Classification of Pneumonia Using Deep Autoencoder for Chest X-ray Images

FEUP | MECD
Advanced Topics in Machine Learning

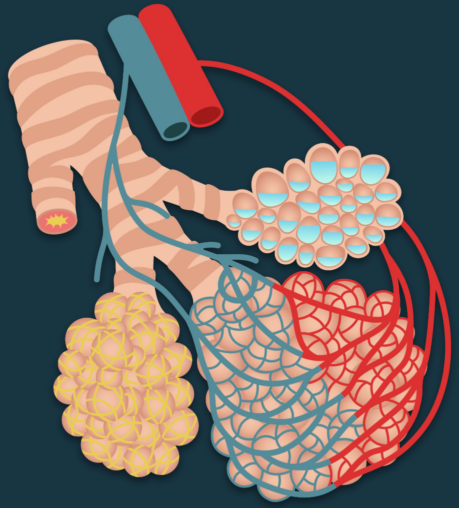
Team :

- Karim Kousa (up202102687)
- Silvia Tavares (up202204392)
- Sónia Ferreira (up202200967)
- Francisco Cuervo (up202302145)

Table of contents

01 Introduction

- Pneumonia is a respiratory disease that causes inflammation in one or both lungs, resulting in symptoms such as cough, fever, and difficulty breathing.
- Deep-learning algorithms can be trained on large datasets of chest X-rays to recognize patterns and features.
- Once trained, they can be used to classify new chest X-rays as either showing signs of pneumonia or not. This can be done in real time, making it a potentially valuable tool for healthcare professionals in diagnosing and treating patients with pneumonia.



02 Project Description

Dataset

- Sourced from Kaggle[1], it comprises **5,863 JPEG X-ray images from pediatric patients** aged one to five years, categorized into Pneumonia and Normal classes.
- The images are organized into 3 folders (train, test, and validation) and contain subfolders for each image category (Pneumonia/Normal).

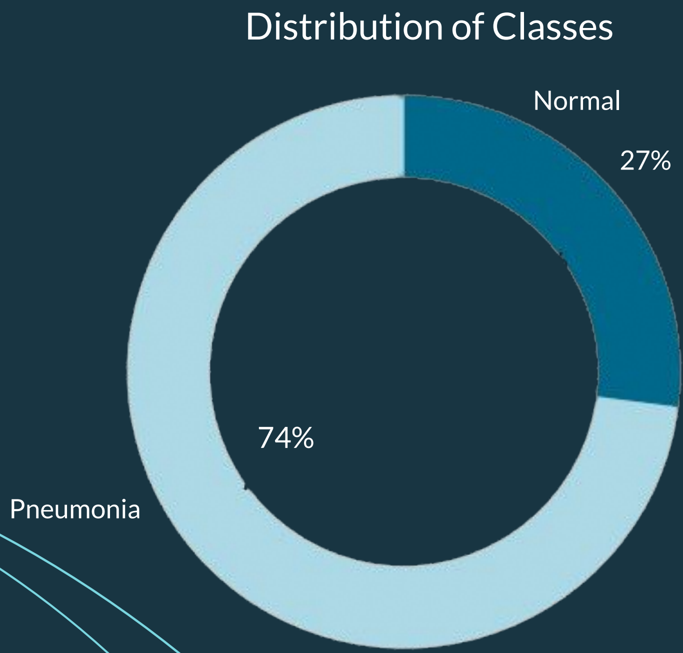
Fig. 1: Examples of X-Rays in Patients with Pneumonia

Proposal

- Perform Pneumonia classification using chest X-ray images a binary classification task to determine whether an X-ray image indicates the presence or absence of pneumonia.
- The goal is that the encoder part of the autoencoder compresses the input X-ray images into a lower-dimensional latent space. Additionally, we implemented VAE, CNN and pre-trained models like VGG, ResNet and DenseNet.

[1] https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

03 Data Exploration



The dataset has originally a dataset of 4273 images of pneumonia and 1583 images of normal x-rays, so is very unbalance.

The dataset was unbalanced so we resample it and got in the end 80% for train, 10% for validation and 10% test.

In order to improve it, data transformation were made on the images, like resizing them at 256x256 and adjust pixels´ scale and distribution.

