Better Ate Than Never

using deep learning and analytics to combat food waste

W251 Section 3
Final Project Presentation
Dec 9 2020

Team Members:
Dinesh Achuthan
Suryabrata Dutta
Ammara Essa
Girija Ghali

Background and Motivation

- Food insecurity impact more than 820 million people worldwide.
- Nearly 33% of food produced is wasted or lost worldwide
- 133 billion pounds (worth $161B) ends up landfills in the United States alone
- Food waste accounts for 21% of the waste stream and associated greenhouse gases

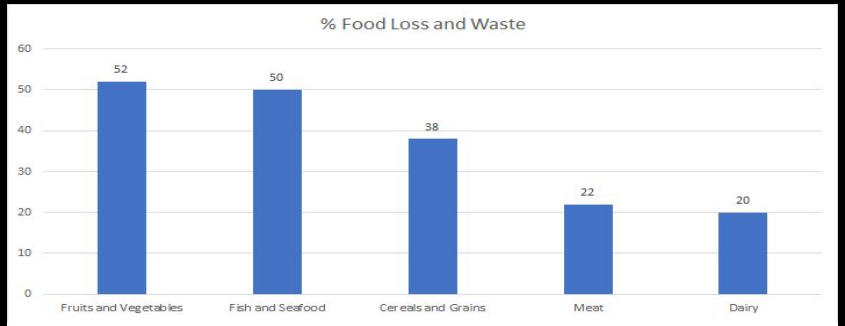


Fig 1. % Food Loss and Waste by Category (calculated collectively for USA, Canada, Australia, and New Zealand)
Source: FAO.org 2011

How Can We Curb Food Waste

- Fresh fruits and vegetables are among highest food items wasted
- **Solution:** Deep learning based inventory management system that
  - Weighs, detects and distinguishes between good produce and rotten fresh produce items
  - Apply analytics to the compiled data to gain insights into gaps in supply chain management by better characterizing food waste timelines, quantity and types of produce most wasted etc
- The system can be placed at various stages of the food cycle e.g.
  - At the time of production / harvest
  - Retail / grocery store level
  - In the consumer's home

