

Relatório de Projeto

Modelo CNN: Dataset de Tomografias Axial Computorizadas de Pulmão

Unidade Curricular: Deep Learning

Docente: Professora Marta Matos

Realizado por:

Raquel Quintão n.º 2023134

Paulo Nunes n.º 2023129

Lisboa, Julho de 2024

Índice

1. Introdução	3
2. Análise Exploratória de Dados	4
2.1 Caracterização do Dataset.....	4
2.2. Importação e Verificação Inicial dos Dados	4
2.3 Divisão do dataset (treino e testes).....	5
2.4 Visualização das imagens por Categoria.....	5
2.5 Normalização do dataset	5
2.6 One-hot encoding.....	5
3. Metodologia	6
3.1 Modelo 0 (model_cnn0) – ponto de partida	6
3.2 Modelo 0_1 (model_cnn0_1).....	8
3.3 Modelo 0_2 (model_cnn0_2).....	10
3.4 Modelo Escolhido	11
3.4 Técnica LIME (Local Interpretable Model-agnostic Explanations)	12
4. Conclusão	13

1. Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular de Deep Learning, integrada no nosso programa de pós-graduação em Health Data Science.

A professora lançou-nos o desafio de aplicar os conhecimentos adquiridos a um problema real na área da saúde, nomeadamente a deteção precoce do cancro do pulmão através da análise de imagens de tomografia axial computadorizada (TAC).

O cancro do pulmão permanece uma das principais causas de morte por cancro a nível mundial, e a sua deteção atempada é crucial para melhorar as hipóteses de sobrevivência dos doentes.

Este desafio permite-nos explorar as capacidades adquiridas na UC de Deep Learning, procurando conectar a teoria ao contexto real da saúde.

O nosso objetivo principal é desenvolver um modelo de Deep Learning capaz de classificar imagens de TAC em três categorias: normal, benigno e maligno.

A importância desta tarefa estende-se além das fronteiras académicas. Na prática clínica, os radiologistas enfrentam diariamente o desafio de analisar um grande volume de imagens.

Um sistema de apoio à decisão baseado em inteligência artificial pode:

- Aumentar a precisão do diagnóstico;
- Reduzir o tempo de análise;
- Priorizar casos potencialmente malignos e avaliação mais rápida.

Com os objetivos desta UC em mente, decidimos abordar este projeto como um pedido de um hospital privado para detetar massas potencialmente malignas. O objetivo deste cliente é poder encaminhar os utentes para uma linha de diagnóstico e tratamento o mais atempado possível.

No dia a dia atual o utente só é encaminhado para biopsia após um processo burocrático demorado entre alguns exames e consultas com o seu médico. Com a possibilidade de avaliar automaticamente os casos com massas potencialmente malignas através de uma imagem de TAC, o cliente pretende encaminhar para a realização de diagnóstico definitivo por biopsia com elevada prioridade nestes casos mais sensíveis.

Neste sentido definimos que o nosso principal objetivo para o modelo é uma sensibilidade elevada para os casos de massas malignas, uma vez que um Falso Negativo poderá levar a um diagnóstico médico preciso tardio, assim como o seu tratamento.

2. Análise Exploratória de Dados

A análise Exploratória de Dados é uma etapa fundamental para compreender os dados, captar insights iniciais que podem influenciar a construção dos modelos e preparar o dataset para a fase de modelação.

2.1 Caracterização do Dataset

O dataset contém 472 imagens de tomografia axial computadorizada (TAC) de 512x512 pixels, distribuídas em "normal" (153 imagens), "benigno" (120 imagens) e "maligno" (199 imagens). Há um desbalanceamento desprezível do dataset.