04 Encoder-Decoder

Autoencoder Architecture :

Encoder :

- EfficientNet-B4 to encode input images. The fully connected layers, average pooling, and dropout layers of EfficientNet-B4 are removed to obtain a feature representation.

Decoder :

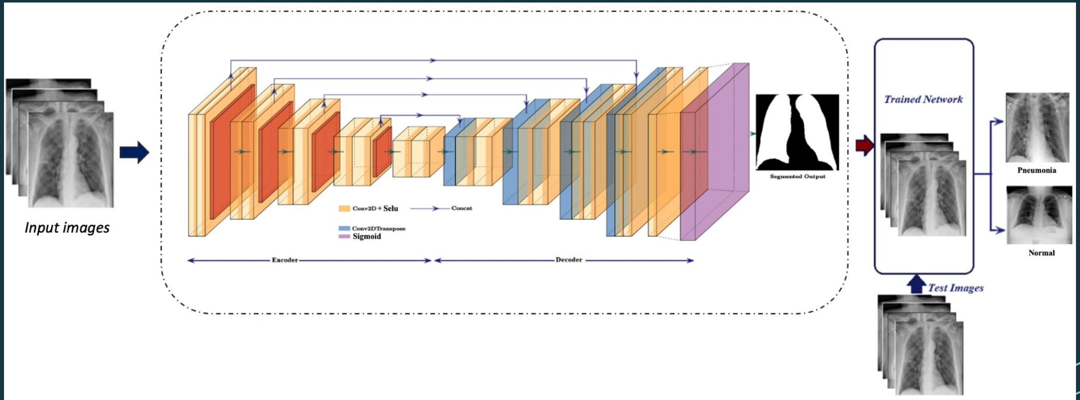
- Composed by series of transpose convolutional layers (ConvTranspose2d) followed by SELU activations. It aims to reconstruct the original image from the encoded features. The last layer uses the Sigmoid activation to scale the output between 0 and 1, making it suitable for image reconstruction.

Forward Pass :

- Pass input through the encoder, flattening the output, padding it to match the required size, and then passing it through the decoder.

Training the Autoencoder:

- Use Cross-Entropy loss and Adam optimizer. The autoencoder is trained to minimize the reconstruction loss.



04 Encoder-Decoder

Workflow:

First we train autoencoder for 10 epochs, extract features using the trained autoencoder, and train a binary classifier using the encoded features. Additionally, we included the evaluation of the binary classifier on validation and test sets, as well as the visualization of the confusion matrix.

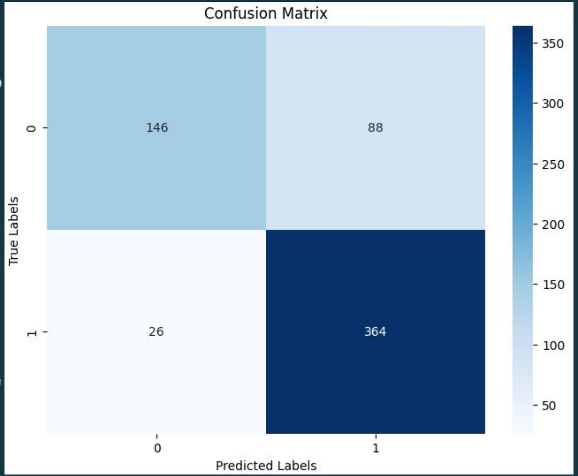
Train and Validation Results:

- Train Loss: 0.3327 ;
- Train Accuracy: 85.98%;
- Train Precision: 89.39%;
- Train Recall: 92.06%;
- Train F1: 90.70%

- Val Loss: 0.3247;
- Val Accuracy: 87.16%;
- Val Precision: 91.69%;
- Val Recall: 90.98%;
- Val F1: 91.33%

Test Results:

- Loss: 0.4562;
- Accuracy: 81.73%;
- Precision: 80.53%;
- Recall: 93.33%;
- F1: 86.46%



