Part 2 – Reasoning-Based Questions

Q1. Choosing the Right Approach

If the goal is to detect whether a product is missing its label, the most suitable approach would be **object detection** or **segmentation** rather than simple classification.
Since the label covers only a small part of the product, a classifier might fail because the overall product appearance is too similar.
Detection allows the model to localize and verify the presence or absence of the label region.
If detection proves unreliable, a **segmentation model** can be used as a fallback to precisely outline the label's area.
This approach not only confirms whether a label exists but also shows *where* it should be located.

Q2. Debugging a Poorly Performing Model

If the trained model performs poorly on new factory images, I would first check for a **domain shift** — changes in lighting, background, camera angle, or quality between training and test data.
I would visualize predictions alongside their ground-truth boxes to identify common failure patterns.
Next, I would inspect whether the dataset is balanced and whether any class dominates.
I would also confirm that my data augmentations simulate real factory conditions.
Finally, I would fine-tune the model using a small portion of new factory data to see whether the problem lies with data quality or model generalization.

Q3. Accuracy vs Real Risk

In this situation, **accuracy** is not the most meaningful metric.
Even with 98 % accuracy, missing one defective product out of ten could have serious operational risks.
Therefore, I would emphasize **Recall**, which measures how many defective items are correctly identified.
A higher recall minimizes false negatives — the most costly errors in this case.
I would also evaluate **Precision** and the **F1-score** to maintain a good balance between detecting all defects and avoiding false alarms.

Q4. Annotation Edge Cases

Blurry or partially visible images should not be discarded entirely, as they reflect real-world scenarios where images are imperfect.
Including some of these samples helps the model become more robust to motion blur, occlusion, and poor lighting.
However, if the image is too unclear to label accurately, it can introduce noise and should be excluded.
A balanced inclusion of such cases ensures the model generalizes better while maintaining data quality.
The key trade-off is between **robustness** and **label accuracy** — keeping enough challenging samples without overwhelming the dataset with poor-quality images.