.

5.3 Resumo/Conclusão

Neste capítulo foi feito a comparação dos treinos e resultados do modelo construído com outros que tenham sido treinados com o mesmo conjunto de dados, referindo quais os pontos positivos e negativos destes na avaliação e classificação de imagens. De seguida foi explicado como o modelo procede para analisar uma imagem, esclarecendo os passos que o mesmo desempenha até chegar à resposta final. Por fim, a aplicação construída foi testada com algumas imagens para perceber se as avaliações feitas eram corretas ou não.

6 Conclusão

Este capítulo apresenta um resumo de todo o trabalho desenvolvido desde a investigação até ao produto final, juntamente com as conclusões. Para além disso é são mencionados os objetivos atingidos, problemas encontrados no processo e futuro trabalho.

Para este projeto foi realizada uma investigação de modo a perceber que vantagens existem na construção de um sistema que deteta problemas em radiografias e ajuda na tomada de decisão dos profissionais de saúde no que diz respeito aos tratamentos para as doenças identificadas. Para isto, foi descrito um contexto, problemas e quais seriam os contributos deste novo sistema na aplicação da área de saúde.

Tiveram de ser analisadas as técnicas e algoritmos existentes no desenvolvimento deste tipo de soluções, como redes neuronais, técnicas de processamento de imagem, entre outras. Depois desta análise foram escolhidas aquelas que melhor se adaptavam à solução desejada, neste caso, a filtragem linear e as redes neuronais convolucionais.

Foi redigido o método proposto para o sistema e o *design* para o processo da interface gráfica, descrevendo todos os passos em ambos.

Foi então criado um modelo de rede neuronal convolucional (CNN) para análise de raios x com o objetivo de identificar problemas e doenças. Ao longo do projeto, foi feito o treino de uma CNN utilizando um conjunto de dados de imagens de raios x rotuladas por especialistas médicos.

Foi mencionado que conjunto de dados foi usado neste treino e as suas características e limitações principais.

Por fim, foi demonstrada a interface gráfica construída e o procedimento para a avaliação de radiografias.

Apesar dos resultados obtidos com o modelo de CNN não serem muito altos, foram altamente promissores comparativamente aos modelos existentes previamente que usaram o mesmo conjunto de dados. Através de uma cuidadosa seleção de arquitetura de rede, técnicas de pré-processamento de dados e otimização de hiper parâmetros, foi alcançado uma taxa de precisão positiva na classificação de doenças através de raios x.

A utilização de redes neuronais convolucionais mostrou-se eficaz na deteção de problemas e doenças a partir de imagens. O modelo foi capaz de identificar com precisão condições como edema pulmonar, hérnias, nódulos, entre outras.

Depois de analisado este modelo e outros idênticos, conseguimos responder à pergunta feita inicialmente.

Podem as técnicas de *deep learning* em imagens de raio x ajudar na deteção preventiva de doenças pulmonares e na tomada de decisão por parte dos profissionais de saúde?

A resposta é sim, foram encontrados sistemas com excelentes resultados na análise de algumas doenças, apresentando valores de precisão maiores que 90% (Figura 9). As técnicas de *deep learning* mostraram boas soluções através das CNNs (Secção 2.4) que conseguem analisar imagens e retornar resultados positivos das mesmas. O sistema desenvolvido também apresenta valores positivos na deteção e classificação de algumas doenças, pelo que poderia ser usado como apoio nessa decisão médica (Secção 4.3). Apesar disso, teriam de ser feitos mais alguns treinos para tornar o mesmo mais equilibrado e preciso, visto que apresenta algumas dificuldades em analisar determinados tipos de doenças, para que desta forma, pudesse ser usado num âmbito profissional (Secção 5.1).

Os algoritmos de *deep learning* são capazes de analisar grandes quantidades de radiografias de maneira rápida e precisa, identificando padrões e características específicas associadas a diferentes doenças pulmonares, como pneumonia, tuberculose e cancro do pulmão. Esta análise automatizada pode ajudar os profissionais de saúde ao fornecer informações relevantes para deteções precoces e precisas destas doenças, permitindo assim intervenções mais rápidas aumentando as probabilidades de os tratamentos serem bem sucedidos.

