

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**INSTITUTO METROPOLE DIGITAL**  
**IMD1101 - Aprendizado de Máquina – 2022.2**  
**Aula11 – Extração de características baseadas em HOG**

A identificação de objetos em imagens tem se tornado um desafio interessante. A partir disso, sua aplicação tem sido verificada em várias de interesse. Dentre os descritores de características mais utilizados está o *Histogram of Oriented Gradients* (HOG). Esse descritor foi proposto em 1994, mas ficou conhecido em 2005 em trabalho de detecção de pedestres em imagens.

O descritor HOG pode ser visto como um descritor de gradientes ou das direções das bordas, e tem aplicação em uma série de tarefas e pode ser usado em diversas linguagens, como C++, Python, Java através de bibliotecas como OpenCV ou Scikit-image.

Visando a elaboração do projeto da disciplina de Aprendizado de Máquina, escolheu-se uma base de imagens de pets (cachorros e gatos) que pode ser acessa através do seguinte link:

- <https://www.robots.ox.ac.uk/~vgg/data/pets/>

Para a extração de características das referidas imagens, cada grupo deverá seguir a seguinte metodologia:

1. Ler as imagens baixadas do site;

```
#exemplo de código abrindo uma imagem do pc, convertendo em scikit image e extraíndo l
from skimage.transform import resize
from skimage.feature import hog
from skimage.io import imread, imshow
import numpy
import PIL
from google.colab import files
from io import BytesIO
from PIL import Image

uploaded = files.upload()

img = PIL.Image.open(BytesIO(uploaded['american_bulldog_66.jpg']))
imgarr = numpy.array(img)

imshow(imgarr)
print(imgarr.shape)
```

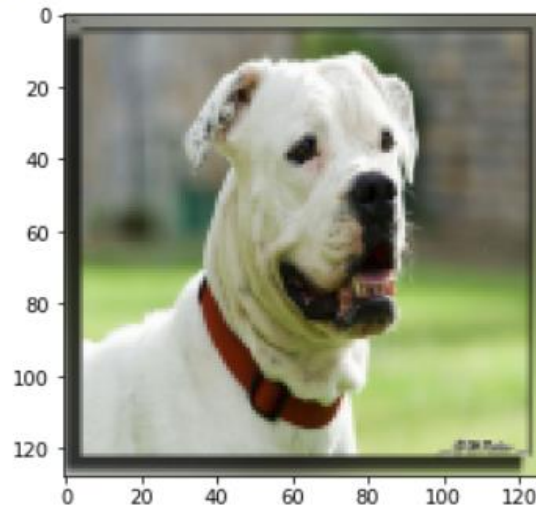
- **american\_bulldog\_66.jpg**(image/jpeg) - 90975 bytes, last modified: 18/06/2012 - 100% d  
Saving american\_bulldog\_66.jpg to american\_bulldog\_66 (6).jpg  
(334, 500, 3)



2. Redefinir o tamanho delas;

```
[26] resized_img = resize(imgarr, (128,128))  
     imshow(resized_img)  
     print(resized_img.shape)
```

(128, 128, 3)



3. Aplicar o HOG para gerar duas versões diferentes das imagens referentes a cada grupos, utilizando as seguintes configurações:

```
fd, hog_image = hog(resized_img, orientations=9, pixels_per_cell=(16, 16),  
                    cells_per_block=(2, 2), visualize=True, multichannel=True)  
  
fd.shape  
  
(1764,)
```

```
fd, hog_image = hog(resized_img, orientations=9, pixels_per_cell=(20, 20),  
                    cells_per_block=(2, 2), visualize=True, multichannel=True)  
  
fd.shape  
  
(900,)
```

4. Montar os dois dataset em csv contendo 1000 imagens, sendo 600 imagens de cachorros e 400 imagens de gatos de acordo com a numeração de cada grupo;
5. Para o primeiro dataset com 1764 atributos, os grupos deverão gerar duas versões aplicando os métodos PCA e seleção de atributos baseada em correlação.
6. De posse de quatro base de dados (duas obtidas através da utilização do HOG e duas obtidas através da utilização de PCA e seleção de atributos), utilize as seguintes opções de treinamento e teste:
  - a. 10-fold cross-validation;
  - b. split-percentage: 90/10, 80/20 e 70/30;

Execute experimentos com as técnicas k-NN e Árvore de Decisão, de tal forma que você possa escolher diferentes valores para alguns parâmetros pertencentes às técnicas estudadas.

Para o k-NN, você deverá variar o número de vizinhos de 1 a 5 (ou mais, até que o desempenho estabilize), de acordo com a Figura 1. Para cada execução, de acordo com os valores de  $k$  e referentes a cada uma das opções de treinamento e teste, guarde o resultado da **acurácia** na célula correspondente, como mostra a figura abaixo. Por último, calcule a média e o desvio padrão para cada valor de  $k$ .

IBK (k-NN)						
Base	Treinamento/Teste	1k	2k	3k	4k	5k
Metodologias		Acc	Acc	Acc	Acc	Acc
Base Original	10-fold CV					
	70/30					
	80/20					
	90/10					
Base v01_70	10-fold CV					
	70/30					
	80/20					
	90/10					
Base v02_50	10-fold CV					
	70/30					
	80/20					
	90/10					
Base v03_30	10-fold CV					
	70/30					
	80/20					
	90/10					
Média =>		#####	#####	#####	#####	#####
Desv. Pad. =>		#####	#####	#####	#####	#####

Figura 1. Resultados experimentais do k-NN (variação de  $k$  de 1 a 5).

DecisionTree Classifier						
Base	Treinamento/Teste	md = 3	md = 4	md = 5	md = 6	md = 7
Metodologias		Acc	Acc	Acc	Acc	Acc
Base Original	10-fold CV					
	70/30					
	80/20					
	90/10					
Base v01_70	10-fold CV					
	70/30					
	80/20					
	90/10					
Base v02_50	10-fold CV					
	70/30					
	80/20					
	90/10					
Base v03_30	10-fold CV					
	70/30					
	80/20					
	90/10					
Média =>		#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
Desv. Pad. =>		#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
Max_Depth = md						

Figura 2. Resultados experimentais do DecisionTree Classifier utilizando diferentes valores para o fator de confiança.

Para a Árvore de Decisão, você deverá escolher diferentes valores para a profundidade da árvore (*max\_depth*), de acordo com a Figura 2. Lembre-se que você poderá explorar um conjunto de valores maior do que está sendo mostrado na referida figura. Além disso, utilize a mesma metodologia de experimentos descrita acima.

Observe que depois de fazer as execuções e guardar todos os valores dos experimentos, crie um documento com imagem para mostrar a completude da tarefa. Esse documento deverá ser salvo em DPF e submetido via SIGAA. Sua submissão valerá a presença referente a essa aula assíncrona.

Bom trabalho!