

Projeto 01 – Classificação de imagens

O conjunto de imagens “MPEG-7_lite” foi gerada a partir do conjunto “MPEG7_CE-Shape-1_Part_B”¹

O conjunto “MPEG-7_lite” é composto por 720 imagens binárias pertencentes a 4 classes. Cada classe consiste em 180 imagens contendo a forma de algum objeto: **0: apple; 1: bat; 2: beetle; 3: ‘bone’**. Todas as imagens são binárias, possuem tamanhos diferentes (largura Vs. altura) e um único objeto representado.

O conjunto foi dividido em **folds** de treino e teste com uma proporção de 70% e 30%, respectivamente.

1) Treinar um classificador para discriminar entre as classes do dataset “MPEG-7_lite”.

- Utilize os classificadores fornecidos com o **scikit-learn**: **K-nn**, **Bayes** ou **SVM**.
- Escolha um conjunto de características de forma usando o método **region_props** da biblioteca **scikit-image**.
 - Dica 1: Plote as características e analise visualmente para tentar identificar as mais adequadas para discriminar as imagens.
 - Dica 2: Visualizar as imagens também pode ajudar a conhecer melhor o problema que estão buscando resolver.
 - Podem usar características além das utilizadas nas aulas.
 - Podem implementar novas características, caso queiram
- Normalize as características.
 - Pode ser manualmente ou usando as classes do **scikit-learn**.
- Utilize o conjunto de validação para otimizar os hiperparâmetros.
 - Como não é fornecido um conjunto de validação, é necessário separar parte do conjunto de treino como conjunto de validação.
 - Dica: Uma proporção entre 20 e 30% do conjunto de treinamento é adequado.
 - Usar o método **train_test_split** do **scikit-learn**.
 - **(Obrigatório)** Defina o argumento **random_state=393**
 - Consulte a documentação do **scikit-learn** para informações sobre hiperparâmetros que podem ser otimizados.

¹https://www.imageprocessingplace.com/downloads_V3/root_downloads/image_databases/MPEG7_CE-Shape-1_Part_B.zip
<https://dabi.temple.edu/external/shape/MPEG7/MPEG7dataset.zip>
<http://www.timeseriesclassification.com/description.php?Dataset=ShapesAll>

Obs.: O conjunto de testes fornecido para vocês é **falso** e não deve ser utilizado para tomar nenhuma conclusão sobre o desempenho do modelo. O conjunto de testes **real** está comigo, e eu usarei para definir o melhor modelo.

- Entregar um arquivo .ipynb ou .py com o melhor modelo que vocês encontraram:
 - Melhor classificador,
 - Melhor conjunto de características,
 - Melhores hiperparâmetros.
 - Manter o código que vocês usaram para localizar os hiperparâmetros.
- **(Obrigatório)** O programa deve, obrigatoriamente, imprimir a matriz de confusão e relatório de classificação para o conjunto de **validação** (gerado por vocês) e de **testes** (o conjunto falso, fornecido).
- **(Obrigatório)** Fixe uma semente de geração de números aleatórios no início do seu código, para evitar flutuações nos resultados.
 - `np.random.seed(393)`
- O grupo que conseguir a melhor **acurácia** sobre o conjunto de **testes** será o grupo vencedor.

- **Entrega: 23/11/2022.**
 - Grupos de 2 pessoas.
 - Criar um repositório publico no GitHub.
 - Enviar o link via PVANet Moodle
 - Redigir um arquivo README.md com as instruções para executar o programa.
 - Executar em um ambiente conda com as criado da seguinte forma:
 - `$ conda create -n env-proj1-393-py38 python=3.8`
 - `$ conda activate env-proj1-393-py38`
 - `$ pip install sklearn==1.1.3`
 - `$ pip install scikit-image==0.19.3`
 - `$ pip install matplotlib==3.6.0`
 - `$ pip install pandas==1.5.1`
 - `$ pip install seaborn==0.12.1`
 - `$ pip install notebook==6.5.1`
 - Se precisar de mais alguma biblioteca, informar no README.md