

Relatório do Projeto

Designação do projeto: Learning Leafs

Participantes:

Nome	Número
André Letras	50032533

Docentes envolvidos:

Nome	Email
Jacinto Estima	jacinto.estima@universidadeeuropeia.pt
José Braga de Vasconcelos	jose.vasconcelos@universidadeeuropeia.pt

Índice

1. ENQUADRAMENTO	3
2. GANTT CHART – PLANO DE TRABALHOS	4
3. PERSONAS	4
4. CENÁRIOS	4
4.1 PRINCIPAL	4
4.2 SECUNDÁRIOS	5
5. REQUISITOS	5
5.1 REQUISITOS FUNCIONAIS	5
5.2 REQUISITOS NÃO FUNCIONAIS	6
6. MOCKUPS	7
7. MODELO DE DOMÍNIO	9
8. IMPLEMENTAÇÃO DO NOTEBOOK	9
8.1 BIBLIOTECAS E PACOTES EXTERNOS UTILIZADOS	9
REFERÊNCIAS	11

1. Enquadramento

O termo Machine Learning tem vindo cada vez mais a ganhar algum impacto na comunidade tecnológica, na medida em que existem cada vez mais inovações na área a um ritmo incrível. A cada dia que passa são feitos cada vez mais estudos e desenvolvidos novos algoritmos para que as máquinas consigam aprender melhor e mais rapidamente em diversas áreas de aplicação.

Esta área de estudo insere-se numa maior área tecnológica que é a Inteligência Artificial, e teve origem entre os anos 50 e 60. Em 1950 dá-se o primeiro acontecimento importante, realizado por Alan Turing, chamado de “Teste de Turing”, onde se verificava se um computador possuía inteligência ou não. Este acontecimento veio revolucionar todo o conhecimento tecnológico até à altura e dar origem a muitas inovações nesta década. Em 1952, Arthur Samuel escreve o primeiro programa de aprendizagem num computador. O programa era um jogo de damas e o computador da IBM conseguia estudar que jogadas conseguiam levar à vitória mais facilmente, e eram depois introduzidas no jogo. Em 1957 é desenhada a primeira rede neuronal para um computador por Frank Rosenblatt, cujo funcionamento simulava o cérebro humano.

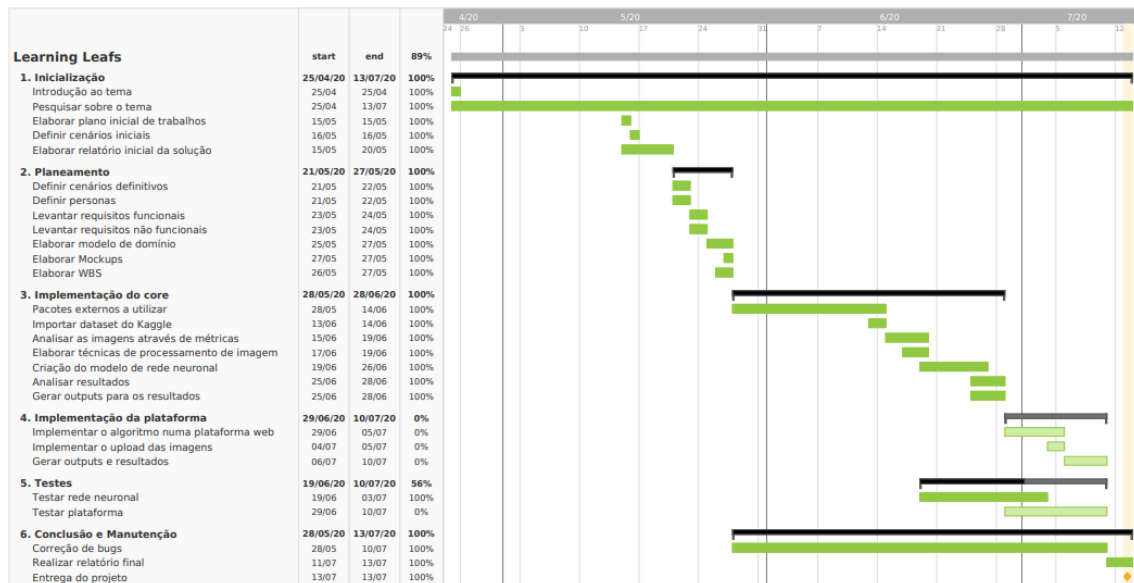
Para a realização deste projeto irão ser utilizados temas desta área, como Machine Learning, Redes Neurais, ou, mais propriamente Redes Neurais Convolucionais.