Assumptions and Limitations

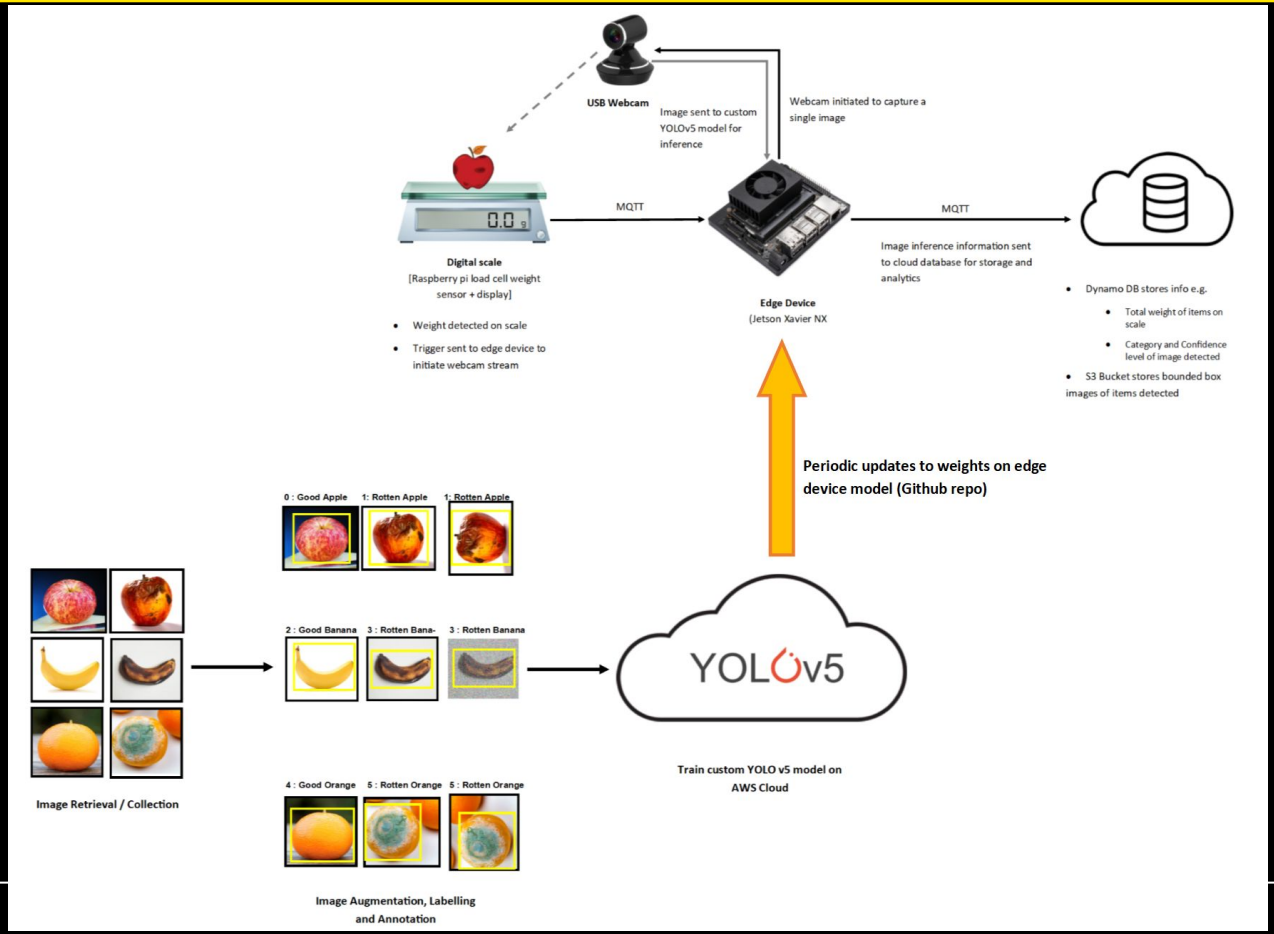
Assumptions

1. Only 6 categories
    a. Good Apple
    b. Rotten Apple
    c. Good Orange
    d. Rotten Orange
    e. Good Banana
    f. Rotten Banana
2. Only one E2E pipeline implemented
3. Manual model deployment to Xavier
4. mAP@0.5 IoU baseline

Constraints

1. Time!
2. Rotten food image dataset availability
3. Budget
    a. AWS instances
    b. Hardware
    c. Licensed images

End-to-End Pipeline Design



End-to-End Pipeline Tour (5 mins)