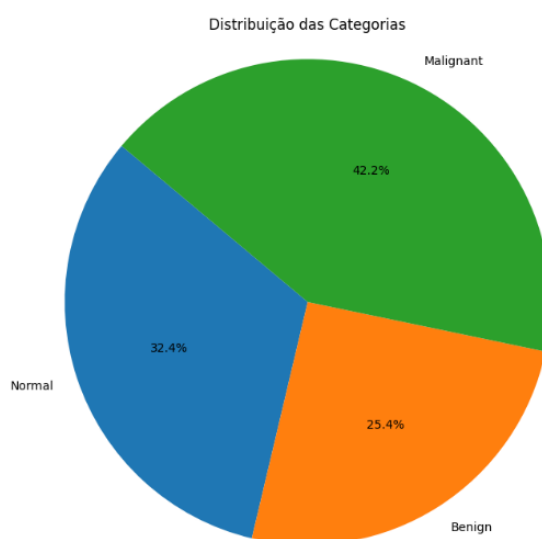


Figura 1 - gráfico circular com a distribuição em % das categorias do dataset

2.2. Importação e Verificação Inicial dos Dados

A função `load_images` foi usada para carregar e processar imagens em três categorias: Normal, Benign e Malignant. Estas categorias são mapeadas para números inteiros através de um dicionário chamado `label_dict` (0, 1, 2, respetivamente), permitindo a atribuição dos rótulos corretos às imagens enquanto são carregadas.

A biblioteca PIL é usada para abrir e converter as imagens para o formato RGB, assegurando a consistência na representação das cores. Cada imagem é então redimensionada para 256x256 pixels para facilitar a entrada no modelo de rede neural, reduzindo a carga computacional e os requisitos de memória. Este redimensionamento também contribui para tempos de treino mais rápidos.

Após o redimensionamento, cada imagem é convertida num 'array numpy' e a sua forma é verificada para garantir que esteja no formato correto (256, 256, 3). Se a forma estiver correta, a imagem e o rótulo são adicionados à lista final.

2.3 Divisão do dataset (treino e testes)

Nesta fase, preparamos o modelo para treino e teste, utilizando a função `train_test_split` do scikit-learn com o parâmetro `'stratify=y'` para garantir que a proporção das classes seja mantida nos conjuntos de treino e teste, assegurando um treino e avaliação justos e balanceados do modelo. Dividimos os dados em 70% para treino (330 imagens) e 30% para teste (142 imagens). Para confirmar a divisão do dataset, imprimimos os shapes dos arrays resultantes: `X_train.shape`, `X_test.shape` e `y.shape`.

2.4 Visualização das imagens por Categoria

A função `show_sample_images` apresenta amostras de imagens categorizadas utilizando Matplotlib. Esta organiza os subplots para visualizar múltiplas categorias de imagens de forma estruturada, usando `plt.subplot` para criar subplots, `plt.imshow` para exibir as imagens, e `plt.xlabel` para rotular as categorias. A função itera sobre cada categoria, seleciona aleatoriamente um número definido de amostras com `np.random.choice` e ajusta o tamanho da figura com `plt.figure`. O objetivo é proporcionar uma visualização rápida e intuitiva das imagens de cada categoria, auxiliando na análise exploratória e verificação dos dados no conjunto de teste (`X_test` e `y_test`) e de treino (`X_train` e `y_train`).

2.5 Normalização do dataset

A normalização dos dados, como em `x_train=X_train/255` e `x_test=X_test/255`, é crucial no pré-processamento de imagens para deep learning. Ajustando os valores de pixel para uma escala de 0 a 1, aumenta-se a estabilidade numérica e evita-se problemas de grandeza escalar, o que acelera o treino e resulta em atualizações de pesos mais consistentes. Modelos com dados normalizados geralmente têm melhor desempenho, garantindo uniformidade e comparações mais justas e consistentes entre diferentes entradas.

2.6 One-hot encoding

No projeto, a conversão das labels para inteiros e a aplicação de one-hot encoding são essenciais para preparar dados de classificação categórica. Primeiro, as labels são convertidas para inteiros, em seguida, aplicamos one-hot encoding para transformar cada rótulo inteiro num vetor binário, onde cada classe é representada de forma única. Este processo é necessário para redes neuronais que utilizam a Loss function 'crossentropy', ajudando a otimizar a precisão do modelo e a distinguir entre classes de maneira eficiente.

3. Metodologia

Após a preparação do dataset passámos para a fase de definir, compilar, treinar e avaliar os modelos. Começamos com um modelo inicial base aplicando as boas práticas estudadas em aula, e do qual foram elaboradas duas versões alternativas para validar se traziam benefícios para o objetivo definido (elevada sensibilidade à categoria Maligna).

Escolhemos um modelo de Rede Neural Convolucional (CNN) para classificar imagens médicas devido à sua eficácia em processar estes dados e capturar características essenciais para diagnósticos precisos.

