Potato Crop Disease Classification using YOLOv7, YOLOv8, Faster-R-CNN, Retinanet Frameworks

The dataset used:

- Download the RGB Image Patches - Augmented Dataset from the link provided. => Link
- Use Roboflow or other websites to convert the Pascalvoc to coco format for mmdetection and use Pascalvoc to .txt for yolov7 and yolov8.

Note: These codes were executed on Google Co-lab. So, if you want to run the code on jupyter notebook, then ensure that you have installed all the dependencies before running the code.

Common step [0]: Mount google drive.

```
[ ]  from google.colab import drive
     drive.mount('/content/drive')
```

YOLOv7:

Link to the drive: yolov7_drive_link

YOLOv7_Potato.ipynb is provided for the reference

1.
```
!git clone https://github.com/WongKinYiu/yolov7
```
2. Install required dependencies
3. Unzip the custom dataset by providing the correct path.

```
[ ]  %cd /content
     !unzip /content/drive/MyDrive/Classes/Yolov7/data_yolov7.zip
```

4. The zip file contains two folder images and labels. The images and labels folders further contain two folders train and val.

5. Change the path of the images

```
[ ]  train_img_path = "/content/images/train"
     val_img_path = "/content/images/val"
```

6. Here we are writing the path of all the images in train.txt

```
[ ]  #Training images
     with open('train.txt', "a+") as f:
       img_list = os.listdir(train_img_path)
       for img in img_list:
         f.write(os.path.join(train_img_path,img+'\n'))
       print("Done")
```

```
Done
```

7. Similarly do for val.txt
8. Now make a data config file. Make a copy of Coco.yaml (yolov7/data/coco.yaml) and rename as custom. yaml.
9. Make the following changes in custom.yaml

```
1
2  train: /content/train.txt
3  val: /content/val.txt
4
5  #number of classes
6  nc: 2
7
8  #class names
9  names: ['healthy','stressed'] #class names
```

10. Download the pre-trained weight

```
[ ]  %cd /content/drive/MyDrive/Classes/Yolov7/yolov7
     !wget "https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt"
```

11. Make a copy of yolov7.yaml (yolov7/cfg/training/yolov7.yaml) and rename as custom_yolov7.yaml
12. In custom_yolov7.yaml change the number of classes from 80 to 2.
13.
```
!python train.py --batch 16 --
cfg /content/drive/MyDrive/Classes/Yolov7/yolov
7/cfg/training/custom_yolov7.yaml --
epochs 100 --
data /content/drive/MyDrive/Classes/Yolov7/yolo
v7/data/custom.yaml --
weights '/content/drive/MyDrive/Classes/Yolov7/
yolov7/yolov7.pt' --device 0
```

14. In 13 change the paths accordingly for training

15. After this follow the steps to do the result and inference.

- **YOLOv8:**

Link to drive: yolov8_drive_link

YOLOv8_potato_crop.ipynb is provided for reference. Here dataset contains 3 folders train, test and val. And each folder further contains the 2 folder images and labels.

1. Install Ultralytics

```
[ ]
    # Pip install method (recommended)

    !pip install ultralytics

    from IPython import display
    display.clear_output()

    import ultralytics
    ultralytics.checks()
```

2. Make data.yaml file and upload it in the folder. Paths of the folders are present in the notebook provided.

```
1 train: /content/drive/MyDrive/Classes/YOLOv8/123/data/train/images
2 val: /content/drive/MyDrive/Classes/YOLOv8/123/data/val/images
3
4 nc: 2
5 names: ["healthy","stressed"]
```

Change the paths of the train and val accordingly.

3. Train the model

```
%cd /content/drive/MyDrive/Classes/YOLOv8

!yolo task=detect mode=train model=yolov8s.pt data=data.yaml epochs=25 imgsz=750 plots=True
```

Task= detect; mode=train; model=yolo8s.pt; data=path to data.yaml

4. After this follow the steps to do the result and inference.

- **Faster-R-CNN (Mmdetection):**

Link to drive: Faster_rcnn_drive_link

Potato_disease_mmdetection_faster_rcnn.ipynb is provided for reference.