Os objetivos deste projeto estão bem definidos, e foram sugeridos num desafio da plataforma Kaggle [1] em parceria com a Universidade de Cornell, Nova Iorque. O objetivo principal será conseguir treinar um modelo de rede neuronal de forma a conseguir: Identificar e classificar as possíveis doenças observadas numa imagem de uma folha de uma árvore, ou, caso não exista nenhuma característica, identificá-la como saudável; Distinguir entre as variadas doenças, por vezes até mais do que uma doença na mesma imagem; Conseguir lidar com novos sintomas ou sintomas raros; Conseguir identificar a imagem a partir de qualquer ambiente da imagem: ângulo, luz, sombra, idade da folha;

Todos os dados utilizados serão os disponibilizados pela plataforma, que por sua vez são de origem do estudo [2] publicado pela universidade de Cornell.

Como complemento do projeto, e como objetivo secundário, haverá uma possibilidade de realização de uma plataforma com o objetivo de ser carregada uma imagem de uma folha com o objetivo de ser identificada alguma doença ou não pela rede neuronal realizada para o objetivo core do projeto.

2. Gantt Chart – Plano de trabalhos



3. Personas

Este projeto está planeado para ser usado por qualquer utilizador que necessite de obter informação acerca do estado atual de uma folha de árvore ou outra planta, na medida em que consegue obter a informação acerca das doenças que tem, ou se realmente não apresenta nenhuma doença.

4. Cenários

4.1 Principal

Elaboração do algoritmo base da rede neural – O utilizador será capaz de visualizar todos os passos e entender o funcionamento do algoritmo para a decisão da rede neuronal na folha, através de um notebook do python onde cada célula terá um output para visualização dos dados.

4.2 Secundários

Plataforma de acesso facilitado ao utilizador – O utilizador conseguirá entrar na plataforma e identificar facilmente o processo de upload de várias imagens para ser decidido pelo algoritmo quais as possíveis doenças das folhas carregadas.

Visualização da informação – Ao carregar as imagens, o utilizador conseguirá visualizar a decisão final das probabilidades das doenças da folha que fez upload na plataforma.

5. Requisitos

5.1 Requisitos Funcionais

Cenário Principal			
ID	Nome SMART	Descrição	Prioridade
FR01	Ler e importar ficheiros de imagem	O algoritmo será capaz de importar as imagens provenientes do dataset da plataforma.	ALTA
FR02	Bibliotecas e Pacotes	O utilizador será capaz de entender quais os pacotes e bibliotecas externas utilizadas para a construção do algoritmo.	ALTA
FR03	Algoritmo de Redes Neurais	O notebook será capaz de executar um algoritmo de rede neuronal para aprender e calcular as probabilidades das doenças nas folhas	ALTA
FR04	Visualização de exemplos de resultados	O utilizador será capaz de visualizar os resultados provenientes da rede neuronal e do algoritmo para a decisão das doenças em folhas exemplo.	ALTA
Cenários Secundários			
ID	Nome SMART	Descrição	Prioridade
FR07	Apresentação simples	A plataforma irá apresentar de forma simples a forma de funcionamento	ALTA

FR08	Importar algoritmo	A plataforma irá importar o algoritmo proveniente do notebook para identificar as doenças na imagem carregada.	ALTA
FR09	Upload da imagem	A plataforma irá permitir o upload de uma imagem para ao algoritmo prever as doenças que essa folha terá.	ALTA
FR10	Apresentação dos dados	A plataforma irá apresentar os dados relativamente ao algoritmo de redes neuronais para a previsão das doenças na folha.	ALTA

