Binary classifier (MLP model) for glomerular lesions

This repository consists of a complete pipeline for a binary classifier used for identfying the presence (or not) of lesions in glomerular images, which is composed (mainly) of two parts:

1. A series of pre-processing transformations that
   - Generate an augmented dataset via gamma correction, Gaussian suavization, Laplace filters and histogram equalization (folder `spatial-transforms`)
   - Extract features LBP, GLCM, contrast vector and Sobel filter from each image taken from the augmented dataset (folder `features`)
2. A MLP (Multilayer Perceptron) model (notebook `glomerulus.ipynb`) that receives a feature vector - the concatenation of all the previous features - for each image in the augmented dataset, and tries to predict the class out of two options:
   - Normal (no lesions)
   - Presence of crescent formations

This codebase is the accumulated result of four different assignments in the graduate course "Tópicos Especiais em Mecatrônica IV" (originally "Tópicos em Computação Visual I" - which roughly translates to "Topics in Computer Vision I").

However, since the dataset is private, this project can not be run, and serves only as an approximate mirror of the work done.

Results obtained

Three different network architectures were trained in this experiment - the results for the first architecture were outlined below (the other two achieved similar value for the measured metrics). Each model's complete set of logs and plots - that give the full overview of its perfomance - is in the respective model's folder.

More specifically, the following three arhcitectures were used, as seen in the following table:

| Parâmetros	Modelo 1	Modelo 2	Modelo 3
Batch size	64	64	64
Camadas escondidas	[1024, 1024, 512, 128]	[64, 128, 128, 32]	[32, 512, 1024, 128]
Dropouts associados	[0.65, 0.3, 0.3, 0.3, 0.15]	[0.4, 0.1, 0.2, 0.2, 0.1]	[0.4, 0.1, 0.3, 0.3, 0.1]
Weight decay (regularização L2)	0	0	10^{-4}
Otimizador	Adam	Adam	SGD
Learning rate	$3 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	$3 \cdot 10^{-4}$

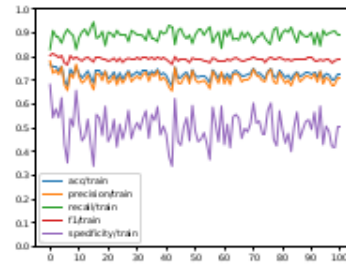Figure 1: Table comparing the structure of the 3 trained models
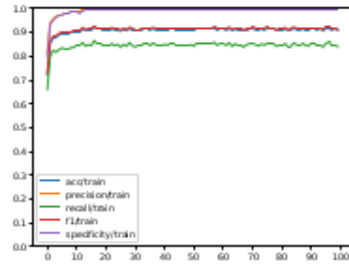
(a) Evolução das métricas de desempenho para o modelo 1 de MLP, no dataset de treino.
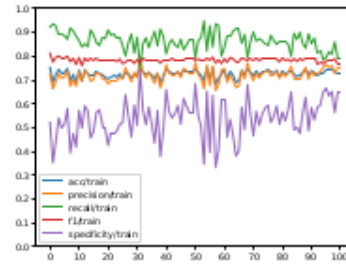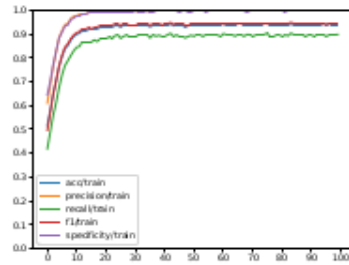
(b) Evolução das métricas de desempenho para o modelo 1 de MLP, no dataset de teste.
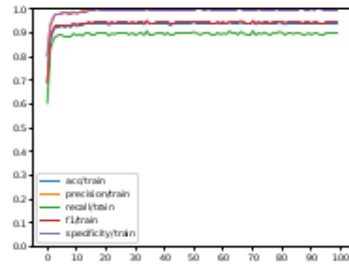
(c) Evolução das métricas de desempenho para o modelo 2 de MLP, no dataset de treino.

(d) Evolução das métricas de desempenho para o modelo 2 de MLP, no dataset de teste.

(e) Evolução das métricas de desempenho para o modelo 3 de MLP, no dataset de treino.