3.1 Modelo 0 (model_cnn0) – ponto de partida

Camada (Tipo)	Output Shape	Param #
Conv2D (Convolutional)	(None, 254, 254, 16)	448
MaxPooling2D (Pooling)	(None, 127, 127, 16)	0
Conv2D (Convolutional)	(None, 125, 125, 32)	4640
MaxPooling2D (Pooling)	(None, 62, 62, 32)	0
Flatten (Flatten)	(None, 123008)	0
Dense (Fully Connected)	(None, 1024)	125961216
Dense (Output)	(None, 3)	3075

Tabela 1- Arquitetura do modelo_cnn0

O modelo base utilizado é uma Rede Neural Convolucional (CNN) projetada para processar imagens de entrada com dimensões de 256x256 pixels e 3 canais de cor (RGB). Esta arquitetura permite ao modelo extrair características hierárquicas das imagens de entrada, reduzir a dimensionalidade dos dados e, finalmente, realizar a classificação. A arquitetura segue um padrão típico de CNNs, onde as primeiras camadas extraem características locais e as camadas finais combinam essas características para a classificação. A redução progressiva do tamanho espacial (de 256x256 para 62x62) permite que a rede capture informações em diferentes escalas, enquanto o aumento no número de filtros (de 16 para 32) permite a deteção de padrões mais complexos.

A arquitetura começa com a camada de entrada (input), que recebe os dados de entrada para a rede CNN. Em seguida, são criadas 2 camadas de convolução: a primeira é uma camada Conv2D com 16 filtros de tamanho (3, 3) e ativação 'ReLU', seguida por uma camada MaxPooling2D (2x2) para reduzir a dimensionalidade, e a segunda camada Conv2D com 32 filtros de tamanho (3, 3) também seguida por uma camada MaxPooling2D (2x2). Depois das camadas de convolução, há 2 camadas ocultas: a camada Flatten, que transforma os dados num vetor unidimensional para a camada Dense de 1024 neurônios, com função de ativação 'ReLU'. O modelo termina com a camada de saída (output), uma camada 'Dense' com 3 neurônios e função de ativação softmax para classificação multitarget. O modelo é compilado com o otimizador Adam e a função de perda (loss function) 'crossentropy', utilizando a métrica de 'accuracy'.

Este modelo foi treinado em 10 épocas, cada uma demorando entre 14 e 16 segundos. Para avaliação de desempenho do modelo, analisamos os gráficos de Loss Score, Accuracy, a Matriz de Confusão e o relatório de classificação

