



# Parte 1: Tarefas de Manipulação

*Projeto de Trainee da área de Visão Computacional*



## Introdução



*"Muito com imagens a fazer você ainda tem antes de classificar",* Mestre Yoda para Luke Skywalker em seu treinamento em Dagobah.

Nesse momento você já está (ou deveria estar) em posse do dataset. No entanto, em sua jornada você perceberá que dificilmente um dataset virá com imagens boas e suficientes para um bom resultado num projeto de classificação. Assim, é extremamente importante sabermos tratar imagens corretamente. Nessa primeira parte de sua missão, você irá aprender algumas das técnicas para isso.

## Tarefa 1

Seu primeiro desafio consiste na simples (porém talvez não tão simples na primeira vez) tarefa de visualizar as imagens do seu dataset.

Crie uma função que recebe o endereço de uma imagem e retorna sua visualização. Por aprendizado, faça isso de três formas diferentes: uma com OpenCV, outra com Pillow e outra com Matplotlib.

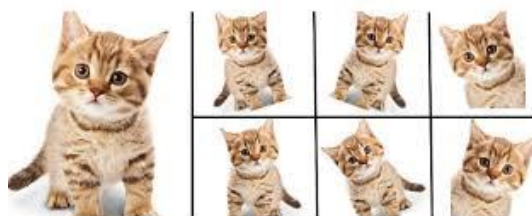
## Tarefa 2

Sua segunda tarefa é aprender a lidar com os espaços de cores.

Descubra como converter uma imagem colorida em uma imagem preto e branco e escreva uma função usando numpy que o faz. Depois veja como converter espaços de cores usando OpenCV. Leia também sobre a diferença entre espaço RGB e BGR. Qual é usado pela biblioteca OpenCV?

## Tarefa 3

Como a maioria dos datasets não possuem um número suficiente de imagens para nosso modelo aprender a distinguir com grande precisão as diversas classes, costumamos aplicar a técnica de Augmentation no nosso dataset. Image Augmentation consiste em fazer algumas modificações em imagens existentes para criar imagens que contenham informações novinhas para nosso modelo.



**Enlarge your Dataset**

Na imagem acima, podemos observar duas modificações usadas: rotação e flip. Nessa tarefa, você usará as seguintes técnicas:

- Inverter a imagem (flip)
- Rotacionar de um ângulo aleatório
- Zoom aleatório na imagem
- Contraste aleatório
- Brilho aleatório

Você deve criar uma função para cada técnica. Escolha pelo menos duas das técnicas para implementar na mão usando numpy. Depois disso, descubra como fazer tais augmentations com as bibliotecas de manipulação de imagens (TensorFlow, PyTorch, OpenCV) e implemente as 5 acima listadas usando alguma lib.

Por fim, crie uma função que recebe uma imagem e aplique o processo de augmentation nela, usando as funções previamente criadas nessa tarefa.

Obs: Aqui você terá que criar uma função para cada, mas quando for treinar seu modelo, há formas de fazer isso direto, procure nas documentações do TensorFlow e/ou do PyTorch.

## Tarefa 4

Outro processo comum é o de aplicar filtros para melhorar a qualidade da imagem, dentre outros resultados que se pode obter. Para isso, aplique em uma imagem de sua escolha um filtro de blur, gaussian blur e median blur (pode usar OpenCV, mas se quiser aprender como fazer na mão vai ser um

grande aprendizado). Para entender mais como os filtros funcionam, crie um kernel de sua escolha e aplique na sua imagem utilizando OpenCV. Agora pode brincar um pouquinho tentando imaginar o que o filtro vai fazer com sua imagem.

Há diversos tipos de filtro que podem ser usados em uma imagem, é uma técnica bem conhecida para detecção de bordas em imagens.

Os filtros de blur costumam ser bons quando queremos um modelo mais robusto, que detecta nossa imagem mesmo com “noise”. Os filtros de média são bons quando queremos reduzir o “noise” da nossa imagem, e torná-la mais uniforme.

## Links úteis para as tarefas

Recomendamos sempre se aventurar nas documentações, tutoriais no medium, toward data science e pyimagesearch, mas no nosso drive de CV também temos alguns vídeos rápidos de ver que dão uma ótima introduzida em visão computacional.

Também indicamos alguns links que podem ser úteis caso você trave em alguma tarefa:

- 1.1
  - <https://www.geeksforgeeks.org/python-pil-image-show-method/>
  - [https://matplotlib.org/3.2.1/api/\\_as\\_gen/matplotlib.pyplot.imshow.html](https://matplotlib.org/3.2.1/api/_as_gen/matplotlib.pyplot.imshow.html)

- [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_gui/py\\_image\\_display/py\\_image\\_display.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_gui/py_image_display/py_image_display.html)
- <https://learn.datacamp.com/courses/image-processing-in-python>
- 1.2
  - [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_colorspaces/py\\_colorspaces.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_colorspaces/py_colorspaces.html)
  - <https://medium.com/tulisan-ibe/image-conversion-using-python-and-opencv-8cd7c0ef7051>
- 1.3
  - <https://towardsdatascience.com/data-augmentation-for-deep-learning-4fe21d1a4eb9>
  - <https://sigmoidal.ai/reduzindo-overfitting-com-data-augmentation/>
  - [https://www.tensorflow.org/tutorials/images/data\\_augmentation](https://www.tensorflow.org/tutorials/images/data_augmentation) (mais legal c:)
- 1.4
  - <https://towardsdatascience.com/image-filters-in-python-26ee938e57d2>
  - [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_filtering/py\\_filtering.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html)