5.2 Requisitos não funcionais

#	Nome SMART	Descrição	Prioridade
NFR01	Processamento de imagem	O utilizador será capaz de entender quais os algoritmos de processamento de imagem que poderiam ser, ou foram utilizados no algoritmo.	MÉDIA
NFR02	Escrita em ficheiro externo	O utilizador conseguirá observar os dados das probabilidades para cada folha num ficheiro externo, gerado no fim do notebook.	MÉDIA

6. Mockups

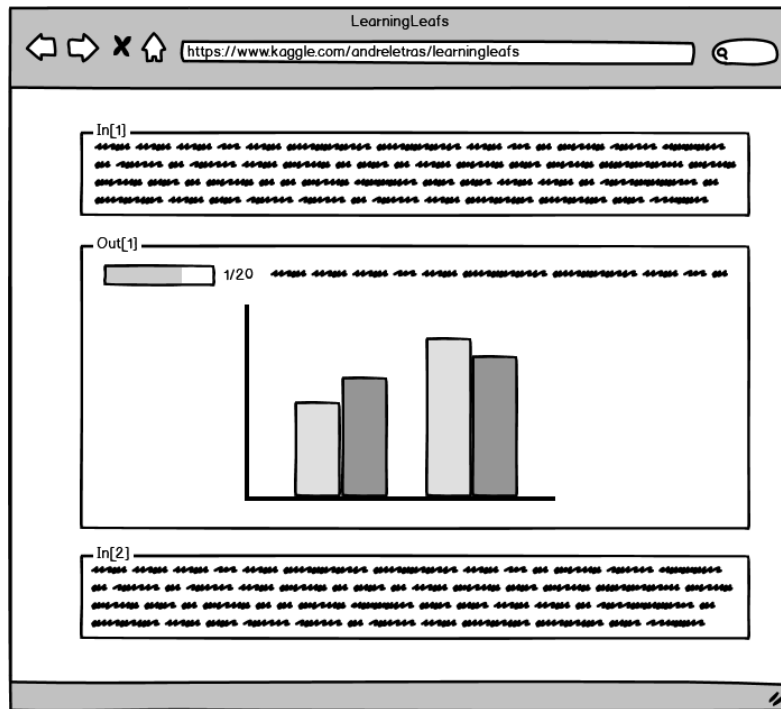


Figura 1: Aspeto do notebook de execução do algoritmo.

Dataset Image	Healthy	Rust	Scab	Multiple_diseases
Test_0	0.000112	0.003262	0.996598	0.000029
Test_1	0.003146	0.024062	0.972512	0.000279
Test_2	0.000039	0.000054	0.000008	0.999899
Test_3	0.999944	0.000010	0.000039	0.000007
Test_4	0.000008	0.001906	0.998037	0.000049
Test_5	0.000112	0.003262	0.996598	0.000029
Test_6	0.003146	0.024062	0.972512	0.000279
Test_7	0.000039	0.000054	0.000008	0.999899
Test_8	0.999944	0.000010	0.000039	0.000007
Test_9	0.000008	0.001906	0.998037	0.000049
Test_10	0.000112	0.003262	0.996598	0.000029
Test_11	0.003146	0.024062	0.972512	0.000279
Test_12	0.000039	0.000054	0.000008	0.999899
Test_13	0.999944	0.000010	0.000039	0.000007
Test_14	0.000008	0.001906	0.998037	0.000049
Test_15	0.000112	0.003262	0.996598	0.000029
Test_16	0.003146	0.024062	0.972512	0.000279
Test_17	0.000039	0.000054	0.000008	0.999899
Test_18	0.999944	0.000010	0.000039	0.000007

Figura 2: Apresentação dos dados completos no final da execução do algoritmo.