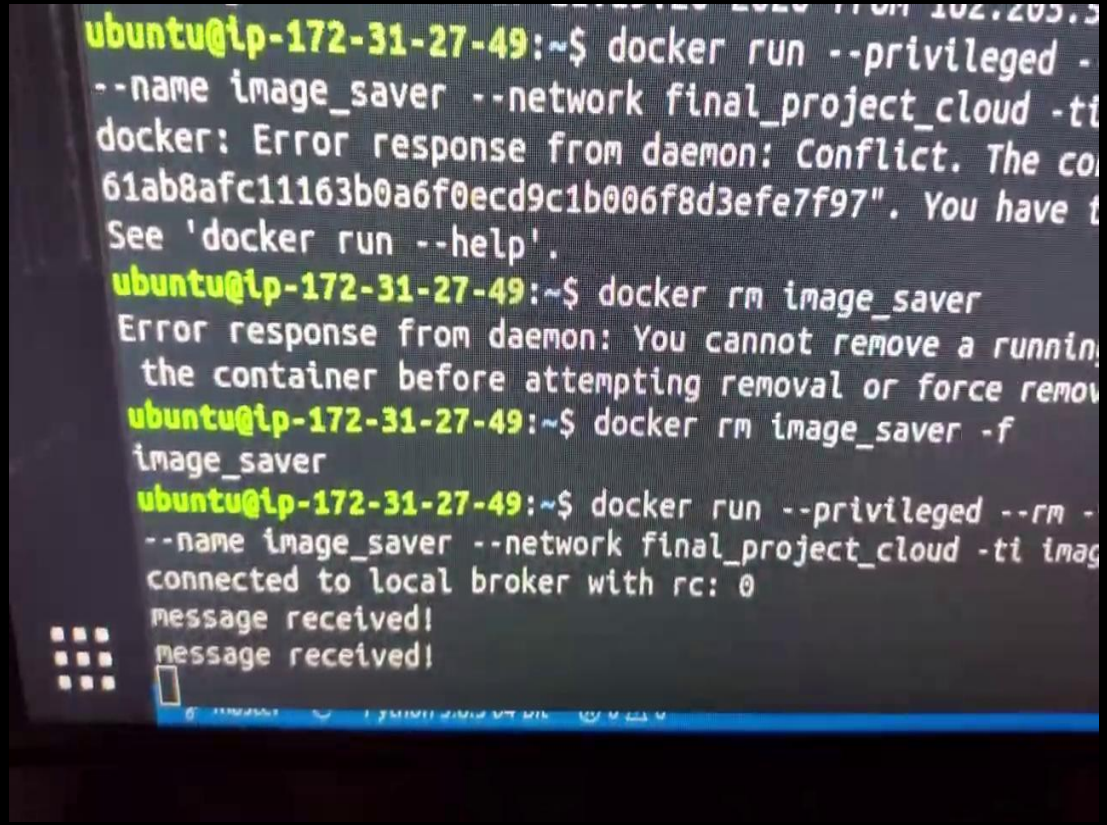
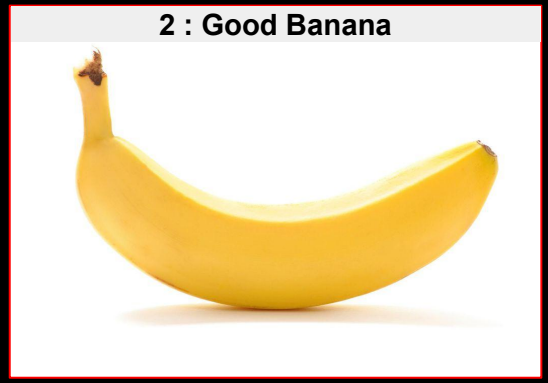


Image Retrieval and Augmentation : Good Fruit

- Good (Ex: Apple , Banana, Orange)
  - Image Retrieval - COCO Dataset
  - Image selection
  - Relabeling



