Part 2: Reasoning-Based Questions (Write-up)

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?

Answer.

I would start with Image Classification because it's the most straightforward choice. My plan would be to just show the computer a bunch of pictures of products with labels and without them, so it learns to tell the difference and categorize any new picture as "labeled" or "unlabeled." This is much simpler than object detection, which would try to draw a box around a label that might not even be there. If that didn't work well, my backup plan would be Object Detection. I'd train a model to specifically find the "label," and if the program runs and can't find a label on a product, it would know that one is missing and flag it as an error.

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?

Answer.

To figure out why the model is failing, my first step would be a simple visual check. I'd create a checklist to compare the new factory images where the model made mistakes directly against my original training images. I would look for obvious differences in lighting, camera angles, and backgrounds, as the factory environment might be different from my training setup. I'd also check if the new images are blurry or if the products themselves look slightly different. Finally, I would try to visualize what the model is "looking at" in the bad images to see if it's focusing on the wrong thing, like a shadow instead of the actual product. This process would tell me if my original dataset doesn't match the real world, which would mean I need to add some of these new factory images to my training data and retrain the model.

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?

Answer.

No, the 98% accuracy score is misleading and dangerous in this case because the real risk is failing to catch a defective product. Accuracy can be very high if most products are good, as the model gets rewarded for every correct "good" prediction, which can hide a high failure rate on the rare "bad" products. Instead of accuracy, I would focus on the Recall metric. Recall specifically answers the most important question here: "Of all the defective products that actually existed, what percentage did

our model successfully find and flag?" In this scenario, the Recall is 90% (9 out of 10), which gives a much more honest picture of the model's performance and the true risk of a bad product getting past the system.

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?

Answer.

Yes, I would definitely keep those tricky images. If you only train the model on perfect, clear pictures, it will fail when it sees the messy images that happen in the real world. The trade-off is that it's harder to label these blurry or partial images, and your model's performance on *perfect* images might be slightly lower. However, in exchange, you get a much more robust and useful model that doesn't panic when an image isn't perfectly clear.