Q1: Choosing the Right Approach

I would use object detection to identify whether a product is missing its label. Detection helps locate and classify specific regions in the image, which is ideal since we're checking if a particular area (the label) exists or not. A simple classification model might not work well here because the overall product looks the same only the label area changes. If detection doesn't perform well, I would try image segmentation, which can more precisely identify pixel level differences in the label area. As a fallback, I could also use a template-matching or image-difference approach to check for missing labels without deep learning.

Q2: Debugging a Poorly Performing Model

First, I would check the lighting, angle, or background of new factory images and whether they differ from the images in training set. Next, I'd check labelling quality in the dataset to make sure the model is learning from correct information. I'd also check whether the bounding boxes are properly drawn around the objects. I'd also look at confidence scores to see if the model is unsure or completely wrong. Finally, I'd test data augmentation or fine-tuning the model with a few factory images to help it adapt to real world conditions.

Q3: Accuracy vs Real Risk

No, accuracy isn't the right metric in this case. Even with 98% accuracy, missing 1 out of 10 defective products could cause serious issues in production. Instead, I'd look at recall (especially for defective products), since it measures how well the model catches all the true positives. Precision is also important if false alarms slow down production, but recall should take priority because missing a defective product is a higher risk. The right balance depends on the cost of each type of error.

Q4: Annotation Edge Cases

I would keep some of the blurry or partially visible images, but only if those kind of images will be encountered by the model in actual production. Including such data helps make the model more resilient to imperfect conditions. However, if the blur is too strong or the object is barely visible, it might add noise and confuse training. The trade off here is between realism and data quality keeping challenging but relevant images can improve generalisation, while too many unclear samples might reduce accuracy.