COMPUTER VISION PROJECT ON

Potato Tracking and Counting

Individual (Option-2)

Group 7

Mustafa Alaff Wasee - 301323608

GitHub Repository: https://github.com/GWasee/Final-Computer-Vision-Project

IAT481 D100 Spring 2024

Instructor: Dr. O. Nilay Yalcin

TA: Maryiam Zahoor

Outline your chosen problem and its significance. Explain the model you chose, which model type (CNN, LSTM etc.) is it, detail the specifications of the model.

The project targets a critical problem in agricultural quality control: the automated tracking and classification of potatoes. For this application, the model chosen is a convolutional neural network (CNN), specifically the latest version of You Only Look Once (YOLOv8). YOLO models are highly regarded for their speed and accuracy, making them suitable for applications requiring immediate feedback, such as sorting agricultural produce on a conveyor belt in real time. The specific variant used in this project is "yolov8s.pt," which is a smaller, more optimized version designed for efficient deployment while still leveraging the high performance of YOLOv8 architecture.

The YOLOv8 model is pre-trained on a comprehensive dataset that includes various categories of potatoes from Roboflow. It identifies and classifies potatoes into four distinct classes: healthy, sprouted, damaged, and defected, each reflecting the quality of the produce. The model operates by analyzing visual data, tracking objects (potatoes) in videos, and assigning them to the relevant class based on their visual characteristics. The implementation of such a model has the potential to significantly improve the efficiency of agricultural produce sorting while maintaining, if not improving, the current quality control standards.

The project idea was already implemented on Roboflow, for personal contribution I have used YOLO for tracking and counting the different types of potatoes simulating a conveyor belt feed from a processing facility. The project and algorithms used have been cited at the end of the report.

What kind of problem and use/case you're focusing on in your application, who are your users, where/how they'll be using the model, what kind of data it requires?

Problem Statement:
In agricultural industries, especially in developing countries like Bangladesh, sorting and grading potatoes by quality is often performed manually, which is labor-intensive and subject to human error. Automating this process using computer vision can increase efficiency, reduce costs, and enhance the consistency of output quality. This is not only an issue of enhancing productivity but also one of occupational safety, particularly in warmer climates like Bangladesh's. There, the intense heat presents a health risk to farmers performing manual sorting, thus a machine learning solution could

substantially reduce human exposure to harsh conditions while also streamlining the sorting process.

Use-Case:
The users of this application are facility managers and quality control technicians at food processing plants. These professionals are currently relying on manual or industrial inspection to sort potatoes, which can be tedious and error-prone. The application will be used within the controlled environment of these facilities, utilizing a video feed mounted above conveyor belts to capture real-time video footage.

The model requires a dataset consisting of annotated images of potatoes, categorized into different quality classes based on their appearance: normal, sprouted, damaged, and defected. These annotations are critical for the model to learn and accurately identify the categories post-training. Once integrated, the model will analyze the video feed, instantly identifying and classifying each potato on the production line, thereby optimizing the sorting process, ensuring consistent product quality, and safeguarding the well-being of the workforce.

User Profile:

- End-Users: Agricultural workers and facility managers in potato sorting facilities primarily in regions like Bangladesh.
- Environmental Conditions: Industrial settings with controlled lighting where potatoes are transported on conveyor belts.

Data Requirements:
Clean & stable video feed of potatoes from cameras set on a conveyor belt/processing plant. The model uses a dataset of annotated potato images to identify and classify the potatoes accurately. For this project, a video feed from YouTube to provide tracking and counting properties is used to simulate the video feed that replicates the conveyor belt environment in processing facilities.

The model was initially trained on a dataset from Roboflow, which consists of over 7000 labeled close-up images of potatoes in varying conditions. This dataset includes a comprehensive .yaml file that organizes the data into a training set with 7188 images, a validation set with 576, and a test set with 270. The images are augmented to enhance the robustness of the model, with augmentations including horizontal and vertical flips, rotations (both 90 degrees and slight tilts between -8° and +8°), blur (up to 1 pixel), and noise (affecting up to 1% of pixels). Bounding boxes were also adjusted to reflect these

augmentations. The training dataset comprised static, close-up labeled images of potatoes, while the use case demands dynamic processing in an industrial environment with variable lighting and backgrounds. To address this, the model required fine-tuning with video data reflective of the actual deployment scenario. However, this fine-tuning couldn't be fully realized due to constraints such as the extensive computational power required for the large dataset and the lack of dynamic, conveyor belt-specific footage for retraining. Consequently, these limitations hindered further refinement as explained in the challenges section, preventing an increase in the model's accuracy for the intended dynamic application.

