Q1: Choosing the Right Approach

You are tasked with identifying whether a product is missing its label on an assembly line. The products are visually similar except for the label.

Q: Would you use classification, detection, or segmentation? Why? What would be your fallback if the first approach doesn't work?

I would use detection model, because the task is to check whether a label is present or missing on the product. Classification alone wouldn't be enough since it only says "label" or "no label" without identifying where the label should be. Segmentation could work too, but it is more complex than needed.If detection struggles, my fallback would be **classification on cropped regions** where the label is expected, to simplify the problem and reduce noise. This layered approach balances accuracy and efficiency on the assembly line.

Q2: Debugging a Poorly Performing Model

You trained a model on 1000 images, but it performs poorly on new images from the factory.

Q: Design a small experiment or checklist to debug the issue. What would you test or visualize?

First, I would check for data mismatch, size, color channel and other factors between the training images and the factory images, factors such as differences in lighting, angle, or resolution. Then I'd visualize some training and test predictions to see if the model is consistently failing in specific conditions. I would also review the labeling quality to confirm if the ground truth is correct and consistent and if the model can give generalise results. Finally, I would verify whether the dataset size (1000 images) is too small and consider collecting more diverse examples.

Q3: Accuracy vs Real Risk

Your model has 98% accuracy but still misses 1 out of 10 defective products.

Q: Is accuracy the right metric in this case? What would you look at instead and why?

Accuracy is not the right metric here because the real risk lies in missing defective products, even if overall accuracy looks high. Instead, I would focus on recall for the defective class, since missing a defect has a high cost. I'd also look at the confusion matrix to see how many defective items are misclassified as good ones. Choosing the right metric ensures the model aligns with the real-world risk.

Q4: Annotation Edge Cases

You're labeling data, but many images contain blurry or partially visible objects.

Q: Should these be kept in the dataset? Why or why not? What trade-offs are you considering?

Blurry or partially visible objects should usually be kept as it make model more generalised, but carefully labeled, because the model needs to learn to handle real-world imperfections. However, including too many poor-quality images could confuse the model and lower performance on clear cases. The trade-off is between robustness and clarity, keeping them improves generalization, while removing them improves clean-case accuracy. A balanced approach is to keep a subset of these cases so the model learns some tolerance but is not dominated by noisy data. This way, the dataset reflects real conditions without overwhelming the learning process.