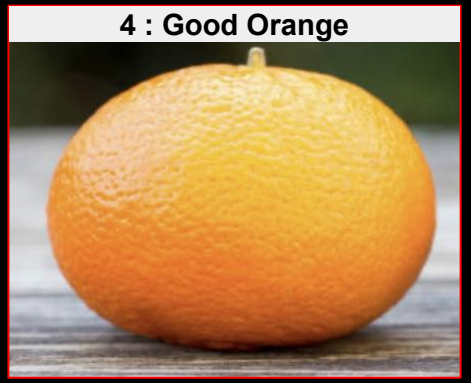


Image Retrieval and Augmentation : Good Fruit

Examples of images discarded

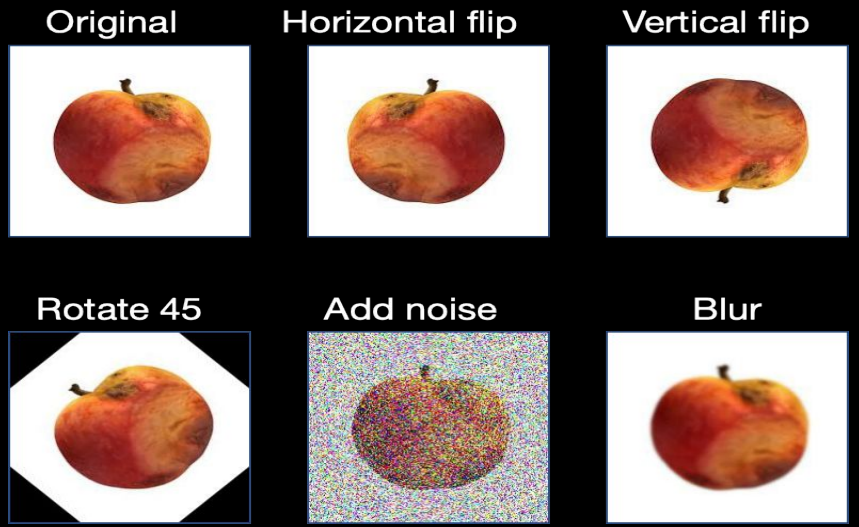


Examples of images used in model

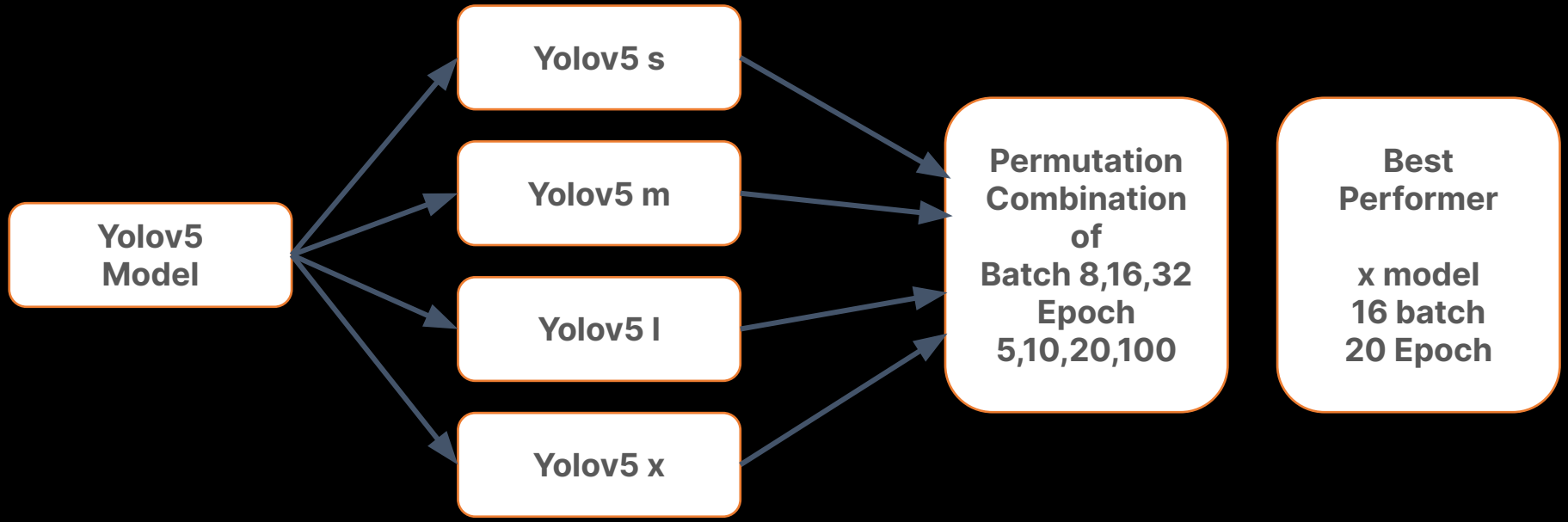


Image Retrieval and Augmentation : Rotten Fruit

- Rotten (Ex: Rotten-Apple, Rotten-Banana, Rotten-Orange)
  - Image Retrieval - web scraping
  - Annotation - makesense.ai
  - Augmentation
    - Horizontal Flip
    - Vertical Flip
    - Add noise
    - Blur
    - Rotate 45%



Model Training on Cloud



G4dn.2xlarge AWS machine is used. Anything above batch 16 errored out due to resource constraints. There is opportunity to further optimize the hyper parameters. Large .pt files posed challenges with git update.