For your application, how do you receive the data from the user? How will the data be cleaned after it was received from the user?

The application retrieves data from users via a custom function that downloads videos from YouTube using the pytube library. The function takes a YouTube video URL and an output path, then downloads the video in MP4 format to the specified path on the local storage, ready to be processed by the model.

For data cleaning, the following steps were taken to process the video for the application:

- Validation: To ensure the downloaded video format is compatible with the processing system (MP4 in this case).
- Frame Extraction: Extract frames from the video to be used as input for the model. This is handled by the cv2 library.
- Preprocessing: Apply necessary preprocessing steps on the extracted frames, such as resizing, normalization, or color correction to align with the input requirements of the model.
- Filtering: Remove any unusable frames, such as those that are blurred or do not contain relevant content.
- Augmentation: Depending on the application, some form of augmentation may be applied to increase the robustness of the model, resizing the video accelerated computation for this application. However, for live-feed video use cases, some form of augmentation may be applied to increase the robustness of the model, like adjusting brightness or contrast to simulate different environmental conditions.

If the users are employing the system in different environmental conditions, such as varying lighting or angles, the model may require additional fine-tuning to accommodate these variations. This could involve adjusting the preprocessing pipeline to include image correction techniques suited to each specific environment the model will encounter.

What are the challenges, ethical and bias issues your users will be facing? How did you address these issues in your application?

Challenges:

Computation Limitations:

- Problem: The large dataset required significant computational resources which were not always available. Training that large dataset, it was taking 3-4 hours on each epoch.
- Solution: Modified the training parameters to reduce computational demand Hyperparameter Changes:
    - epochs=5 → 1
        - Reducing the number of epochs decreased the total number of times the entire dataset was passed forward and backward through the neural network. This cut down the overall training time dramatically, from potentially 15-20 hours to just 3-4 hours. This also caused underfitting and provided a more practical solution for showcasing capabilities for further development
    - batch=8 → 30
        - Increasing the batch size allowed more data to be processed simultaneously during each iteration of an epoch. This not only speeds up the training process due to fewer updates needed to the model's weights but also helps in stabilizing the gradient estimates, potentially enhancing the learning process. This required a lot of ram, but the decrease in image size compensated for this, but compromised the overall clarity of detections.
    - image size=640 → 416
        - Reducing the input image size decreased the number of pixels the model had to process, lowering the computational load and speeding up training. This caused the model to generalize on tracking from the YouTube video as it didn't learn the fine details and kept tracking round objects. Overall best.pt had less spatial

information to learn from, which is usually a trade-off between speed and accuracy.
    - Learning rate=0.0001 → 0.001
        - This was to ensure that each epoch made sufficient progress, which updated model weights more significantly with each batch when the total number of epochs was limited to just one.

Partial Occlusion:

- Problem: Potatoes overlapping on the conveyor were misclassified.
- Solution: Improved spatial hierarchy in training to enhance partial object recognition.

Varying Potato Sizes:

- Problem: Difficulty in recognizing potatoes of various sizes and shapes.
- Solution: Augmented training data to effectively identify diverse potato dimensions.

Ethical Concerns:

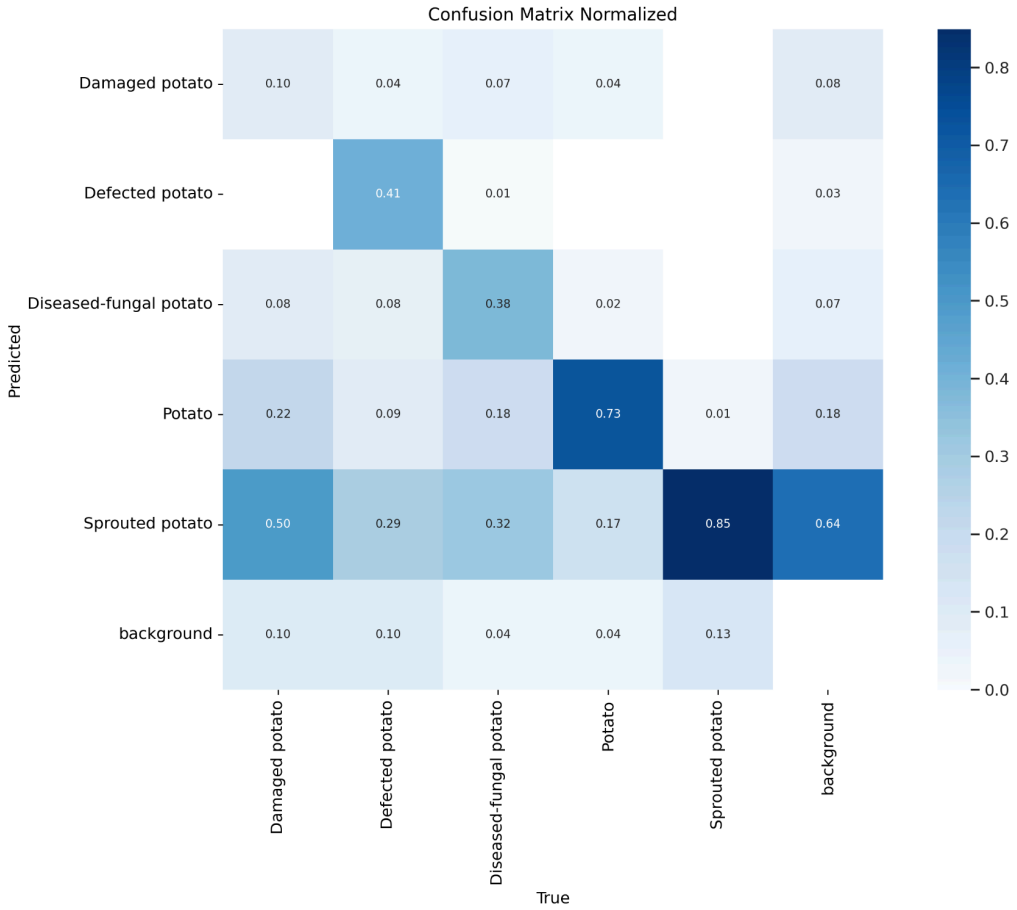
Addressing ethical concerns involves mitigating job impact by positioning the system as a supportive tool for workers, not a replacement, thus preserving jobs and enhancing safety. Transparency about the model's capabilities and limits is maintained to set realistic expectations and promote a safer, more efficient workplace.

Edge Cases:

- Extreme Lighting Conditions: While the model performs well under controlled lighting, extreme changes could affect performance. This issue can be addressed by integrating adaptive brightness and contrast adjustments in real time.
- Unusual Defects: Rare defects not represented in the training dataset may not be recognized correctly. Continuous model training with newly gathered data can gradually minimize this issue.

Testing and Result Analysis

Confusion Matrix:

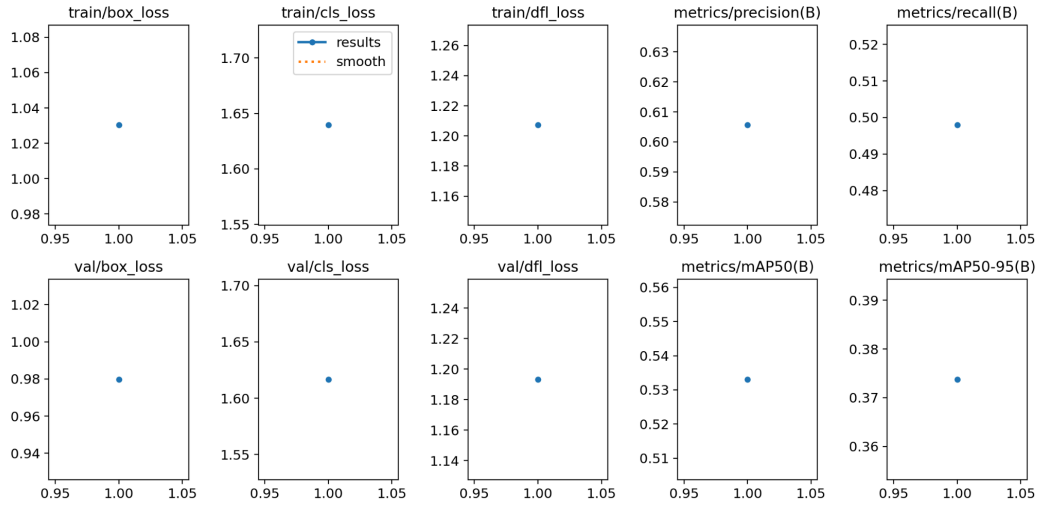


The normalized confusion matrix shows that ‘Potato’ and ‘Sprouted Potato’ categories have the highest true positive rates. However, the model had significant misclassifications between ‘damaged potato’ and ‘defected potato, suggesting the insufficient model training data differentiation.