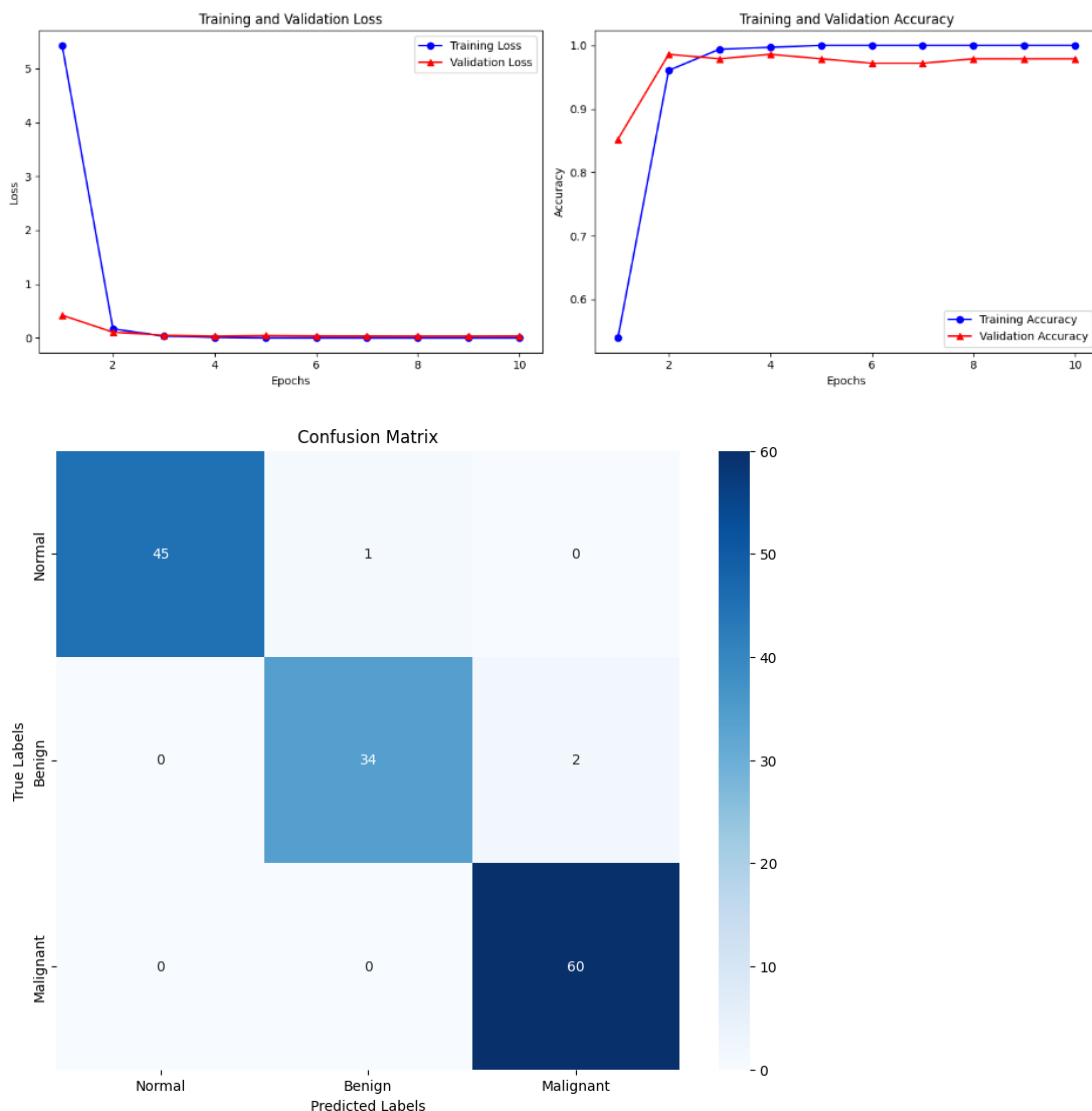


Ilustração 0.1- output loss, accuracy e matriz de confusão modelo_cnn0

modelo_cnn0	Precision	Recall	F1-Score	Support
Classe 0	1.00	0.98	0.99	46
Classe 1	0.97	0.94	0.96	36
Classe 2	0.97	1.00	0.98	60
Accuracy	0.98			142
Val_loss	0.0337			

Tabela 2 – Métricas de Classificação modelo_cnn0

Estas métricas mostram que o modelo tem um desempenho excelente, com alta precisão, recall (sensibilidade) e F1-Score em todas as classes, e uma baixa perda de validação, indicando que o modelo generaliza bem e não ocorre de overfitting. Verificam-se curvas de treino estáveis e convergentes.

Na matriz de confusão podemos verificar que todos os casos malignos foram detetados corretamente, pelo que todos os utentes com real necessidade de biopsia seriam priorizados, o que sustenta o recall de 1.00 para categoria 2 (Maligno).

Os restantes erros detetados como 2 casos de massas benignas indevidamente classificados de Malignas, levariam estes casos a biopsia de forma prioritária, mas não teria impacto significativo, uma vez que estes doentes seriam apenas diagnosticados após confirmação de biopsia.

Este modelo preenche já os requisitos pedidos pelo cliente, porém tentaremos otimizar alterando a sua arquitetura, já que não temos gráficos que apresentem características de overfitting, não sendo necessário aplicar técnicas de regularização.

3.2 Modelo 0_1 (model_cnn0_1)

Camada (Tipo)	Output Shape	Param #
Conv2D (Convolutional)	(None, 252, 252, 32)	2432
MaxPooling2D (Pooling)	(None, 126, 126, 32)	0
Conv2D (Convolutional)	(None, 122, 122, 64)	51264
MaxPooling2D (Pooling)	(None, 61, 61, 64)	0
Conv2D (Convolutional)	(None, 57, 57, 128)	204928
MaxPooling2D (Pooling)	(None, 28, 28, 128)	0
Flatten (Flatten)	(None, 100352)	0
Dense (Fully Connected)	(None, 504)	50577912
Dense (Output)	(None, 3)	1515

Tabela 3 Arquitetura do modelo_cnn0

Neste modelo tentamos fazer alterações ao Modelo 0 para validar o seu impacto:

- **Adição de uma camada de convolução** – com o aumento do número de camadas tentámos que o modelo fosse capaz de identificar características mais complexas.
- **Alteração da kernel para 5x5 (ao invés de 3x3)** – aumentando a capacidade do modelo para identificar características de maiores dimensões.
- **Diminuição do número de neurónios da camada oculta dense** – com o objetivo de ajustar a carga computacional extra gerada pelas opções anteriores.

