**Machine Learning (Aprendizagem de Máquina)**

**Trabalho Prático**

**Tema**: Classificação de Imagens

**Desafio**: Criar e avaliar modelos de predição para a classificação de imagens usando os indutores aprendidos durante as aulas (Árvore de Decisão, Naive Bayes, K-Vizinhos mais Próximos, Redes Neurais, Máquina de Vetor de Suporte), ou a combinação destes em ensembles homogêneos ou heterogêneos.

**Materiais disponíveis**:

**Locais**: scripts para treinamento de classificadores ([www.ppgia.pucprp.br/~alceu/am](http://www.ppgia.pucprp.br/~alceu/am))

e no **Blackboard** um script para extração de características, e a base de imagens.

1. **Base de imagens**: a base de imagens fornecida para criação dos modelos possui 10 classes e 1000 imagens, sendo 100 exemplos de cada classe. As imagens estão organizadas em pastas com o nome das classes: 'cavalos', 'comida', 'dino', 'elefante', 'flores', 'humanos', 'montanhas', 'obras', 'onibus' e 'praia'.
2. **Extrator de Atributos** (*FeatureExtraction*): este programa em Python usa as bibliotecas OpenCV, Skimage e Keras para realizar a extração de características a partir das imagens fornecidas. Duas abordagens são consideradas para representação das imagens. Uma formada por *handcrafted features,* ou seja, um vetor composto por descritores de cor (histogramas RGB), forma (HOG, Histogramas de Orientação do Gradiente) e textura (LBP, Local Binary Pattern). Tais descritores são armazenados em vetor com 3387 entradas (valores reais entre 0 e 1). A outra abordagem de representação utiliza a extração de características via rede convolucional (CNN, denominada InceptionV3) pré-treinada na base de imagens ImageNet, gerando um vetor com 2048 valores para cada imagem.

**Entrada do script**: as imagens devem estar no Google Drive organizadas em pastas. Na versão original deste script as imagens estão dentro de diretórios com os nomes das classes, por exemplo: as imagens contendo cavalos estão na pasta:

/content/drive/My Drive/Base/cavalos

**Saída do script**: este geram três arquivos no formato “csv”, sendo:

y.csv : arquivo contendo o rótulo de cada imagem representado como um inteiro de 0 a 9. São 1000 entradas, cada uma contendo um valor entre 0 e 9.

X.csv: arquivo com as *handcrafted features*, sendo 1000 entradas, cada uma contendo 3387 valores.

X\_deep.csv: arquivo contendo as *deep features* extraídas usando a CNN já treinada em outra base maior denominada ImageNet. Total de 2048 valores para cada imagem.

**Resultados esperados**:

1. Geração de um modelo de predição (classificador) para cada abordagem de representação.
2. Comparação das duas abordagens de representação e indicação do melhor resultado.
3. Taxa de acerto dos modelos avaliados
4. Matriz de confusão dos melhores resultados.
5. Análise de erros

**Material de apoio:**

Link para funções do OpenCV/Skimage utilizadas no cálculo dos descritores de cor, forma e textura.

* Histogramas de cor (cv2.calcHist()): calcula a quantidade de pixels de cada cor na imagem. Observe que isto foi feito para cada canal do sistema de cor RGB (Red, Green, Blue). Isto nos dá um vetor de 256x3=768 atributos. Os valores resultantes foram divididos pelo total de pixels da imagem (normalização).

<https://docs.opencv.org/2.4/modules/imgproc/doc/histograms.html>

* Local Binary Pattern (LBP, feature.local\_binary\_pattern()): calcula um histograma de códigos de textura. Usamos o LBP uniforme, o que nos dá um vetor com 59 atributos, também normalizado pelo total de códigos de textura na imagem.

<https://www.pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>

* HOG (hog()): calcula histograma de direções do gradiente, ou seja, informações de forma dos objetos na imagem. O HOG é estimado com base na contagem de direções (8 neste caso) encontradas nos contornos dos objetos presentes na imagem. Total de 2560 atributos.

<https://scikit-image.org/docs/dev/api/skimage.feature.html#skimage.feature.hog>

<https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html>

Link para uma descrição da rede InceptionV3:

<https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202>