Comparison Report: YOLOv8 Detection vs Transfer Learning Classification

Dataset link:

https://www.kaggle.com/datasets/moltean/fruits

1. Summary

Two different approaches were applied to the same Fruits-360 dataset:
1. YOLOv8 Detection Model: Focused on object detection, identifying fruit types and their bounding boxes in images.
2. Transfer Learning Classification Model: Used a pre-trained CNN (with frozen layers initially) to classify whole images of fruits into categories.

2. YOLOv8 Model Observations

- Dataset was annotated manually using RoboFlow for a small subset(30 images).
- Pre-trained YOLOv8m weights were loaded and fine-tuned on the custom fruit dataset.
- The training ran for 10 epochs, with metrics like mAP, precision, and recall computed.
- The output included bounding boxes and class labels for each detected fruit.

3. Transfer Learning Model Observations

- Used a pre-trained CNN architecture(VGG16) as the base model.
- This code trains a fruit image classifier using a pretrained VGG16 model. First, the VGG16 model is loaded without its top layers, and new layers are added for the fruit classification task. At first, the original VGG16 layers are kept frozen, and only the new layers are trained for a few rounds. Then, the last part of VGG16 is unfrozen and trained again with a small learning rate to fine-tune the model. Finally, the model is tested, and its accuracy is reported.
- Model compiled with Adam optimizer and categorical cross-entropy loss.
- Training was conducted for 10 epochs, tracking training, validation, and test accuracy.

4. Model Comparison

YOLOv8:
- Strengths: Can detect multiple fruits in a single image, provides location

and class.
- Weaknesses: Requires more complex annotation (bounding boxes), training is heavier.

Transfer Learning:
- Strengths: Simpler to implement for classification tasks, needs only image-label pairs.
- Weaknesses: Cannot localize fruits, only predicts a single label per image.

5. Results

YOLOv8 results included detection visualizations with bounding boxes, while Transfer Learning provided numerical accuracy results on test data.
Both approaches achieved strong performance for their respective goals: YOLOv8 for object detection, Transfer Learning for image classification.

6. Requirements list:

For YOLO notebook:

- ultralytics

- os

- random

-shutil

- A sample of the train images

and then annotate it externally.

- yaml

- YOLO form ultralytics

- PIL

For VGG16 notebook:

- os

- pathlib

- tensorflow

- Seaborn

- sklearn