Nesse sentido treinamos o modelo e procedemos à sua avaliação. Este modelo foi treinado no mesmo número de épocas que o Modelo 0, as quais demoraram entre 15 e 17 segundos.

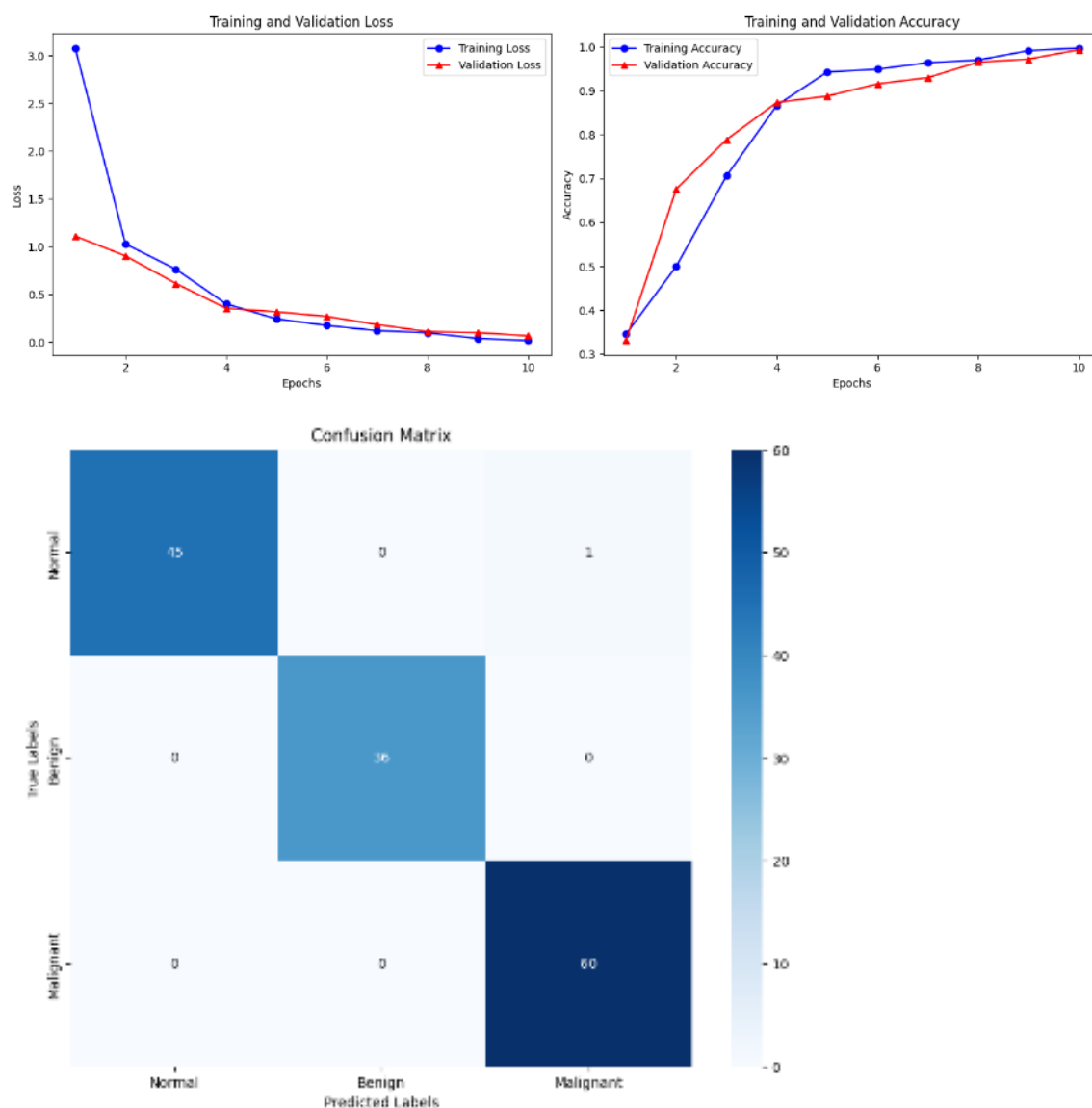


Ilustração 0.2 output loss, accuracy e matriz de confusão `modelo_cnn0_1`

modelo_cnn0_1	Precision	Recall	F1-Score	Support
Classe 0	1.00	0.98	0.99	46
Classe 1	1.00	1.00	1.00	36
Classe 2	0.98	1.00	0.99	60
Accuracy			0.99	142
Val_loss			0.0639	