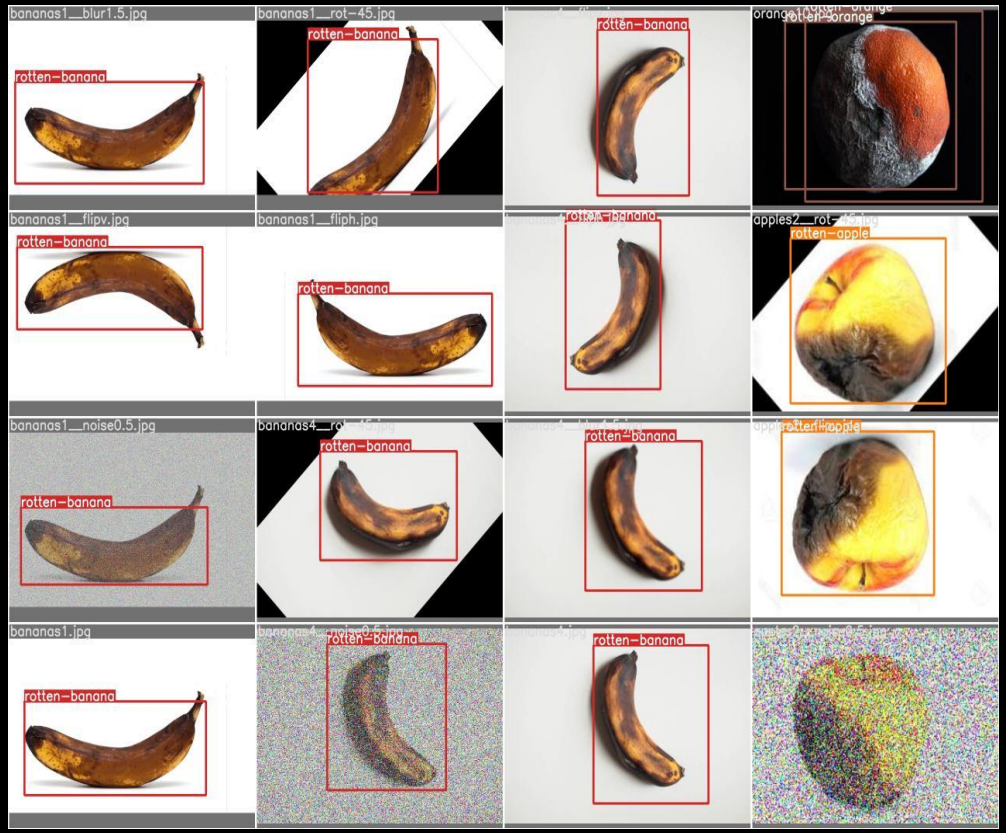
Model Training

Training Inputs

Approximately **500 images** per label, including augmented images

- 0: Good Apples
- 1: Rotten Apples
- 2: Good Bananas
- 3: Rotten Bananas
- 4: Good Oranges
- 5: Rotten Oranges

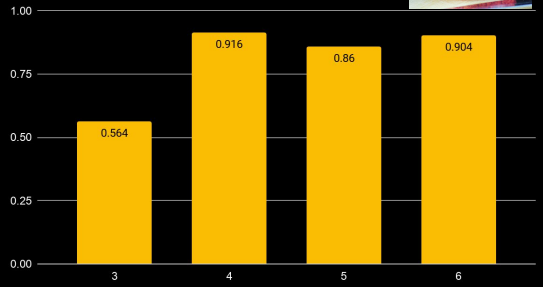
Model Performance on Validation Dataset



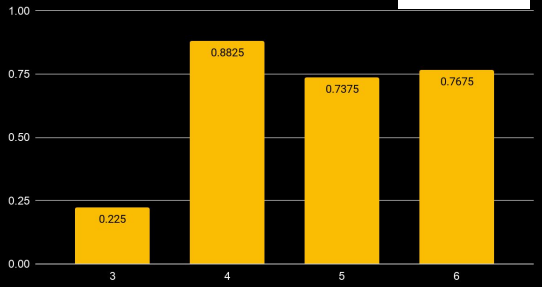
#	YoloV5 Model	mAP@0.5
1	L Batch-16, Epoch-5	0.504
2	M Batch-08, Epoch-05	0.518
3	X Batch-16, Epoch-05	0.548
4	X Batch-08, Epoch-20	0.697
5	X Batch-16, Epoch-20 (adding more images)	0.696
6	X Batch-16, Epoch-20	0.681

