**README - Fruits Detection using Mask R-CNN**

**Project Overview**

This project implements an object detection model using Mask R-CNN to detect and classify fruits (apples, bananas, and oranges) in images. The model is trained on a custom dataset and evaluation metrics using Mean Average Precision (mAP) and Intersection over Union (IoU) metrics.

The Implementation of Mask R-CNN I used for this project: [Matterport Mask R-CNN](https://github.com/matterport/Mask_RCNN)

**Installation Requirements**

**Dependencies**

Libraries were used:

pip install tensorflow numpy matplotlib scikit-image mrcnn

Additional dependencies:

* Python 3.7.3
* numpy==1.20.3
* scipy==1.4.1
* Pillow==8.4.0
* cython==0.29.24
* matplotlib
* protobuf==3.20.0
* scikit-image==0.16.2
* tensorflow==2.0.0
* keras== 2.2.4
* opencv-python==4.5.4.60
* h5py==2.10.0
* imgaug==0.4.0
* Matterport Mask R-CNN library
* scikit-image
* ElementTree (for XML parsing)

**Dataset Structure**

The dataset should be organized as follows:

/dataset

/images\_sample

- image\_1.jpg

- image\_2.jpg

...

/annotation

- image\_1.xml

- image\_2.xml

...

* images\_sample/ contains images of fruits.
* annotation/ contains XML files with bounding box information.

**Training the Model**

1. **Prepare the Dataset**
   * The dataset is loaded and split into training and testing sets.
   * XML annotations are parsed to extract bounding boxes and class labels.
2. **Train the Model**
   * Load the COCO pre-trained weights (mask\_rcnn\_coco.h5).
   * Train for 3 epochs using the heads layers.

To start training, run:

python train\_model.py

**Inference and Testing**

After training, use the trained model to detect objects in new images:

python inference.py --image\_path path\_to\_test\_image.jpg

This will output the image with detected objects, bounding boxes, and confidence scores.

**Evaluation Metrics**

The model is evaluated using:

* **Mean Average Precision (mAP)**
* **Intersection over Union (IoU)**

Run evaluation with:

python evaluate.py

**Example Evaluation Results:**

A screenshot of a computer program

Description automatically generated

Mean AP: 0.2317

Mean IoU: 0.7284

**Future Improvements**

* Increase dataset size for better accuracy.
* Experiment with hyperparameters.
* Implement real-time object detection.
* Deploy on edge devices.

**License**

<https://github.com/ahmedfgad/Mask-RCNN-TF2/blob/master/LICENSE>