6.1 Objetivos Alcançados

O principal objetivo deste projeto era a criação de um modelo que conseguisse classificar doenças através de imagens raios x, por isso, percebemos que este foi atingido pois o modelo desenvolvido tem esta capacidade de análise. Apesar do mesmo não apresentar uma precisão muito alta, tem a capacidade de reconhecer uma vasta lista de doenças, o que não está presente em muitos modelos, onde normalmente só é identificado um pequeno conjunto de problemas.

Foi também desenvolvida uma interface gráfica, que permite aos utilizadores do sistema, radiologistas e médicos especializados, usar o modelo criado para ajudar na análise de casos.

6.2 Problemas Encontrados

Foram encontrados alguns problemas durante o desenvolvimento deste projeto.

Alguns destes estavam relacionados com o *dataset* que foi usado para o treino do modelo, como mencionado anteriormente, apesar de ser um conjunto de dados bastante completo no que toca a quantidade de doenças, o mesmo apresentou resultados fracos devido ao excesso

de imagens relacionadas com algumas destas doenças. Desta forma, tornou-se num conjunto de dados desequilibrado, tornando todo este processo mais complicado. Com isto foi necessário um trabalho de revisão e análise para tornar o conjunto de dados o mais compatível possível para o treino do modelo.

6.3 Futuro Trabalho

Para o projeto em questão poderia ter sido abordado de uma forma prática, a técnica de crescimento da região, mas infelizmente, devido a algumas complicações durante o desenvolvimento do modelo, o mesmo não foi possível.

6.3.1 Modelo

As principais melhorias focam-se no modelo que é o núcleo de toda esta solução. No futuro, este modelo deveria ser treinado com outros conjuntos de dados para melhorar a sua precisão e eficácia na deteção de problemas e doenças. Deveria também ser treinado com mais imagens que contivessem doenças que inicialmente estiveram pouco presentes no *dataset* usado para treino, de forma a equilibrar a solução existente. Por fim, podia ainda ser treinado novamente com as mesmas imagens do *dataset* usado, mas individualmente para cada doença.

6.3.2 Interface Gráfica (UI)

Para a interface gráfica também poderia ser aplicado algumas melhorias. A principal e a mais interessante seria uma opção que permitisse ao utilizador trocar o modelo que a aplicação usa, desta forma, permitiria usar o modelo mais adequado caso fosse feita uma análise um pouco diferente do habitual. Para além disso, também seria interessante adicionar uma opção para permitir o treino do modelo depois do mesmo ser analisado por um conjunto de dados, dessa forma, após os resultados, caso os mesmos fossem positivos o modelo seria treinado para melhorar a sua performance.

Para concluir, este projeto demonstrou que o uso de redes neuronais convolucionais para análise de raios x é uma abordagem promissora para a deteção e diagnóstico de problemas e doenças. Com um desenvolvimento cuidadoso e validação adequada, estes tipos de modelos podem ser uma valiosa ferramenta de apoio à prática médica, proporcionando resultados rápidos e precisos para auxiliar os profissionais de saúde nas suas decisões clínicas.

