Q1: Choosing the Right Approach

I would use classification since the products are identical except for labels - it's a simple binary problem of "has label" vs "no label." Classification is faster and needs less data than detection or segmentation. If that fails because labels appear in different positions, I'd switch to object detection to actually locate where the label should be on the product. This method i have practically experienced myself in realtime.

Q2: Debugging a Poorly Performing Model

First, I will compare my training images to the actual factory images - different lighting, angles, or backgrounds could be the issue. Then I would look at what the model is getting wrong to spot patterns in the failures. I will also check if my 1000 training images actually represent the real factory conditions, and make sure my labels are consistent and accurate.

Q3: Accuracy vs Real Risk

No, accuracy is not right here because missing defective products is way worse than flagging good ones as bad. I'd focus on recall instead to catch more defects, even if it means more false alarms. The business cost of shipping bad products to customers is usually much higher than stopping the line to double-check a good product.

Q4: Annotation Edge Cases

Yes, keep them but be smart about it. Blurry and partial images help the model handle real-world messiness, but I'd set clear rules like "only label if 30% visible" to stay consistent. The tradeoff is between making the model robust versus keeping the training data clean and easy to learn from.

Note : I HAVE DONE THIS SAME FOR MY PRESENT PROJECT AND HAVE EVIDENCE. TO PROVE THIS FURTHER.