- 'Potato' Class: The model is most accurate when identifying healthy potatoes (73% true positive rate), suggesting that it has effectively learned features of undamaged potatoes.
- 'Sprouted Potato' Class: The model also performs well in identifying sprouted potatoes, with an 85% true positive rate, indicating clear feature recognition for this category.
- Confusion Between Classes: There is notable confusion between 'Sprouted potato' and both 'Damaged potato' and 'Defected potato', possibly due to similar visual features or inadequate representation in the training data.

- Background Classification: The model has learned to distinguish potatoes from the background fairly well, with a 64% true positive rate for background classification, indicating good foreground-background separation abilities.

Results:



Since the model was trained on only 1 epoch, there are insufficient metrics

Training Losses:

- train/box_loss: 1.0303, indicating the average loss for predicting bounding box positions.
- train/cls_loss: 1.6395, reflecting the loss related to class predictions.
- train/dfl_loss: 1.2072, representing the directional feature loss.

Training Metrics:

- metrics/precision(B): Precision of 0.6056 suggests that when the model predicts an object, it is correct around 60.56% of the time.
- metrics/recall(B): Recall of 0.49793 indicates that the model detects nearly 49.793% of all relevant objects.
- metrics/mAP50(B): The mean average precision at IoU (Intersection over Union) threshold of 0.5 is 0.53308, which is moderate performance.
- metrics/mAP50-95(B): The mAP calculated over IoU thresholds from 0.5 to 0.95 is 0.37375, suggesting performance drops as the IoU threshold increases.

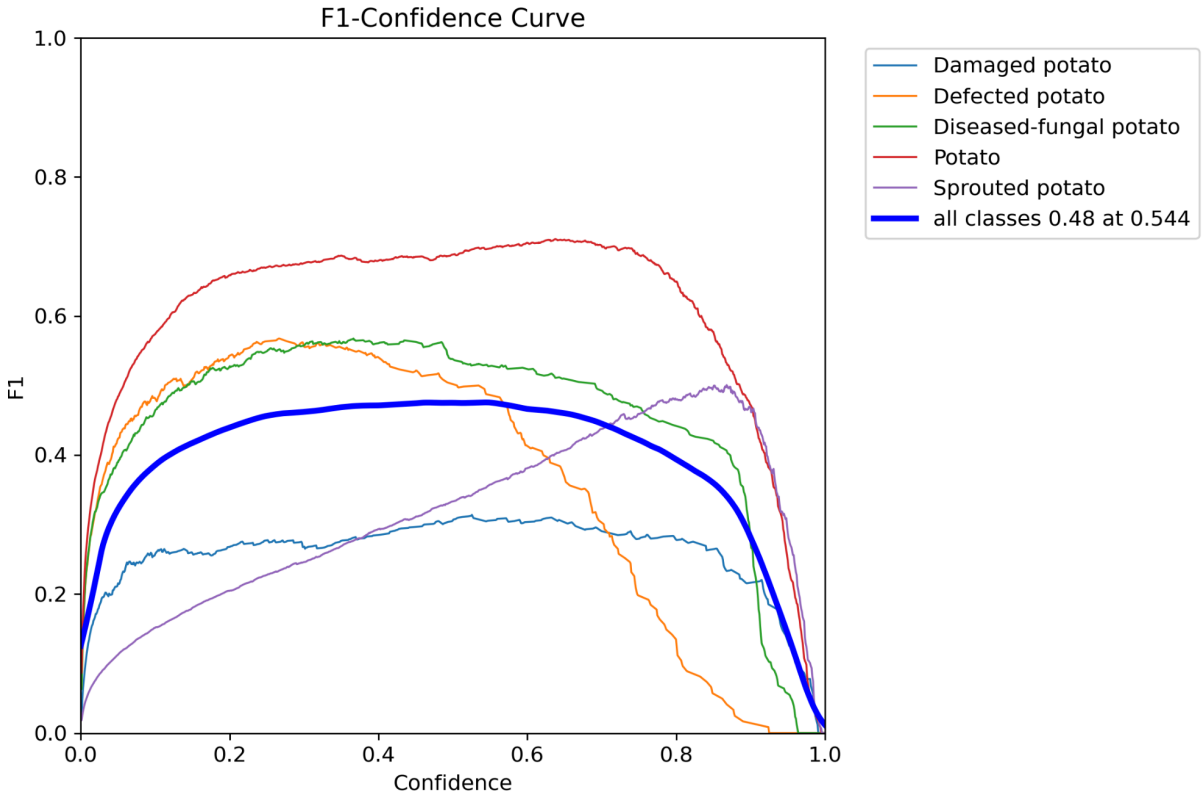
Validation Losses:

- val/box_loss: 0.97978, slightly less than training box loss.
- val/cls_loss: 1.6167, almost similar to training class loss.
- val/dfl_loss: 1.1933, comparable to the training directional feature loss.

Learning Rates:

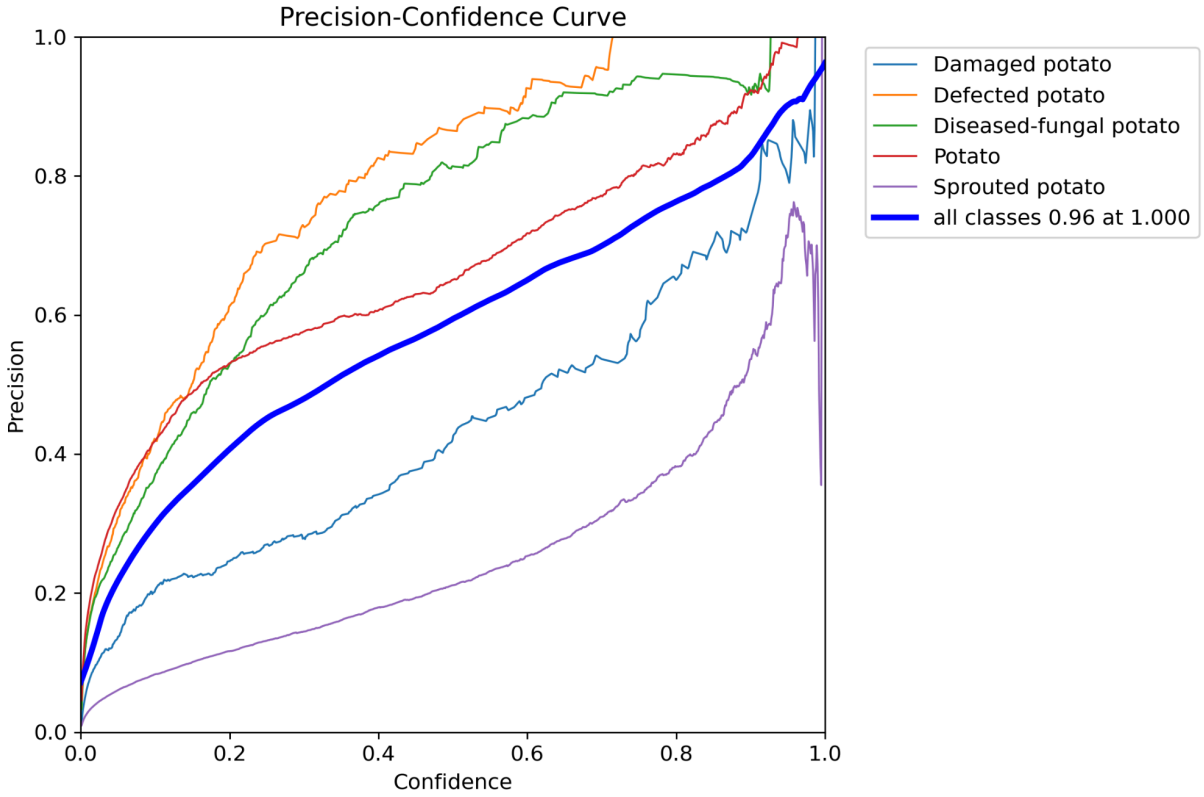
- lr/pg0, lr/pg1, lr/pg2: All have the same learning rate of 0.000369 for different parameter groups, indicating a uniform learning rate across the model's parameters.