Tabela 4 Métricas de Classificação `modelo_cnn0_1`

O Modelo 0_1 apresenta resultados ligeiramente melhores que o Modelo 0, com métricas superiores ou iguais, exceto de `val_loss`. As suas curvas não são tão estáveis como o modelo base. Por outro lado, neste modelo um caso de um doente saudável é identificado como um caso Maligno pelo que esta situação pode encaminhar um utente a um exame desnecessário, contudo é um caso em 46.

Este modelo cumpre os objetivos pretendidos, porém apresenta uma maior complexidade, consumindo assim mais recursos computacionais, sem trazer valor significativo para o cliente.

3.3 Modelo 0_2 (model_cnn0_2)

Camada (Tipo)	Output Shape	Param #
Conv2D (Convolutional)	(None, 254, 254, 16)	448
MaxPooling2D (Pooling)	(None, 127, 127, 16)	0
Flatten (Flatten)	(None, 258064)	0
Dense (Fully Connected)	(None, 1024)	264258560
Dense (Output)	(None, 3)	3075

Tabela 5 – Arquitetura do modelo_cnn0_2

Neste modelo, inversamente, fizemos alterações ao Modelo 0 no sentido de simplificar a sua arquitetura, como tal removemos uma camada convolucional e respetiva camada de pooling. Pretendíamos avaliar as diferenças de desempenho e performance, já que usaria menos recursos computacionais.

Este modelo foi igualmente treinado em 10 épocas, e o seu treino teve uma performance mais baixa, demorando entre 28 e 44 segundos por época. O que poderá indicar uma fraca utilização dos recursos computacionais disponíveis em relação ao modelo 0.