Referências

- 9 Types of Neural Networks: Applications, Pros, and Cons. (n.d.). Retrieved January 5, 2023, from <https://www.knowledgehut.com/blog/data-science/types-of-neural-networks>
- Leite, J. (n.d.). 74721-Tese+Mestrado+João+Leite.
- Advantages of X-Ray | Disadvantages of X-Ray. (n.d.). Retrieved January 5, 2023, from <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-X-Ray.html>
- Aggarwal, C. C. (2018). Neural Networks and Deep Learning. *Neural Networks and Deep Learning*. <https://doi.org/10.1007/978-3-319-94463-0>
- Aggarwal, C. C., St, T., Croup, F., Chapman, A., & Book, H. (2014). *Data Classification Algorithms and Applications*.
- Albahli, S., Rauf, H. T., Alghosaibi, A., & Balas, V. E. (2021). AI-driven deep CNN approach for multi-label pathology classification using chest X-Rays. *PeerJ Computer Science*, 7, 1–17. <https://doi.org/10.7717/PEERJ-CS.495>
- Alexandre, P., Ferreira, C., & Pereira, B. (2011). *Paulo Alexandre da Costa Ferreira Brandão Pereira Erro em Imagiologia Médica Erro em Imagiologia Médica*.
- Alves, A. S., Mestre, C.-O., Carvalho, M., Pais, S., & Assistente, C. (2012). *O contributo da imagiologia no diagnóstico dos distúrbios temporomandibulares*. <https://repositorio-aberto.up.pt/handle/10216/63742>
- Apostolopoulos, I. D., & Mpesiana, T. A. (2020). Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine*, 43(2), 635–640. <https://doi.org/10.1007/S13246-020-00865-4>
- Atelectasis - Symptoms and causes - Mayo Clinic. (n.d.). Retrieved June 22, 2023, from <https://www.mayoclinic.org/diseases-conditions/atelectasis/symptoms-causes/syc-20369684>
- Bagherieh, H., Hashemi, A., & Pilevar, A. H. (2014). Mass Detection in Lung CT Images using Region Growing Segmentation and Decision Making based on Fuzzy Systems. *Image, Graphics and Signal Processing*, 1, 1–8. <https://doi.org/10.5815/ijigsp.2014.01.01>
- Briganti, G., & Le Moine, O. (2020). Artificial Intelligence in Medicine: Today and Tomorrow. *Frontiers in Medicine*, 7. <https://doi.org/10.3389/FMED.2020.00027>
- Capítulo 54 - Introdução às Redes Adversárias Generativas (GANs - Generative Adversarial Networks) - Deep Learning Book. (n.d.). Retrieved January 5, 2023, from <https://www.deeplearningbook.com.br/introducao-as-redes-adversarias-generativas-gans-generative-adversarial-networks/>
- Capítulo 58 - Introdução aos Autoencoders - Deep Learning Book. (n.d.). Retrieved January 5, 2023, from <https://www.deeplearningbook.com.br/introducao-aos-autoencoders/>

- Caseli, H. D. M. (2017). Tradução Automática: estratégias e limitações. *Domínios de Linguagem*, 11(5), 1782. <https://doi.org/10.14393/DL32-V11N5A2017-21>
- Chan, H. P., Samala, R. K., Hadjiiski, L. M., & Zhou, C. (2020). Deep Learning in Medical Image Analysis. *Advances in Experimental Medicine and Biology*, 1213, 3–21. https://doi.org/10.1007/978-3-030-33128-3_1/COVER
- Couto, L. N. (2012). *Sistema para localização robótica de veículos autônomos baseado em visão computacional por pontos de referência*. <https://doi.org/10.11606/D.55.2012.TDE-04092012-110014>
- CT Scans: *MedlinePlus*. (n.d.). Retrieved January 5, 2023, from <https://medlineplus.gov/ctscans.html>
- De Magalhães, A., & Faro, R. (2007). *UNIVERSIDADE DO ALGARVE Faculdade de Ciências e Tecnologia Manual de Tomografia Computorizada do Técnico de Imagiologia Médica (Dissertação para a obtenção do grau de Mestre em Imagiologia Médica)*.
- Demir, F., Sengur, A., & Bajaj, V. (2019). Convolutional neural networks based efficient approach for classification of lung diseases. *Health Information Science and Systems*, 8(1). <https://doi.org/10.1007/S13755-019-0091-3>
- Diagnostic Imaging: *MedlinePlus*. (n.d.). Retrieved December 22, 2022, from <https://medlineplus.gov/diagnosticimaging.html>
- Efeitos da Inteligência Artificial na Medicina | APD. (n.d.). Retrieved December 22, 2022, from <https://www.apd.pt/efeitos-da-inteligencia-artificial-na-medicina/>
- Ética na medicina: importância, interpretação e novo código ética médica. (n.d.). Retrieved January 16, 2023, from <https://telemedicinamorsch.com.br/blog/etica-na-medicina>
- Fang, J., Xu, Y., Zhao, Y., Yan, Y., Liu, J., & Liu, J. (2021). Weighing features of lung and heart regions for thoracic disease classification. *BMC Medical Imaging*, 21(1). <https://doi.org/10.1186/S12880-021-00627-Y>
- Fernando, L., & Uzai, D. O. (2018). *UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO” FACULDADE DE CIÊNCIAS-CAMPUS BAURU DEPARTAMENTO DE COMPUTAÇÃO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO RECONHECIMENTO ÓPTICO DE CARACTERES MANUSCRITOS USANDO REDES NEURAIAS CONVOLUCIONAIS*.
- Ferreira, A. (2009). *UNIVERSIDADE DO ALGARVE FACULDADE DE CIÊNCIAS E TECNOLOGIA PROTEÇÃO RADIOLÓGICA DO SERVIÇO DE RADIOLOGIA DO HOSPITAL DE FARO E.P.E. MESTRADO EM IMAGIOLOGIA MÉDICA ANA FILIPA PACHECO DO CARMO FERREIRA*.
- Filho, F. (2020). *PRISMA como metodologia para Revisão Sistemática*. | by fabiano filho | *dadosesaude* | *Medium*. <https://medium.com/dados-e-saude/prisma-como-metodologia-para-revis%C3%A3o-sistem%C3%A1tica-b3f55b4ebc5c>
- Goos, G., Hartmanis, J., Van, J., Board, L. E., Hutchison, D., Kittler, J., Kleinberg, J. M., Mattern, F., Zurich, E., Mitchell, J. C., Naor, M., Steffen, B., Terzopoulos, D., Tygar, D., & Weikum, G.