F1 Curve:



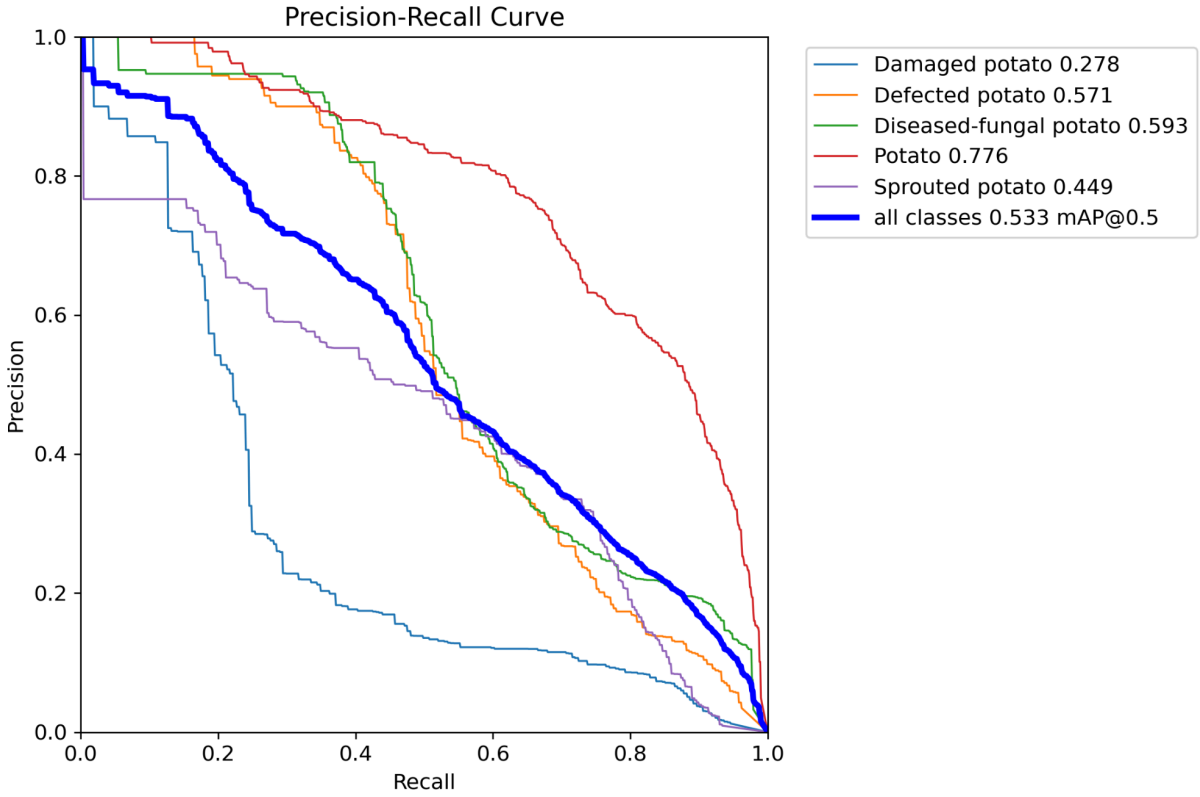
- ‘Damaged potato' and 'Defected potato' classes have higher F1 scores across most confidence levels, suggesting that the model is more adept at detecting and correctly classifying these conditions.
- The 'Potato' class, likely representing healthy potatoes, has a consistent F1 score for a range of confidence levels before it starts to drop, indicating reliable identification up to a point.
- The 'Sprouted potato' and 'Diseased-fungal potato' classes have lower peak F1 scores, suggesting that the model is less certain when classifying these categories or that they are harder to differentiate.
- The peak of the "all classes" curve indicates the best balance between precision and recall across all categories at a particular confidence threshold.
- The optimum point on the "all classes" curve marks where the model achieves its highest overall accuracy, here noted as F1=0.48 at a confidence of 0.544.