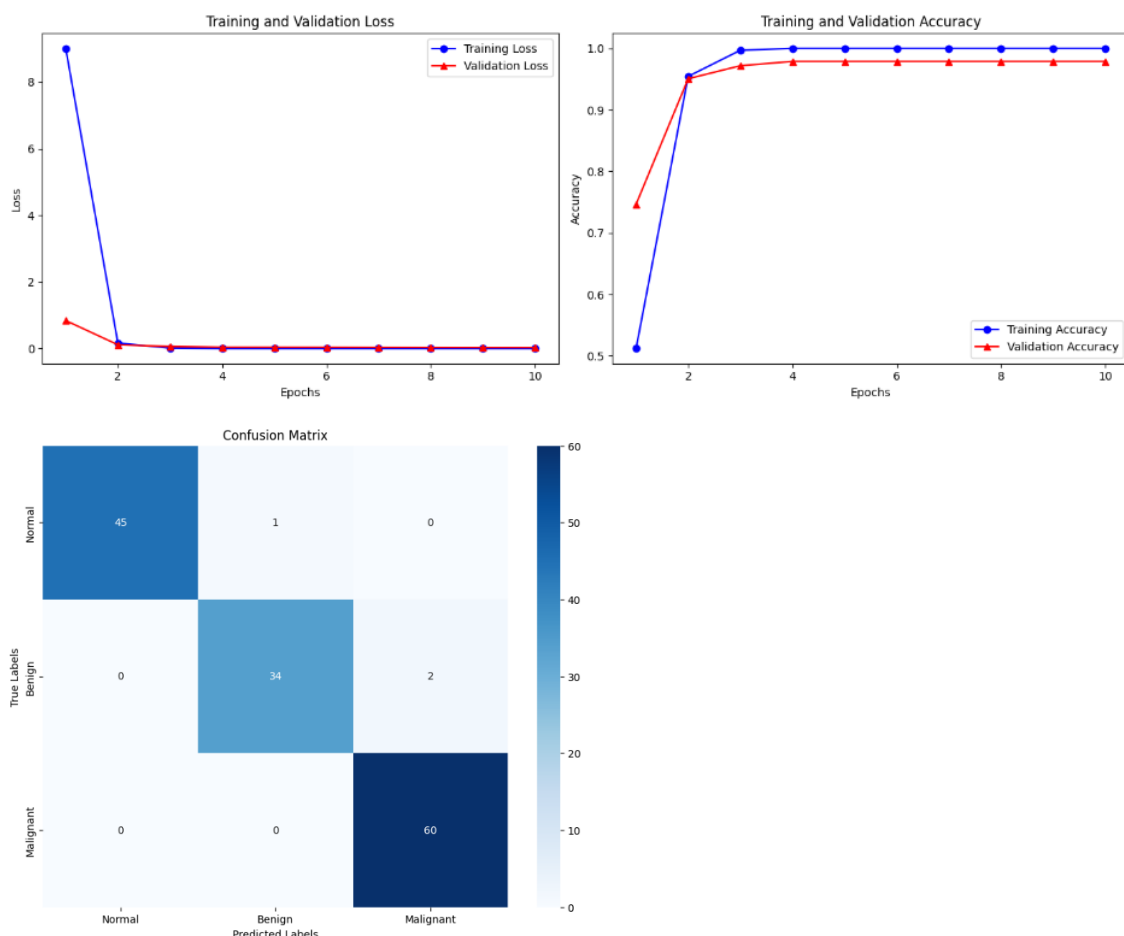


Ilustração 0.3 output loss, accuracy e matriz de confusão modelo_cnn0_2

model_cnn0_2	Precision	Recall	F1-Score	Support
Classe 0	1.00	0.98	0.99	46
Classe 1	0.97	0.94	0.96	36
Classe 2	0.98	1.00	0.98	60
Accuracy			0.98	142
Val_loss			0.0296	

Tabela 6-Métricas de Classificação modelo_cnn0_2

Este modelo apresenta um desempenho muito idêntico ao Modelo 0, diferenciando apenas tem curvas de accuracy ligeiramente mais estáveis e assim como Validation Loss Score marginalmente menor do que o modelo base.

3.4 Modelo Escolhido

Qualquer um dos modelos apresentados reflete desempenhos que permitem o projeto ser implementado. Em todos é possível apurar com bastante precisão os casos potencialmente malignos, não havendo evidência de que um caso Maligno fosse identificado como Normal ou Benigno.

Devido à performance equivalente e aos resultados de desempenho idênticos entre o Modelo 0 e o Modelo 0_2, optamos por excluir o Modelo 0_2. Assim, a decisão recai agora entre o Modelo 0 e o Modelo 0_1.

Ao ponderarmos entre o Modelo 0 e o Modelo 0_1, tivemos em consideração vários fatores.

O Modelo 0 é mais simples, com menos camadas e parâmetros, o que facilita a sua interpretação. Além disso, é mais eficiente em termos de tempo de treino e uso de recursos computacionais. Esta eficiência é uma consideração importante para nós, especialmente quando lidamos com recursos computacionais limitados, como é frequentemente o caso em contextos hospitalares.

Por outro lado, o Modelo 0_1 tem um desempenho ligeiramente superior na classificação de casos benignos e na redução de falsos positivos para casos malignos. Este modelo, com uma camada de convolução adicional e mais filtros em cada camada, pode ser capaz de capturar características mais complexas nos dados. Isto pode ser útil quando lidamos com imagens que apresentam muita variação ou detalhes subtis que o modelo precisa aprender.

No entanto, o nosso objetivo principal é maximizar a capacidade do modelo de identificar corretamente os casos malignos. Neste sentido, optámos pelo Modelo 0. Apesar da sua simplicidade, este modelo já demonstrou uma sensibilidade perfeita de 1, o que significa que é capaz de identificar todos os casos malignos corretamente. Além disso, a sua maior eficiência computacional e simplicidade tornam-no uma opção atraente para

alcançar o nosso objetivo. Contudo se o objetivo fosse identificar com maior precisão todas as classes o modelo que escolheríamos seria o Modelo 0_1.

Em suma, após considerarmos todas estas variáveis, optámos pelo Modelo 0. Todos os modelos têm as suas vantagens, mas acreditamos que o Modelo 0 é a melhor escolha para o nosso contexto específico.

3.4 Técnica LIME (Local Interpretable Model-agnostic Explanations)

A técnica LIME desvenda a “blackbox” dos modelos de deep learning, tornando as suas decisões mais interpretáveis. Ela fornece insights sobre como um modelo toma decisões, através da geração de explicações locais para previsões específicas. Isto é feito perturbando a entrada original e observando o impacto no desempenho do modelo. Assim, a LIME contribui para uma maior transparência na tomada de decisão dos modelos.