End-to-End Deployment Performance: Confidence Levels

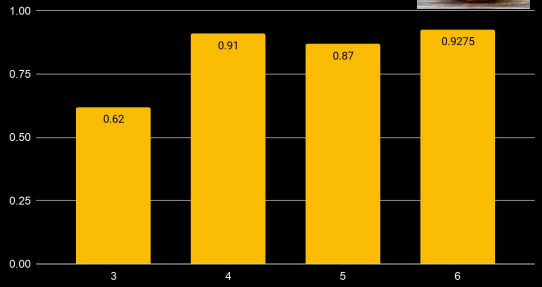
Good Red Apple



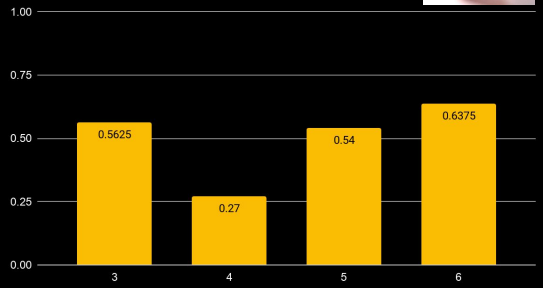
Good Banana



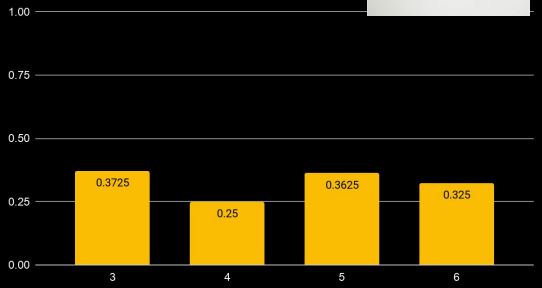
Good Orange



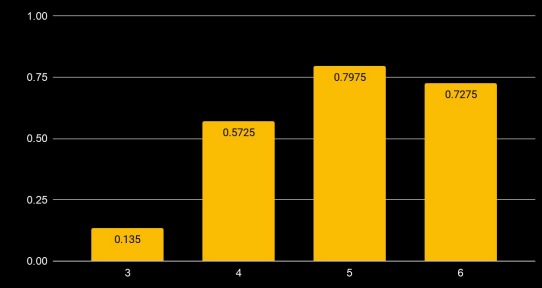
Rotten Red Apple



Rotten Banana

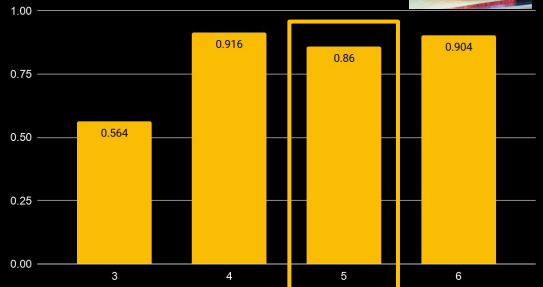


Rotten Orange

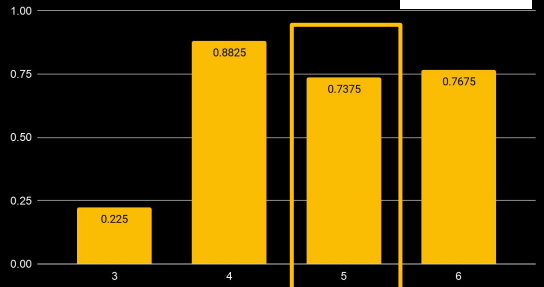


End-to-End Deployment Performance: Confidence Levels

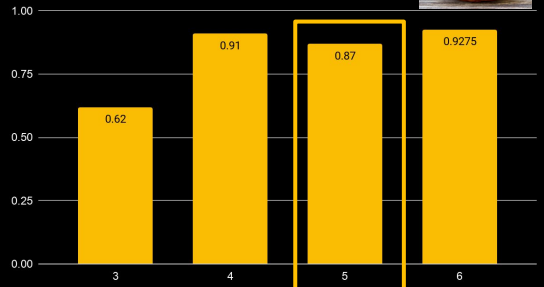
Good Red Apple



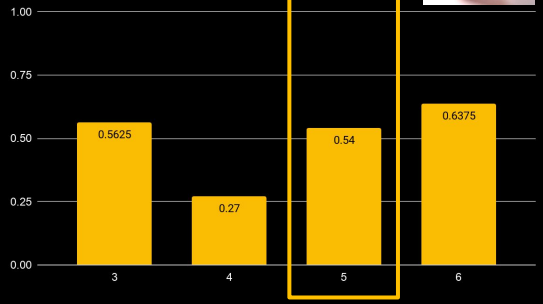
Good Banana



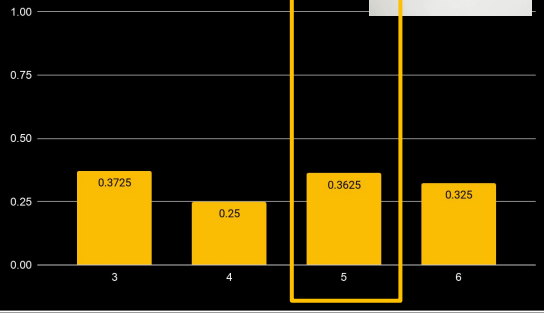
Good Orange



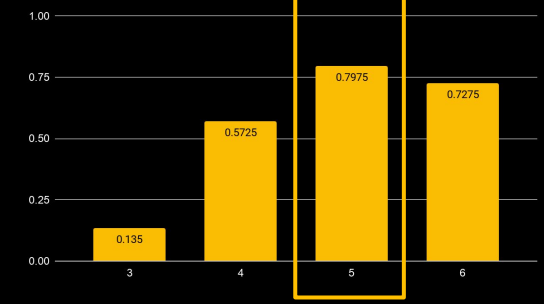
Rotten Red Apple



Rotten Banana



Rotten Orange



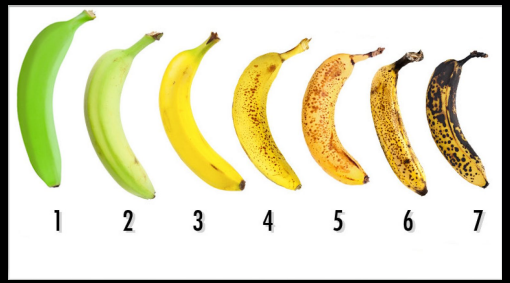
Summary

Takeaways and Learnings

- Transfer learning effective, but takes large model and many images to fix underpowered rotten fruit detection
    - Underpower comes from augmentation and transfer learning process
- Training constrained by memory and batch size bandwidth - informs AWS instance size
- Jetson GPIO pins and Raspberry Pi camera interface were extremely finicky!

Future Work

- Feedback loop of weight + metadata into retraining model
- Further hyperparameter tuning / freezing more/fewer layers
- More categories
- More complex scenes (cafeteria food waste, canned goods, meats)
    - binary classification -> multiple outputs