- (n.d.). Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. *Lecture Notes In*. <https://doi.org/10.1007/978-3-319-67558-9>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity Mappings in Deep Residual Networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9908 LNCS, 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
- Hesamian, M. H., Jia, W., He, X., & Kennedy, P. (2019). Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *Journal of Digital Imaging*, 32(4), 582–596. <https://doi.org/10.1007/S10278-019-00227-X/TABLES/2>
- Horváth, I. (2007). Comparison of Three Methodological Approaches of Design Research. *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007*, 361-362 (exec. Summ.), full paper no. DS42_P_341. <https://www.designsociety.org/publication/25512/Comparison+of+Three+Methodological+Approaches+of+Design+Research>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
- Hussain, S., Mubeen, I., Ullah, N., Shah, S. S. U. D., Khan, B. A., Zahoor, M., Ullah, R., Khan, F. A., & Sultan, M. A. (2022). Modern Diagnostic Imaging Technique Applications and Risk Factors in the Medical Field: A Review. *BioMed Research International*, 2022. <https://doi.org/10.1155/2022/5164970>
- Image Processing using CNN | Beginner's Guide to Image Processing*. (n.d.). Retrieved June 30, 2023, from <https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnn-a-beginners-guide/>
- Inteligência artificial na medicina: aplicações e benefícios*. (n.d.). Retrieved January 20, 2023, from <https://telemedicinamorsch.com.br/blog/inteligencia-artificial-na-medicina>
- Inteligência artificial na medicina: como ela pode ser usada? - Blog da IPEMED*. (n.d.). Retrieved January 20, 2023, from https://www.ipemed.com.br/blog/inteligencia-artificial-na-medicina-como-ela-pode-ser-usada?utm_source=google&utm_medium=organic
- Jähne, B. (n.d.). *Digital Image Processing - Bernd Jähne - Google Livros*. Retrieved July 1, 2023, from https://books.google.pt/books?hl=pt-PT&lr=&id=GVWYBrDXMNkC&oi=fnd&pg=PA3&dq=+Digital+Image+Processing.&ots=0KWMMbXLAY&sig=Sb55Bzrpoj2BXfywucP2OW1-Ct0&redir_esc=y#v=onepage&q=Digital%20Image%20Processing.&f=false
- Kasban, H., El-Bendary, M. A. M., & Salama, D. H. (2015). A Comparative Study of Medical Imaging Techniques. *International Journal of Information Science and Intelligent System*, 4(2), 37–58.

- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999–7019. <https://doi.org/10.1109/TNNLS.2021.3084827>
- Lima, J. R., Barbosa Da Costa, L., & Cury, D. (n.d.). *Reconhecimento de voz para Inclusão de deficientes visuais em ambientes virtuais de aprendizagem*. Retrieved July 3, 2023, from <http://dl.acm.org/>
- Miranda, E., Aryuni, M., & Irwansyah, E. (2017). A survey of medical image classification techniques. *Proceedings of 2016 International Conference on Information Management and Technology, ICIMTech 2016*, 56–61. <https://doi.org/10.1109/ICIMTECH.2016.7930302>
- Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2010). *Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement*. <https://doi.org/10.1016/j.ijsu.2010.02.007>
- MRI Scans: MedlinePlus*. (n.d.). Retrieved January 5, 2023, from <https://medlineplus.gov/mriscans.html>
- Nahiduzzaman, Md., Goni, Md. O. F., Hassan, R., Islam, Md. R., Syfullah, M. K., Shahriar, S. M., Anower, Md. S., Ahsan, M., Haider, J., & Kowalski, M. (2023). Parallel CNN-ELM: A multiclass classification of chest X-ray images to identify seventeen lung diseases including COVID-19. *Expert Systems with Applications*, 229, 120528. <https://doi.org/10.1016/J.ESWA.2023.120528>
- NIH Chest X-Ray Multi-Classification | Kaggle*. (n.d.). Retrieved June 22, 2023, from <https://www.kaggle.com/code/adamjgoren/nih-chest-x-ray-multi-classification>
- NIH Chest X-rays | Kaggle*. (n.d.). Retrieved June 22, 2023, from <https://www.kaggle.com/datasets/nih-chest-xrays/data>
- Nuclear Scans: MedlinePlus*. (n.d.). Retrieved January 5, 2023, from <https://medlineplus.gov/nuclearscans.html>
- Nunes, L. (n.d.). *Aspetos éticos na investigação de enfermagem*. Retrieved July 1, 2023, from www.ess.ips.pt
- O que é um Estudo PET? - SPMN – Sociedade Portuguesa de Medicina Nuclear*. (n.d.). Retrieved January 16, 2023, from <https://www.spmn.org/publico-e-utentes/o-que-e-um-estudo-pet/>
- O que são os raios X? - Brasil Escola*. (n.d.). Retrieved January 16, 2023, from <https://brasilescola.uol.com.br/o-que-e/fisica/o-que-sao-os-raios-x.htm>
- Pacheco Lacerda, D., Dresch, A., Proença, A., Valle, J. A., & Júnior, A. (n.d.). *Design Science Research: método de pesquisa para a engenharia de produção Design Science Research: a research method to production engineering*.
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M.,

- Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., Mcdonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews The Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) statement. *International Journal of Surgery*, 88, 105906. <https://doi.org/10.1016/j.ijso.2021.105906>
- Pello, R. (2018, October 31). *Design science research — a short summary | by Rauno Pello | Medium*. <https://medium.com/@pello/design-science-research-a-summary-bb538a40f669>
- Portal do INE. (n.d.). Retrieved December 22, 2022, from https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_pesquisa&frm_accas=PESQUISAR&frm_show_page_num=1&frm_modos_pesquisa=PESQUISA_SIMPLES&frm_texto=%22causas+de+morte%22&frm_modos_texto=MODO_TEXTO_ALL&frm_data_ini=&frm_data_fim=&frm_tema=QUALQUER_TEMA&frm_area=o_ine_area_Destaques&lang=pt
- Preti, G., & Dissertação, S. (2014). *Um processo para modelagem e aplicação de técnicas computacionais para detecção de fraudes em transações eletrônicas*. <https://doi.org/10.11606/D.45.2014.TDE-23052014-144032>
- Putha, P., Tadepalli, M., Reddy, B., Raj, T., Chiramal, J. A., Govil, S., Sinha, N., KS, M., Reddivari, S., Jagirdar, A., Rao, P., & Warier, P. (2018). *Can Artificial Intelligence Reliably Report Chest X-Rays?: Radiologist Validation of an Algorithm trained on 2.3 Million X-Rays*. <http://arxiv.org/abs/1807.07455>
- Qin, Z. Z., Sander, M. S., Rai, B., Titahong, C. N., Sudrungrot, S., Laah, S. N., Adhikari, L. M., Carter, E. J., Puri, L., Codlin, A. J., & Creswell, J. (2019). Using artificial intelligence to read chest radiographs for tuberculosis detection: A multi-site evaluation of the diagnostic accuracy of three deep learning systems. *Scientific Reports*, 9(1). <https://doi.org/10.1038/S41598-019-51503-3>
- Rajeshwari, D., & Shyamala, K. (2021). A Survey on Segmentation Clustering technique to detect brain tumor for MRI. *2021 International Conference on Smart Generation Computing, Communication and Networking, SMART GENCON 2021*. <https://doi.org/10.1109/SMARTGENCON51891.2021.9645865>
- Rede neural artificial – Wikipédia, a enciclopédia livre. (n.d.). Retrieved January 16, 2023, from https://pt.wikipedia.org/wiki/Rede_neural_artificial
- Region growing - Wikipedia. (n.d.). Retrieved January 5, 2023, from https://en.wikipedia.org/wiki/Region_growing
- Region Segmentation Computação Visual e Multimédia Chap. 7-Region Segmentation. (n.d.).
- Ressonância Magnética (RMN): como funciona e tipos de exames. (n.d.). Retrieved January 16, 2023, from <https://star.med.br/o-que-e-ressonancia-magnetica-rmn/>
- Rethlefsen, M. L., & Page, M. J. (2022). PRISMA 2020 and PRISMA-S: common questions on tracking records and the flow diagram. *Journal of the Medical Library Association : JMLA*, 110(2), 253. <https://doi.org/10.5195/JMLA.2022.1449>