Nesse sentido tentamos perceber quais os elementos que mais contribuíram positiva e negativamente as classificações do modelo escolhido (Modelo 0), usando esta técnica.

Apesar de termos como principal objetivo identificar os casos Malignos, decidimos, numa perspetiva académica, avaliar para todas as classes quais os elementos que influenciaram a decisão do modelo.

Optamos por uma abordagem usando os super-píxeis dado que o output era

Optamos por uma abordagem usando os super-píxeis e identificando quer contribuições positivas quer negativas dado que explorando outra visualização não estávamos a obter resultados relevantes. Desta forma a visualização das áreas a vermelho são as contribuições negativas e a azul as contribuições positivas.

A imagem abaixo é um exemplo da execução da técnica num caso benigno. Optamos por este exemplo pois era o que identificava os dois tipos de contribuições. O não domínio de imagens TAC do pulmão não nos permite fazer uma avaliação crítica das características que o modelo considerou. Apesar de tudo conseguimos perceber que no pulmão esquerdo existem evidências que contribuíram positivamente para classificar como um caso benigno, enquanto as características do pulmão direito provavelmente contribuem negativamente pela não existência deste tipo de massas.

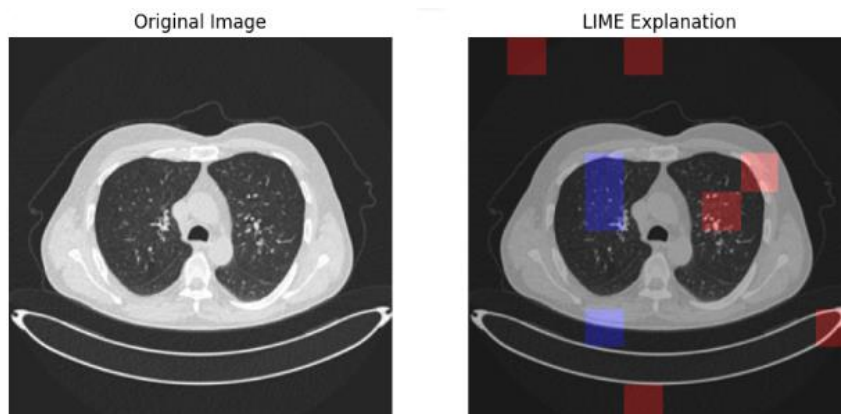


Ilustração 0.4 - comparação imagem modelo LIME

Para as restantes categorias não chegamos a conclusões que permitissem a compreensão da decisão do modelo. Contudo certificamos que o modelo estava a fazer uma correta classificação, validando que o `top_label` correspondia à categoria da imagem analisada.

4. Conclusão

Este projeto, desenvolvido no âmbito da unidade curricular de Deep Learning do nosso programa de pós-graduação em Health Data Science, focou-se em aplicar conhecimentos teóricos a um problema prático na área da saúde: a deteção precoce do cancro do pulmão através da análise de imagens de tomografia axial computadorizada (TAC).

Os modelos desenvolvidos neste projeto têm o potencial de aumentar a precisão do diagnóstico, reduzir o tempo de análise e priorizar casos potencialmente malignos para avaliação mais rápida. Na prática clínica, isso pode significar um encaminhamento mais eficiente dos utentes para biópsias, melhorando as hipóteses de sobrevivência dos doentes com cancro do pulmão.

Este projeto ilustra a aplicabilidade prática de técnicas de Deep Learning nos problemas de saúde objetivos, salientando o potencial tecnológico da inteligência artificial (IA) na transformação e otimização do diagnóstico médico.

O desenvolvimento deste projeto foi uma experiência profundamente enriquecedora, permitindo-nos aplicar os conhecimentos adquiridos em sala de aula e alimentar a nossa ambição por um futuro nesta área. Apesar de reconhecermos que o overfitting não era um problema no nosso contexto, decidimos, ainda assim, explorar técnicas para o combater. Essa exploração está documentada na secção 7, Anexos, do nosso notebook a qual tem uma breve explicação do realizado. Acreditamos que esta abordagem proativa não só fortaleceu o nosso entendimento atual, mas também nos equipou com ferramentas valiosas para futuros projetos.