05 VAE

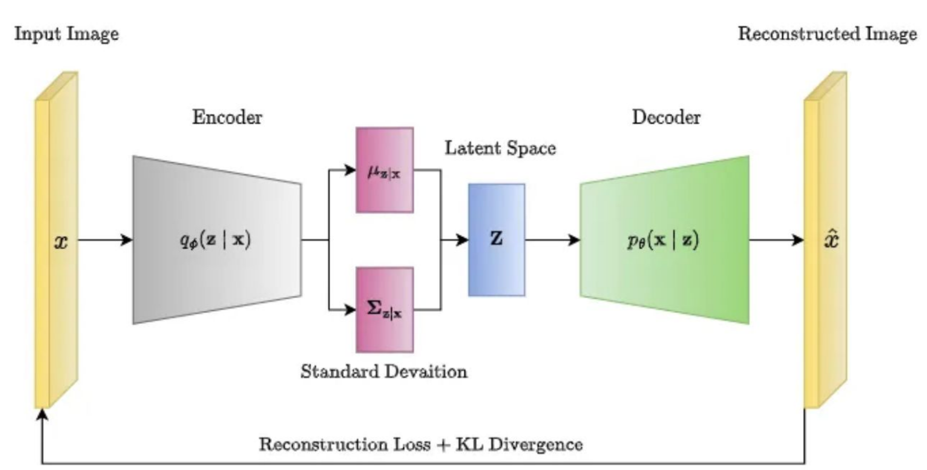
Architecture :

- The encoder uses EfficientNet-B4 to encode input images, and fully connected layers are added to compute the mean and log-variance of the distribution in the latent space.
- The decoder is a simple fully connected neural network that reconstructs the original image from the latent space.
- Reparameterization Trick: Reparameterize method is used to sample from the latent space using the mean and log-variance.

Loss Function: Reconstruction loss (MSE) and a KL divergence term, which encourages the latent space to follow a simple distribution.

Model Training: VAE is trained for 10 epochs using the Adam optimizer.

Extracting Features: Features are extracted from the VAE for both the training and validation sets.



05 VAE

Workflow:

VAE is trained to learn a probabilistic mapping from input images to a latent space, incorporating variational inference for improved generative modeling. The learned latent representations are then utilized to train a binary classifier. The binary classifier is designed using a simple fully connected neural network, and its training involves using the latent features extracted by the pre-trained VAE. The entire pipeline, from VAE trained to binary classifier is then evaluated on validation and test sets.

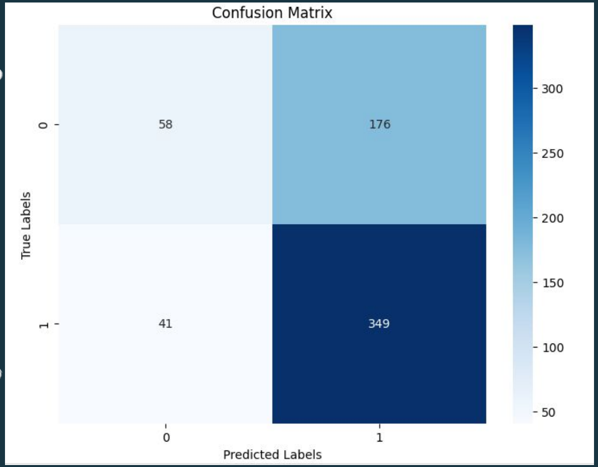
Train and Validation Results:

- Train Loss: 0.4835 ;
- Train Accuracy: 76.95%;
- Train Precision: 79.19%;
- Train Recall: 93.55%;
- Train F1: 85.77%

- Val Loss: 0.4806;
- Val Accuracy: 79.89%;
- Val Precision: 83.77%;
- Val Recall: 90.46%;
- Val F1: 86.99%

Test Results:

- Loss: 0.7136;
- Accuracy: 65.22%;
- Precision: 66.48%;
- Recall: 89.49%;
- F1: 76.28%



06 Convolutional Neural Network (CNN)

Inspired by the human visual system, it excels at processing visual data and capturing local patterns within large images.

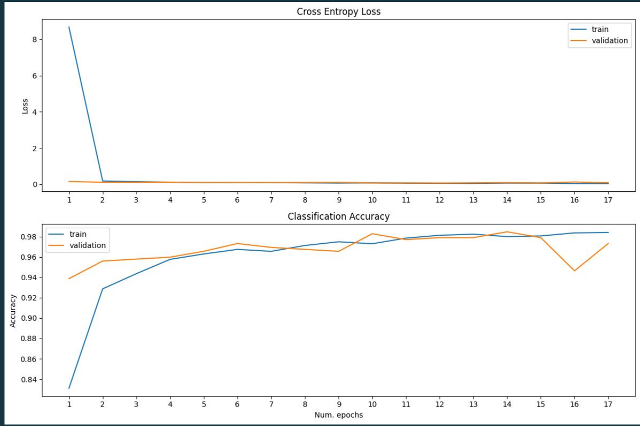
Developed architecture:

- Based on **ConvNet principles** for efficient, less complex feature extraction.
- **Architecture:** multiple convolutional and max-pooling layers, ending with a fully connected layer for classification.
- **Image Size:** 256x256 pixels to harness CNN's capacity for more significant inputs.