1. Here I am mentioning only the steps which must be changed while running the code. Strictly follow the Potato_disease_mmdetection_faster_rcnn.ipynb for correct output.
2. Change the config file accordingly. For this code use faster_rcnn_x101_64x4d_fpn_2x_coco.py

```
[ ] from mmcv import Config
    cfg = Config.fromfile('/content/drive/MyDrive/Classes/MMdetection/MMdetection1/configs/faster_rcnn/faster_rcnn_x101_64x4d_fpn_2x_coco.py')
```

3. Here the dataset is in the coco format.
4. Change the path and location of the images and annotations in the code shown below. Make the change for all three tests, train and val. Also, change the classes. In this case, the classes are healthy and stressed.

```
[ ] from mmdet.apis import set_random_seed

    path = '/content/drive/MyDrive/Classes/MMdetection/MMdetection1/configs/Potato_new/'

    # Modify dataset type and path
    cfg.dataset_type = 'COCODataset'
    cfg.data_root = path

    cfg.data.test.data_root = path
    cfg.data.test.ann_file = 'test/_annotations.coco.json'
    cfg.data.test.img_prefix = 'test'
    cfg.data.test.classes = ('healthy','stressed')

    cfg.data.train.data_root = path
    cfg.data.train.ann_file = 'train/_annotations.coco.json'
    cfg.data.train.img_prefix = 'train'
    cfg.data.train.classes = ('healthy','stressed')

    cfg.data.val.data_root = path
    cfg.data.val.ann_file = 'test/_annotations.coco.json'
    cfg.data.val.img_prefix = 'test'
    cfg.data.val.classes = ('healthy','stressed')
```

5. Download the model check_points from the mmdetection github link and change the path of cfg.load_from. The check points are in loadcheck_points folder.

```
# We can still the pre-trained Mask RCNN model to obtain a higher performance
cfg.load_from = '/content/drive/MyDrive/Classes/MMdetection/MMdetection1/loadcheck_points/faster_rcnn_x101_64x4d_fpn_2x_coco_20200512_161033-5961fa95.pth'
```

6. Set up the working directory to save files and logs.

```
# Set up working dir to save files and logs.
cfg.work_dir = '/content/drive/MyDrive/Classes/MMdetection/MMdetection1/wrk_test_3'
```

7. Change the number of epoch and other parameters.
8. Train the model and after that follow the steps to do the result and inference.

- **Retinanet (Mmdetection):**

Link to drive: Retinanet_drive_link

MMdetection_Retinanet.ipynb is provided for reference.

1. Here I am mentioning only the steps which must be changed while running the code. Strictly follow the MMdetection_Retinanet.ipynb for correct output.
2. Change the config file accordingly. For this code use retinanet_x101_64x4d_fpn_1x_coco.py

```
[ ] from mmcv import Config
    cfg = Config.fromfile('/content/drive/MyDrive/Classes/MMdetection/MMdetection1/configs/retinanet/retinanet_x101_64x4d_fpn_1x_coco.py')
```

3. Here the dataset is in the coco format.
4. Change the path and location of the images and annotations in the code shown below. Make the change for all three tests, train and val. Also, change the classes. In this case, the classes are healthy and stressed.

```
[ ]  from mmdet.apis import set_random_seed

     path = '/content/drive/MyDrive/Classes/MMdetection/MMdetection1/configs/Potato_new/'

     # Modify dataset type and path
     cfg.dataset_type = 'COCODataset'
     cfg.data_root = path

     cfg.data.test.data_root = path
     cfg.data.test.ann_file = 'test/_annotations.coco.json'
     cfg.data.test.img_prefix = 'test'
     cfg.data.test.classes = ('healthy','stressed')

     cfg.data.train.data_root = path
     cfg.data.train.ann_file = 'train/_annotations.coco.json'
     cfg.data.train.img_prefix = 'train'
     cfg.data.train.classes = ('healthy','stressed')

     cfg.data.val.data_root = path
     cfg.data.val.ann_file = 'test/_annotations.coco.json'
     cfg.data.val.img_prefix = 'test'
     cfg.data.val.classes = ('healthy','stressed')
```

5. Change the number of classes and remove roi_head. The changed code is shown below.

```
cfg.model.bbox_head.num_classes = 2
```

6. Download the model check_points from the mmdetection github link and change the path of cfg.load_from. The check points are in loadcheck_points folder.

```
# We can still the pre-trained Mask RCNN model to obtain a higher performance
cfg.load_from = '/content/drive/MyDrive/Classes/MMdetection/MMdetection1/loadcheck_points/retinanet_x101_64x4d_fpn_1x_coco_20200130-366f5af1.pth'
```

7. Set up the working directory to save files and logs.

```
# Set up working dir to save files and logs.
cfg.work_dir = '/content/drive/MyDrive/Classes/MMdetection/MMdetection1/wrk_test_retina1'
```

8. Change the number of epochs and other parameters.
9. Train the model and after that follow the steps to do the result and inference.