# Elaboração de novo experimento baseado no escopo proposto na última reunião

## Resumo

Subconjuntos são construído a partir de um conjunto com mais de 12.000 imagens médicas de Raio-X da região do tórax para que possam ser processados em computadores de baixo desempenho. As técnicas EHD, SIFT, LBP, Gabor e Zernike são utilizadas para caracterizar as amostras dos conjuntos formados. Os conjuntos criados são utilizados para treinar e testar os modelos criados por classificadores multirrótulo. Por fim, é avaliado o desempenho dos algoritmos selecionados para a tarefa de classificar imagens com múltiplos rótulos.

## Abstract

Subset are constructed from a set with more than 12.000 ray-X medical images from breast region to can processed in low-performance computer. The EHD, SIFT, LBP, Gabor and Zernike techniques are used to feature the samples from formed set. The created sets are used to train and test the created model by multi-label classifiers. Finally, are evaluated the performance from selected algorithms to the task of classify images with multiples labels.

## Fase 0 - Construção do ambiente de avaliação

Main.java

### Descrição

Recebe as entradas para a execução das fases seguintes e registra em um arquivo os logs do experimento.

### Algoritmo

1. Inicio o registro de logs do experimento chamado **Experiments04-Exe[yyyymmddhhmm].log**;
2. Abro o arquivo com os parâmetros de entrada para as fases seguintes;
3. Executo cada uma das fases enviando os parâmetros de entrada necessários;
4. Finalizo o registro de logs do experimento.

### Entradas

1. Arquivo **Conf.ini**.

### Saídas

1. Arquivo de log do experimento **Experiments04-Exe[yyyymmddhhmm].log**.

## Fase 1 - Preparação dos dados

Preparing.java

### Descrição

Obter NS subconjuntos de NI imagens, sem reposição, do conjunto de imagens IRMA CP que não contém a extensão TIF.

### Algoritmo

1. Insiro em uma lista o nome de todas as imagens;
2. Crio uma lista e insiro os nomes de NI imagens obtidas a partir da remoção aleatória de nomes da lista principal. Gravo no arquivo **Experiments04-Exe[yyyymmddhhmm]-Sub[nn].lst** arquivo para liberar RAM (definir padrão para o nome do arquivo).
3. Repito esta operação (NS-1) vezes.

### Entradas

1. Caminho para as imagens;
2. Número de subconjuntos NS do conjunto principal CP, tal que NS <= CP/NI;
3. Número de imagens NI em cada subconjunto de NS, tal que NI <= CP/NS.

### Saídas

1. NS arquivos LST com os nomes das imagens.

## Fase 2 - Construção das bases ARFF

Building.java

### Descrição

Para cada subconjunto, construir cinco bases ARFF. Uma para cada técnica de extração de características.

Técnicas de extração de características utilizadas

* 1. EHD;
  2. Histograma SIFT;
  3. Histograma LBP;
  4. Gabor;
  5. Momentos de Zernike;

### Algoritmo

1. Verifico se há alguma técnica de extração de características a ser utilizada. Caso contrário, encerro a fase;
2. Construo o cabeçalho da base ARFF para o tipo de característica em questão;
3. Para cada imagem do subconjunto, extraio as características e crio uma amostra;
4. Armazeno a amostra na base ARFF;
5. Salva a base ARFF e armazeno no arquivo **Experiments04-Exe[yyyymmddhhmm]-Tec-Sub[nn].arff**.

### Entradas

1. Técnicas de extração de características que devem ser utilizadas;
2. Caminho para a estrutura de códigos IRMA;
3. Caminho para o arquivo que relaciona nome da imagem com o respectivo rótulo.

### Saídas

1. (5 \* NS) bases ARFF.

## Fase 3 - Avaliação dos classificadores

Evaluating.java

### Descrição

As bases ARFF construídas deverão ser usadas da seguinte maneira:

1. Uma base deverá ser o conjunto de treinamento usado para criar o modelo do classificador;
2. As (NS-1) demais serão os conjuntos de testes usados para avaliação do modelo criado.

Para a avaliação dos classificadores serão consideradas as médias das medidas obtidas na avaliação dos nove conjuntos de testes.

Os classificadores previstos para serem avaliados são:

1. Multilabel kNN (MLkNN);
2. Binary Relevance kNN (BRkNN);
3. ClassifierChain;
4. CLUS.

### Algoritmo

1. Verifica se há um classificador a ser avaliado. Caso contrário, encerro fase;
2. Instancio o conjunto de treino;
3. Instancio o classificador;
4. Construo um modelo para o classificador a partir do conjunto de treinamento;
5. Instancio um avaliador;
6. Instancio o conjunto de teste;
7. Instancio uma lista de medidas;
8. Instancio uma avaliação e atribuo a ela o resultado do classificador avaliado;
9. Armazeno em arquivo o resultado da avaliação para liberar RAM;
10. Repito (NS-2) vezes o algoritmo a partir do passo 5.

### Entradas

1. Classificadores que serão avaliados.

### Saídas

1. (NS-1) resultados de avaliação.