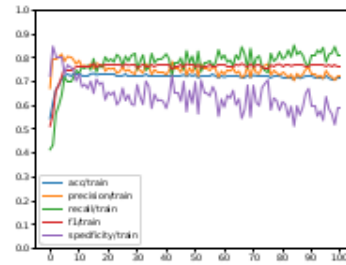(f) Evolução das métricas de desempenho para o modelo 3 de MLP, no dataset de teste.

Figure 2: Plot of training and test metrics (accuracy, precision, recall, specificity, F1) for each of the 3 models
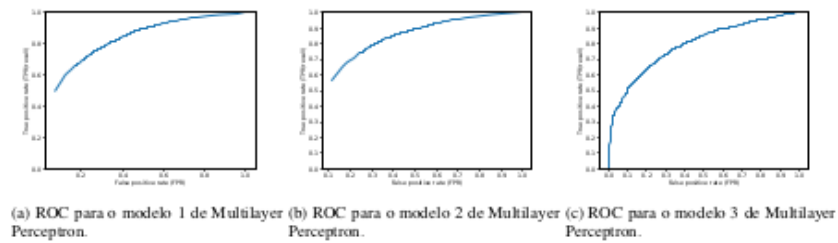
(a) ROC para o modelo 1 de Multilayer (b) ROC para o modelo 2 de Multilayer (c) ROC para o modelo 3 de Multilayer
Perceptron.   Perceptron.   Perceptron.

Figure 3: Plot of ROC curves for each of the 3 models

(a) Precision-Recall para o modelo 1 de (b) Precision-Recall para o modelo 2 de (c) Precision-Recall para o modelo 3 de
Multilayer Perceptron.   Multilayer Perceptron.   Multilayer Perceptron.
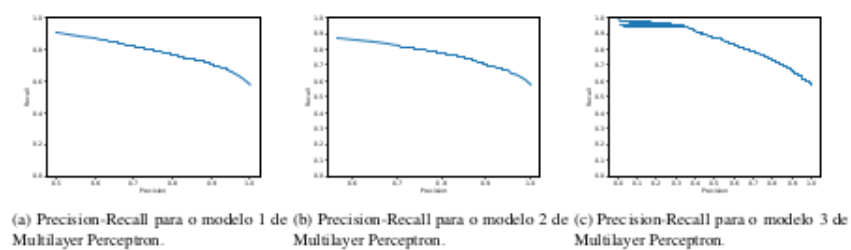
Figure 4: Plot of Precision-Recall/PR curves for each of the 3 models
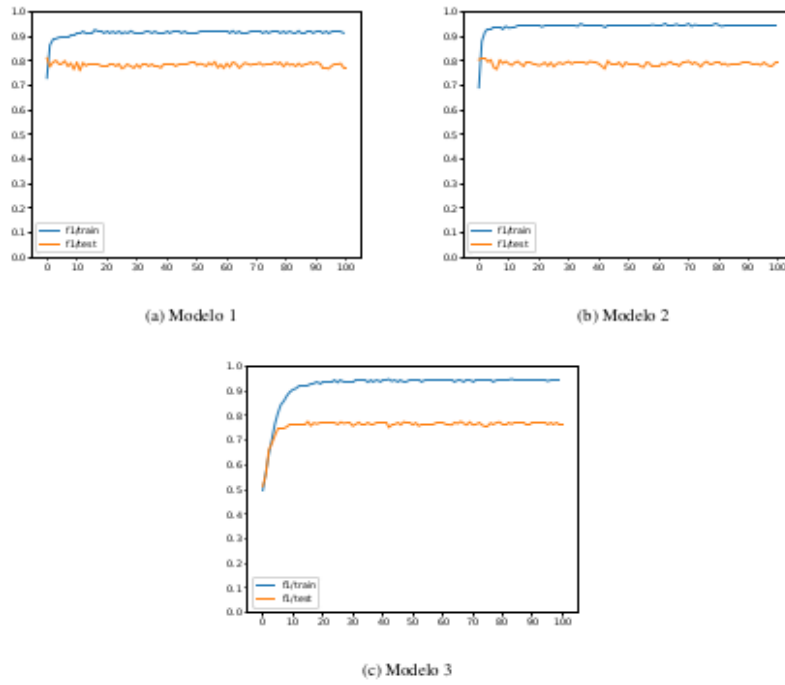
(a) Modelo 1

(b) Modelo 2

(c) Modelo 3

Figure 5: Plot of simultaneous evolution of training and test F1 score for each of the 3 models