- Sarvamangala, D. R., & Kulkarni, R. V. (2022). Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, 15(1), 1–22. <https://doi.org/10.1007/S12065-020-00540-3/FIGURES/2>
- Sharma, N., Ray, A. K., Shukla, K. K., Sharma, S., Pradhan, S., Srivastva, A., & Aggarwal, L. (2010). Automated medical image segmentation techniques. *Journal of Medical Physics / Association of Medical Physicists of India*, 35(1), 3. <https://doi.org/10.4103/0971-6203.58777>
- Sharma, P., Hans, P., & Gupta, S. C. (2020). Classification of plant leaf diseases using machine learning and image preprocessing techniques. *Proceedings of the Confluence 2020 - 10th International Conference on Cloud Computing, Data Science and Engineering*, 480–484. <https://doi.org/10.1109/CONFLUENCE47617.2020.9057889>
- Soares, A. (2016). *INSTITUTO SUPERIOR DE CIÊNCIAS DA SAÚDE EGAS MONIZ MESTRADO INTEGRADO EM MEDICINA DENTÁRIA*.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 4278–4284. <https://doi.org/10.1609/aaai.v31i1.11231>
- Tiwari, A., Srivastava, S., & Pant, M. (2020). Brain tumor segmentation and classification from magnetic resonance images: Review of selected methods from 2014 to 2019. *Pattern Recognition Letters*, 131, 244–260. <https://doi.org/10.1016/J.PATREC.2019.11.020>
- Tomografia computadorizada (TC) em distúrbios neurológicos - Distúrbios neurológicos - Manuais MSD edição para profissionais*. (n.d.). Retrieved January 16, 2023, from <https://www.msmanuals.com/pt-pt/profissional/dist%C3%BArbios-neurol%C3%B3gicos/exames-e-procedimentos-neurol%C3%B3gicos/tomografia-computadorizada-tc-em-dist%C3%BArbios-neurol%C3%B3gicos>
- Types of Neural Networks and Definition of Neural Network*. (n.d.). Retrieved January 5, 2023, from <https://www.mygreatlearning.com/blog/types-of-neural-networks/>
- Ultrasound: MedlinePlus Medical Test*. (n.d.). Retrieved January 5, 2023, from <https://medlineplus.gov/lab-tests/sonogram/>
- Vasuki, P., Kanimozhi, J., & Devi, M. B. (2017). A survey on image preprocessing techniques for diverse fields of medical imagery. *Proceedings - 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering, ICEICE 2017, 2017-December*, 1–6. <https://doi.org/10.1109/ICEICE.2017.8192443>
- Vista do Deep Learning para Geração Automática de Legenda de Imagem*. (n.d.). Retrieved July 3, 2023, from <https://sol.sbc.org.br/index.php/eniac/article/view/9314/9216>
- Você sabe o que é o ultrassom obstétrico?* - *BedMed*. (n.d.). Retrieved January 16, 2023, from <https://bedmed.com.br/o-que-e-ultrassom-obstetrico/>
- What is PyCharm? Features, Advantages & Disadvantages*. (n.d.). Retrieved June 23, 2023, from <https://hackr.io/blog/what-is-pycharm>

- X-rays: Overview, side effects, risks, and more.* (n.d.). Retrieved January 5, 2023, from <https://www.medicalnewstoday.com/articles/219970#benefits>
- Yadav, S. S., & Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 6(1), 1–18. <https://doi.org/10.1186/S40537-019-0276-2/TABLES/16>
- Zambalde, A. L., Souza, T. A. de, & Souza, D. (n.d.). *Steps of the DSR method | Download Scientific Diagram.* Retrieved January 10, 2023, from https://www.researchgate.net/figure/Steps-of-the-DSR-method_fig2_344783533