Precision-Confidence Curve:



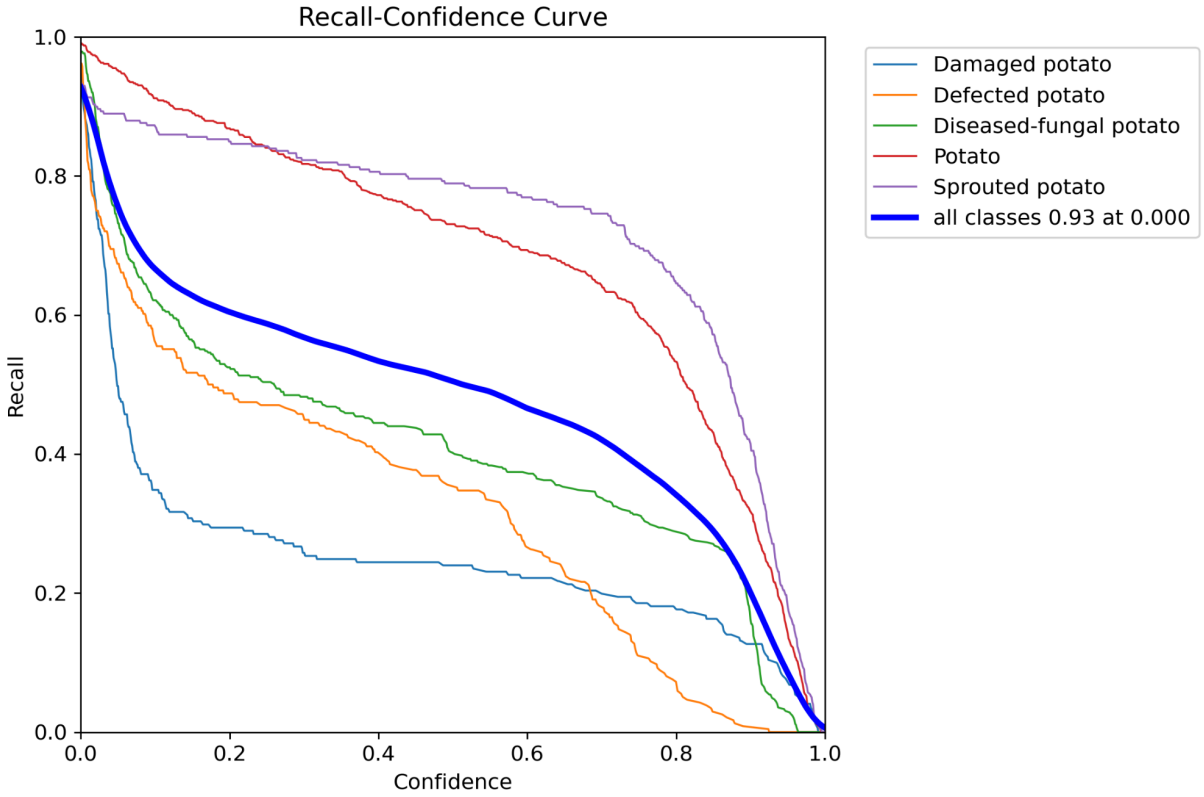
The 'Potato' class maintains high precision across confidence levels, while 'Sprouted potato' has lower precision, indicating more false positives. The curve for 'all classes' reaching high precision at maximum confidence suggests the model, when certain, is highly accurate. The peak precision of 0.96 at a confidence of 1.0 for all classes implies near-perfect precision at the highest confidence level, although it likely coincides with a lower recall.

Precision-Recall Curve:



Healthy 'Potatoes' are detected with high precision and recall, whereas 'Damaged potatoes' are less accurately identified. The mean Average Precision (mAP) at a 0.5 IoU threshold is 0.533 across all classes, indicating moderate overall performance. The curve demonstrates the model's strength in distinguishing 'Defected' and 'Diseased-fungal' potatoes, but also highlights challenges in accurately classifying 'Damaged' and 'Sprouted' potatoes, which could be due to the similar features and insufficient training data representation for these conditions.

Recall-Confidence Curve:



The 'Potato' class shows the highest recall at lower confidence thresholds, while 'Damaged potato' has the lowest, indicating a tendency to miss this condition. The curve for all classes combined reaches a recall of 0.93 at a confidence level near zero, suggesting that when the model is less stringent, it can detect nearly all positive cases, but this may include a higher number of false positives.

References:

https://universe.roboflow.com/vegetable-quality-detection/potato-detection-3et6q

Skalski, P. (2024). Track and count objects using yolov8. Roboflow Blog. https://blog.roboflow.com/yolov8-tracking-and-counting/

Chatgpt & Colab AI for troubleshooting. https://chat.openai.com

https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/how-to-track-and-count-vehicles-with-yolov8.ipynb#scrollTo=FMiFSEV9RIC-