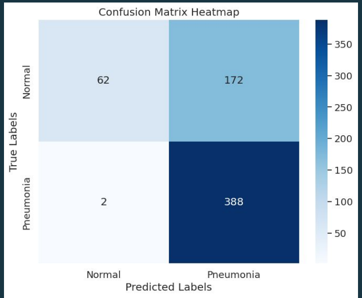
Hyperparameters:

- Adam optimizer; Cross-entropy loss function; Batch size: 32; Learning Rate: 0.01.
- Trained for 30 epochs, with an early stopping patience of 5.

Train Results



Test Confusion Matrix



Train Results:

- Early Stop: 17th epoch
- The best results, at the 12th epoch, with an **accuracy** of **Train: 98.1% ; Validation: 97.9%**.

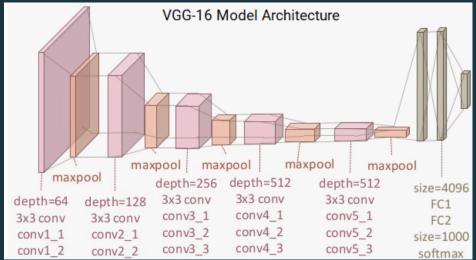
Test Results:

- Loss:1.418; Accuracy:72.1%; Precision:79.6%; Recall:72.1%; F1:66.7%

07 Transfer Learning Models

07.01 VGG16

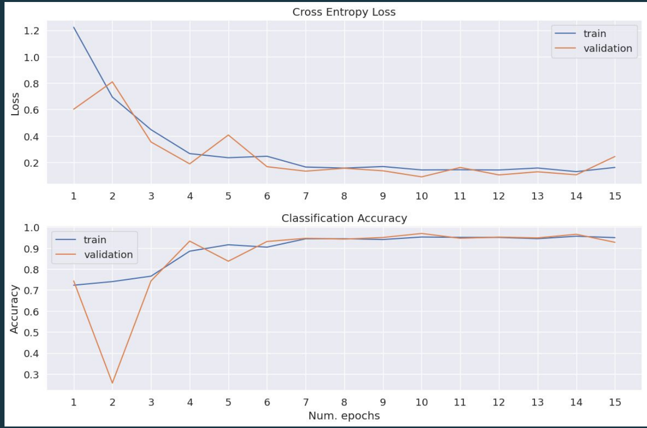
- It is known for its simplicity and effectiveness in image classification.
- Extensively trained on ImageNet, enabling detailed feature extraction for low-level visual details.



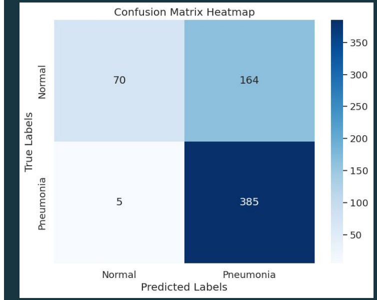
VGG16 defined hyperparameters:

- Adam optimizer; Cross-entropy loss function; Batch size: 32, Learning Rate: 0.001 with a step size of 10 epochs; Momentum: 0.9; Weight Decay of 0.01.
- Trained for 30 epochs, with an early stopping patience of 5.

Train Results



Test Confusion Matrix



Train Results:

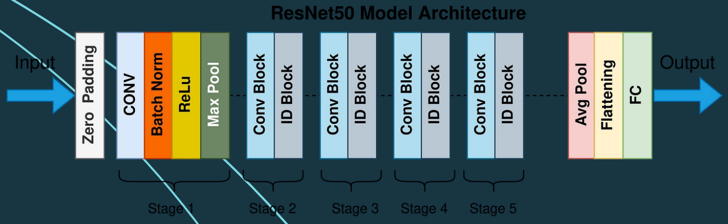
- Early Stop: 15th epoch
- The best results, at the 10th epoch, with an **accuracy** of **Train: 95.2% ; Validation: 96.9%**.

