



CURSO: Engenharia de Computação (EC)

Projeto

Rev.: 20/07/2023

Disciplina: OPCPF-Processamento Digital de Imagens

Professor: Fabio Alexandre SPANHOL, faspanhol@gmail.com

Turma: CP94

Período: 9

Turno: Integral

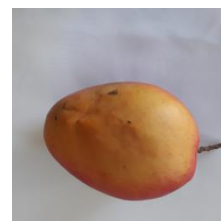
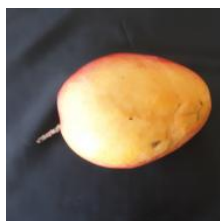
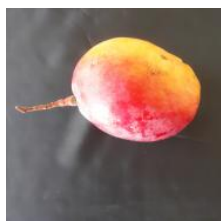
1

Crie uma **base de imagens**. Utilize o **GitHub** para hospedar o projeto.

- Escolha um dos temas: (a) **fruits** (ft-DB) ou (b) **kitchen utensils** (KiUt-DB).
- Nomeie os itens em inglês com **underscore** substituindo espaços quando necessário. Exemplos: (a) orange, apple, banana, mango, pineapple, guava, clementine, fig, etc. e (b) glass, mug, fork, pizza\_cuter, spoon, pasta\_server, pan, etc.
- Defina 10 classes, isto é, 10 frutas ou 10 utensílios. Crie uma tabela de codificação das classes, enumerando as classes 0...n-1. Ordene alfabeticamente os nomes das classes para codificar. Ex.:

0	apple
1	banana
2	clementine
...	...
9	yellow_passion_fruit

- Para cada item capture duas imagens em ângulos de visão diferentes (V1 e V2), variando dois fundos distintos (B e W). No mínimo 5 itens e no máximo 10.



- Utilize seu **smartphone** ou câmera digital para capturar imagens dos itens escolhidos.
- Aplique o mesmo corte para que todas as imagens tenham as mesmas dimensões (W x H).
- Exporte as imagens para o formato PNG.
- Codifique os nomes dos arquivos das imagens utilizando a regra **<CLASSID>-<IMG\_SEQUENCE>-V1|V2-B|W.png**

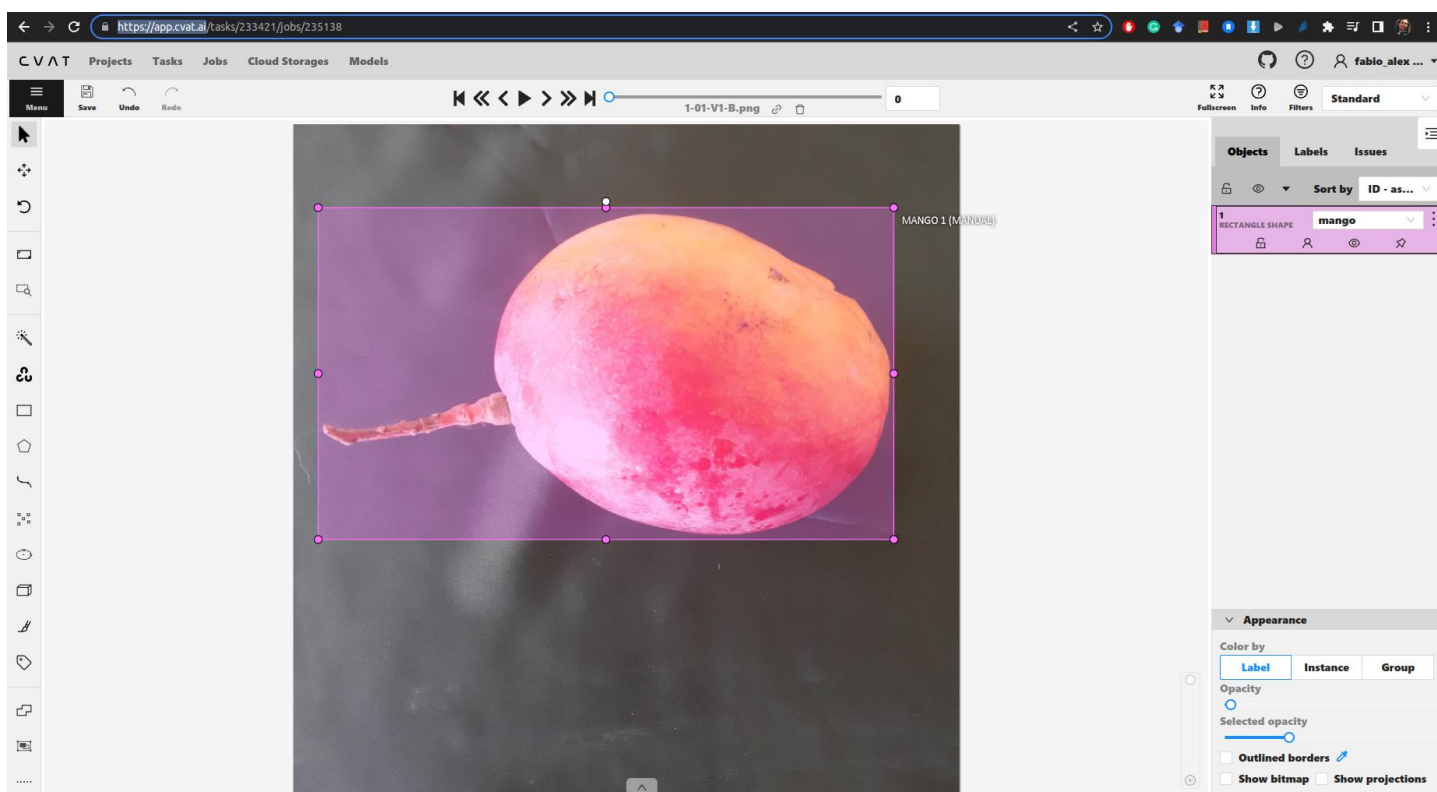
Exemplo:

```
0-01-V1-B.png
0-01-V1-W.png
0-01-V2-B.png
0-01-V2-W.png
...
0-10-V2-B.png
0-10-V2-W.png
```