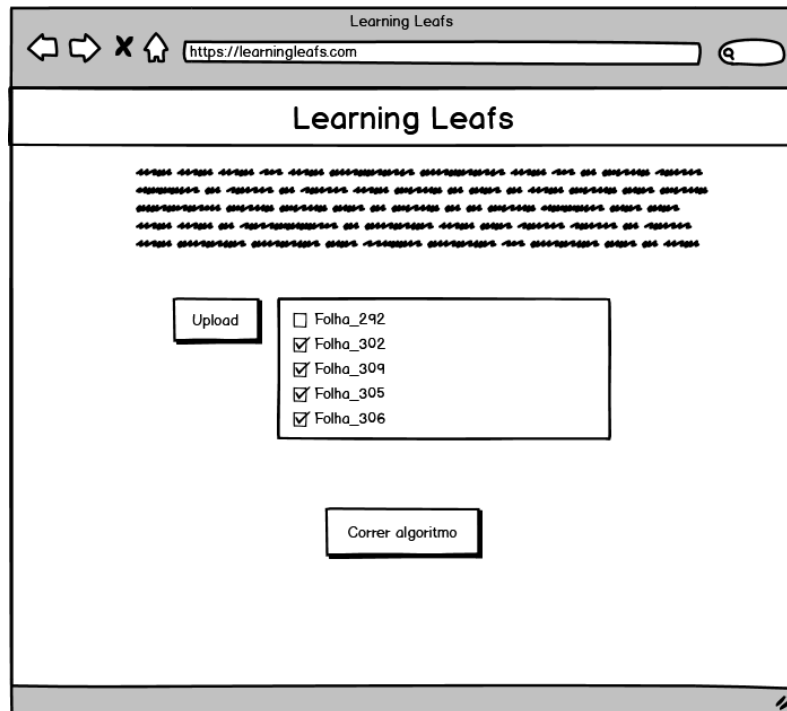


Figura 3: Plataforma para um utilizador carregar as imagens.

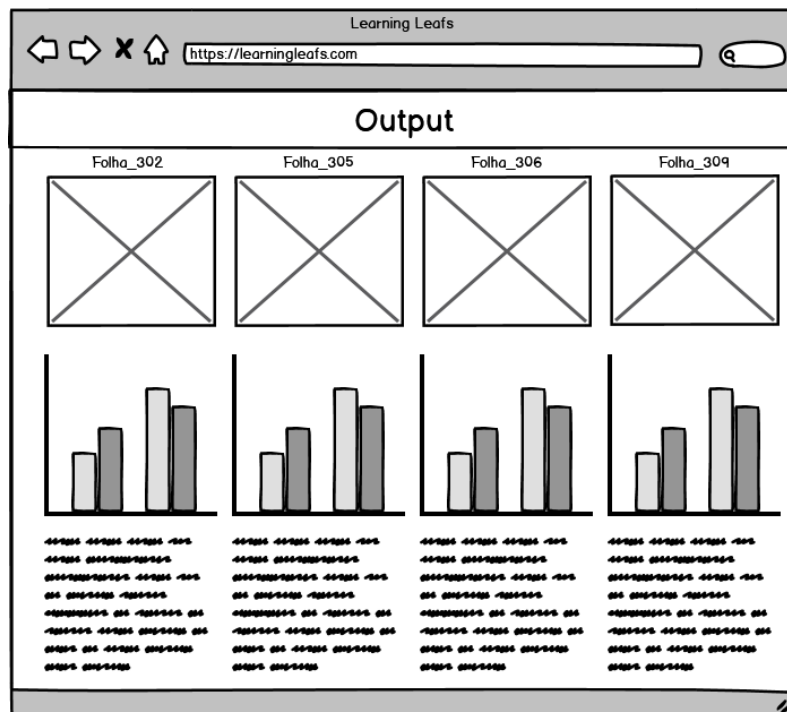
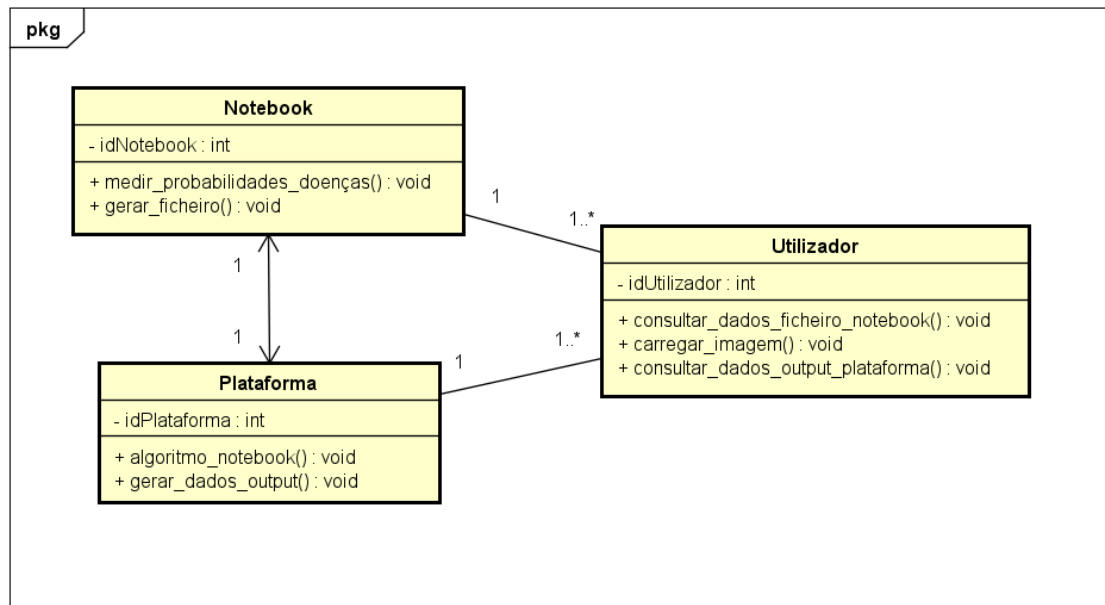