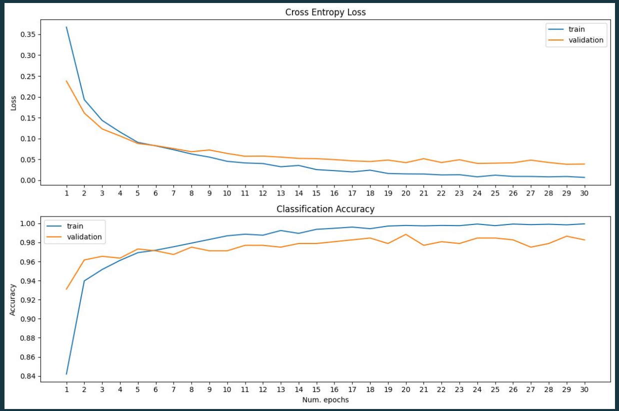
Test Results:

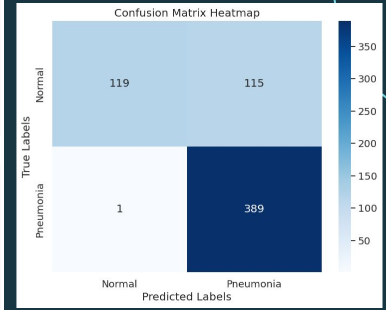
- Loss:0.836; Accuracy:72.9%; Precision:78.8%; Recall:72.9%; F1:68.9%

07.02 ResNet

- Initialization:
  - Cross-entropy loss function suitable for multi-class classification.
- Optimizer and Scheduler Setup:
  - Adam optimizer.
  - Learning rate scheduler: 0.001
  - Batch size: 32
- Training:
  - 30 epochs.







Train Results:

- **Best results: Epoch 29th**
- **Train acc: 99,9%**
- **Validation acc: 98,7%**

Test Result:

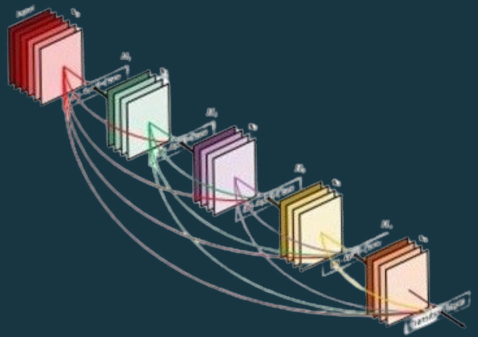
- **Test Loss: 0.691**
- **Test Accuracy: 81.4%**
- **Test Precision: 85.4%**
- **Test Recall: 81.4 %**
- **Test F1: 79.6 %**

07.03 DenseNet

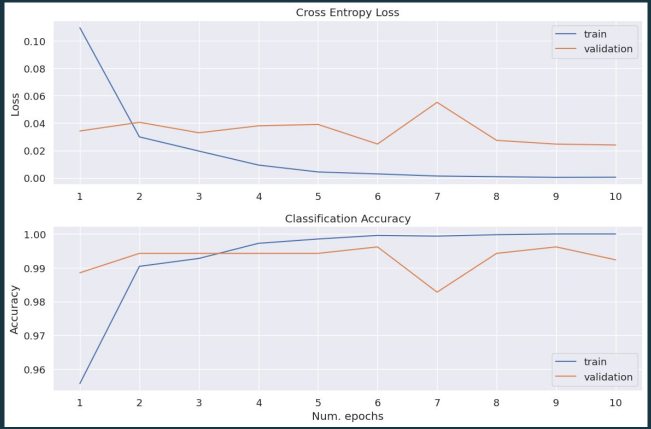
Initialization: Define the loss function as cross-entropy for multi-class classification.

Optimizer and Scheduler Setup: Adam optimizer; Learning rate scheduler: 0.005; Momentum: 0.9; Batch size: 32

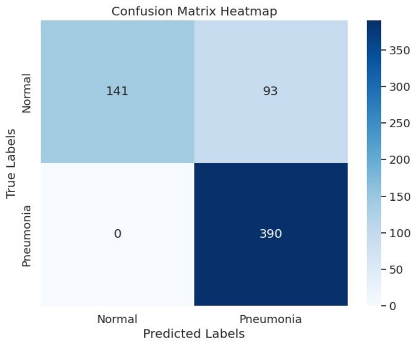
Training Configuration: 10 epochs.