- Organize uma estrutura de diretório com um diretório para cada classe identificado com o nome da respectiva classe. Compacte a base de imagens em um arquivo tar.gz.
- Crie um arquivo de metadados.
- Crie um **Jupyter Notebook (NB)** para exibir as imagens, organizadas pelas classes. Dica: `skimage.util.montage`
- Crie uma função para exibir os metadados.

**2** **Data annotation.** Faça a anotação dos objetos nas imagens, isto é, a marcação da **bounding box** e a respectiva classe. Sugestão: Uso da ferramenta **Web CVAT** disponível em <https://app.cvat.ai/>



**3** **Data augmentation.** Crie um **augmented\_dataset**: uma versão aumentada em 3 vezes da base de imagens original.

- Aplique funções de **data augmentation** para aumentar a variabilidade de imagens da base. Veja <https://www.kaggle.com/code/parulpandey/overview-of-popular-image-augmentation-packages/notebook>
- Aplique as funções:
  - Logaritmo da imagem
  - Exponencial da imagem
  - Filtro da média usando convolução



- Crie um **Jupyter Notebook (NB)** para exibir as imagens, organizadas pelas classes. Dica: `skimage.util montage`

#### 4 **Data normalization.** Crie um **normalized\_dataset** utilizando equalização de histogramas.

- Calcule e apresente

Protótipo médio de cada classe  
Histograma médio de cada classe  
Variância do histograma de cada classe

#### 5 **Segmentação do objeto de interesse.**

- Segmente o objeto do seu fundo, gerando uma imagem binária (máscara): 0 para o fundo e 1 para o objeto. Dois tipos de segmentação serão usados (a) manual, para geração do **ground truth** e (b) automática usando algum algoritmo. Escolha um. Exemplo:

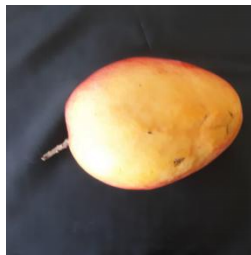
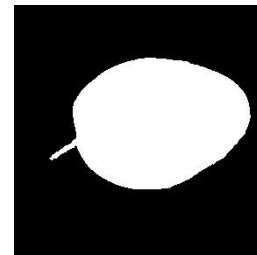


Imagem original



Ground truth

- O **ground truth** deve ser gerado para pelo menos 25% das amostras de cada classe.
- Crie um NB para gerar **bounding box (bounding rectangle)** nas imagens.

#### 6 **Classificação.**

- Para cada imagem com **bounding box** extraia características, gerando um **feature vector**.
- Treine um classificador. Separe a base de imagens em 80% para treinamento, 10% para validação e 10% para teste.
- Calcule e apresente as métricas de classificação acurácia média e F1-score.
- Apresente uma matriz de confusão.



*"Today everything exists to end in a photograph."*

*Susan Sontag(1933-2004). American writer, philosopher, and political activist.*