

# Classificação de Melanomas

Edgar Alves <sup>[33017]</sup> <sup>1</sup> and Bernardo Poiares <sup>[39100]</sup> <sup>2</sup>

<sup>1</sup>MEIC, Instituto Superior de Engenharia de Lisboa, a33017@alunos.isel.pt and <sup>2</sup>MEIC, Instituto Superior de Engenharia de Lisboa, a39100@alunos.isel.pt

## Abstract

Classificar um conjunto de imagens de melanomas utilizando aprendizagem profunda.

**Key words:** PyTorch, TensorFlow, Rede Neural, Aprendizagem Profunda, Melanomas

## Introdução

O avanço da tecnologia de aprendizagem profunda tem revolucionado várias áreas da ciência e da tecnologia, incluindo a medicina e a saúde. Um dos campos em que a aprendizagem profunda tem mostrado grande potencial é no diagnóstico e classificação de doenças através da análise de imagens médicas. No contexto específico do cancro de pele, como o melanoma, a deteção precoce desempenha um papel crucial na eficácia do tratamento e na sobrevivência dos pacientes.

Neste contexto, a utilização de algoritmos de aprendizagem profunda apresenta-se como uma abordagem promissora para melhorar a precisão e eficiência do diagnóstico de melanoma cutâneo. Ao treinar modelos de rede neural com grandes conjuntos de dados de imagens dermatoscópicas, é possível desenvolver sistemas automatizados capazes de identificar padrões subtis associados ao melanoma, auxiliando os médicos no processo de decisão clínica.

Este relatório apresentará uma visão geral dos métodos e ferramentas a utilizar.

## PyTorch vs TensorFlow

Uma das grandes decisões que foi necessário tomar foi em relação a que ferramenta utilizar. Das várias existentes no mercado, existem duas que se destacam: *Pytorch* e *TensorFlow*

### PyTorch

*PyTorch* é uma estrutura de aprendizagem profunda de código aberto baseada na linguagem de programação *Python* e na biblioteca *Torch*. Foi desenvolvida pela *Meta AI* e tem ganho atenção devido à sua flexibilidade e fácil aprendizagem. (1)

### TensorFlow

*TensorFlow* é uma estrutura de *machine learning* e de Redes Neurais primariamente desenvolvida pela *Google* baseada em *Python*, mas também em *C++* e *JavaScript*. Geralmente é conhecida pelo uma curva de aprendizagem grande, apesar de também disponibilizar ferramentas tão ou mais poderosas que o *PyTorch*. Esta ferramenta está mais orientada para grandes volumes de informação. (2)

## Decisão

Após averiguação dos pontos a favor e contra de ambas as ferramentas, foi decidido avançar com *PyTorch*. A versatilidade da linguagem *Python* foi um dos factores-chave, bem como a sua fácil aprendizagem e adaptabilidade.

## Dataset de melanomas a utilizar

Apesar de existirem vários conjuntos de imagens *online* que seja possível utilizar, foi optado pelo *dataset* disponibilizado pela *Kaggle*<sup>1</sup>. Esta decisão é devido às imagens serem claras, com margem de erro negligenciável e de conter imagens para aprendizagem como de teste (amostra de 10.000 imagens).

## Rede Neural

TODO

## Estratégia e fluxo de funcionamento

Como estratégia, temos fluxo de funcionamento conforme demonstrado na Figura 1. Existe a necessidade de realizar um primeiro tratamento das imagens para que sejam todas uniformes nas suas dimensões, tanto para as imagens de aprendizagem como para as imagens de irão servir de teste. Com essas imagens de aprendizagem, é gerado um conjunto de imagens tratadas, que por sua vez, serão usadas para criar o modelo de teste, que é o que irá ser utilizado para realizar a previsão, juntamente com o conjunto de imagens de teste.

## Análises

De futuro, poderá ser realizado testes com outros *datasets* com menor qualidade de forma a verificar as alterações na previsão do modelo;

Variação da previsão consoante o número de camadas usada

<sup>1</sup> <https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images?resource=download>

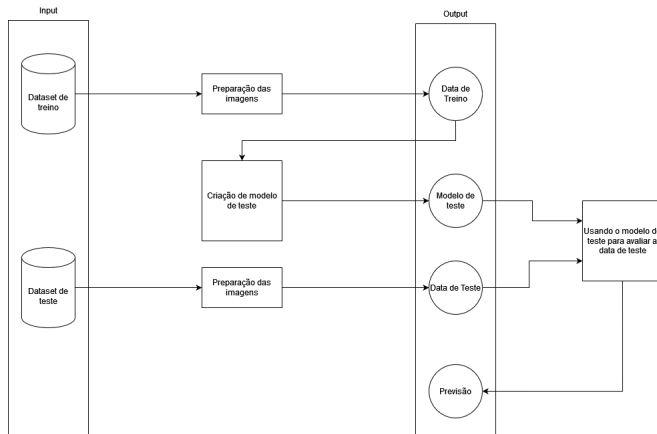


Fig. 1. Diagrama de fluxo de funcionamento

e tempo demorado a fazer a previsão sobre um conjunto de imagens;

Como o número de *epochs* (ou passagens pelo *dataset*) influencia as previsões;

## References

1. PyTorch, documentação oficial  
<https://pytorch.org/docs/stable/index.html>,  
acedido pela última vez em 23 Março 2024 .
2. TensorFlow, documentação oficial  
<https://www.tensorflow.org/learn?hl=pt>,  
acedido pela última vez em 23 Março 2024 .
3. Neural Networks from Scratch in Python  
<https://nnfs.io/>,  
acedido pela última vez em 23 Março 2024 .