Train Results



Test Confusion Matrix



Train Results:

- The best results, at the 10th epoch, with an **accuracy** of **Train: 100% ; Validation: 99,6%**.

Test Results:

- Loss:1.118 ; Accuracy:85.1%; Precision: 88.0%; Recall:72.1%; F1:84.0%

08 Results

Training and Validation parameters and results comparison per model

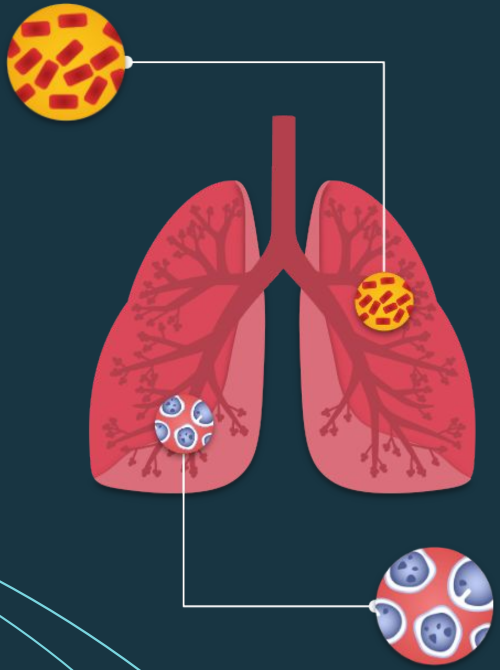
Models	Nr of Epochs	Params		Training		Validation	
		Trainable	Non-trainable	Accuracy (%)	Loss	Accuracy (%)	Loss
AutoEncoder	10	1003605779	0	85.9	0.3327	87.1	0.3247
VAE	10	121242736	0	76.9	0.4835	79.8	0.4806
CNN	30	19578946	0	98.1	0.0520	97.9	0.0670
VGG16	30	134268738	0	95.2	0.1420	96.9	0.0890
ResNet	30	23512130	0	99.9	0.009	98.7	0.038
DenseNet	10	4022920	0	100	0.000	99.2	0.024

Testing results comparison per model

Measures	AutoEncoder	VAE	CNN	VGG16	ResNet	DenseNet
Loss	0.4562	0.7136	1.418	0.836	0.6910	1.118
Accuracy (%)	81.7	65.2	72.1	72.9	81.4	85.1
Precision (%)	80.53	66.48	79.6	78.8	85.4	88.0
Recall (%)	93.33	89.49	72.1	72.9	81.4	85.1
F1 (%)	86.46	76.28	66.7	68.2	79.6	84.0

- **DenseNet** achieved a testing accuracy of 85.1%, with excellent precision and recall, **demonstrating its robustness across several metrics,** but high accuracy on train show a tendency to overfitting**.**
- **Resnet** a test accuracy of 81.4% with a good balance in precision and recall, **demonstrating its dependability.**
- **The AutoEncoder,** while showing a moderate drop in testing accuracy **excelled in recall, making it very efficient in identifying relevant cases. Same can be said for VAE** despite their lower overall accuracy and precision.
- **CNN and VGG16** results suggests that the models are experiencing overfitting in the data.

09 Conclusion



- Traditional diagnosis relies heavily on interpreting chest X-rays, which can be complex and subtle, requiring expertise and experience. With this project, we hope to contribute valuable insights into applying deep learning in medical imaging, paving the way for more advanced, reliable, and efficient diagnostic tools. Our exploration of various deep-learning models highlights the trade-offs between precision, recall, and accuracy in medical image analysis.

- It’s important to note that these models should be used as decision support tools, and qualified healthcare professionals should always confirm the final diagnosis.

- For the future work, can include the enhancement of hyperparameters, using advanced deep learning architectures, or hybrid models that can combine strengths with the ones used in this project.

Q&A

Thanks for your presence!

Contacts:

Karim Kousa - up202102687@fe.up.pt
Silvia Tavares - up202204392@fe.up.pt
Sónia Ferreira - up202200967@fe.up.pt
Francisco Cuervo - up202302145@fe.up.pt