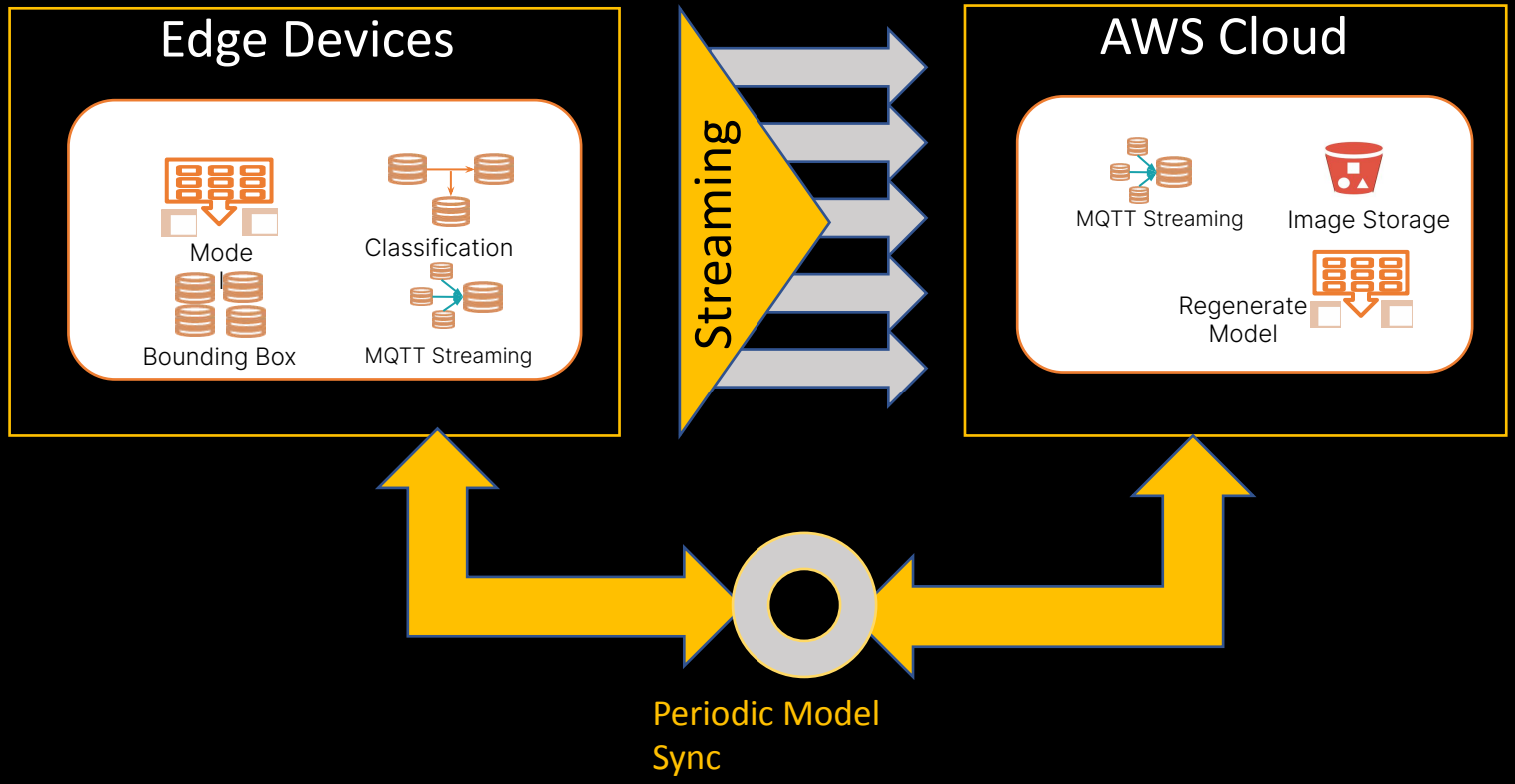
Thank You & Questions?

Repo: https://github.com/w251-final-project-fall2020/final_project

References

1. YOLOv5 Github: https://github.com/ultralytics/yolov5
    a. Transfer Learning: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data
2. Image Augmentation
    a. https://blog.paperspace.com/data-augmentation-for-bounding-boxes/
    b. https://github.com/codebox/image_augmentor
3. https://cocodataset.org/#home

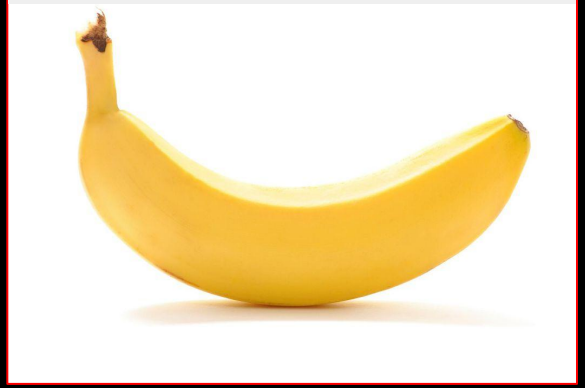
End-to-End Product Design



Assumptions and Limitations (TEAM)

0 : Good Apple

2 : Good Banana



4 : Good Orange

1 : Rotten Apple

3 : Rotten Banana

5 : Rotten Orange

Our Results at a Glance - Remove this slide?

- Ran multiple runs of yolov5x models in AWS varying batch and epoch
- Batch 16 and Epoch 20 showed better results of all