Figura 4: Output das imagens carregadas pelo utilizador.

7. Modelo de Domínio



8. Implementação do notebook

A implementação do notebook foi realizada recorrendo à linguagem Python, e também à própria plataforma Kaggle. Esta plataforma disponibiliza Kernels para implementação dos notebook com bibliotecas e pacotes externos disponíveis para realizar importações. Além disso, a plataforma disponibiliza ainda aceleradores de hardware, nomeadamente GPU e TPU, tendo sido utilizado o TPU nesta solução.

8.1 Bibliotecas e pacotes externos utilizados

- **OpenCV:** Biblioteca Open-Source de visão computacional e de machine learning. Contém um número imenso de funções de machine learning com possíveis aplicações neste caso, como filtros, algoritmos de processamento e redimensionamento de imagens.
- **Scipy e Numpy:** Duas bibliotecas de análise de dados, com ferramentas de análise de dados obtidos e de modelação dos mesmos.
- **Pandas:** Ferramenta de análise e manipulação de dados, utilizado para leitura e escrita em ficheiros externos.

- **Tensorflow e Keras:** APIs de machine learning e de deep learning, com função de implementação das redes neuronais e com biblioteca de várias arquiteturas com pesos da rede já predefinidos, tendo por base imagens da base de dados ImageNet tal como foi utilizado: arquitetura DenseNet121 [3].
- **Scikit learn (sklearn):** Ferramentas para análise de dados e de criação de gráficos com base nos dados obtidos, obtendo métricas de avaliação como a precisão e o erro, tal como utilizado.
- **Matplotlib e Plotly:** Ferramentas para obtenção de imagens e gráficos no output.
- **Tqdm:** Implementação da barra de progresso.

Referências

- [1] - <https://www.kaggle.com/c/plant-pathology-2020-fgvc7>
- [2] - Thapa, R., Snaveley, N., Belongie, S., & Khan, A. (2020). *The Plant Pathology 2020 challenge dataset to classify foliar disease of apples*. *arXiv preprint arXiv:2004.11958*.
- [3] - <https://keras.io/api/applications/densenet/>