Chayan Khetan
chayankhetan1@gmail.com
Phone : 8448034652
Github : https://github.com/Chayan-03/Vision_Pipeline

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.

Answer :
I would use object detection because the problem is not just classifying the whole product, but specifically identifying whether the label is present in the correct spot. Detection will allow the model to localize the label region and confirm its presence. If detection fails to capture small variations or subtle differences, I would fall back on segmentation, since it can provide a pixel-level understanding of whether the label area is filled or empty. As a last fallback, a simple classification model could be trained just on cropped regions of the label area, though it would be less robust. This layered approach balances precision with practicality in an industrial setup.

Q2: Debugging a Poorly Performing Model

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

Answer :
The first thing I'd check is whether the training dataset truly represents the factory conditions—things like lighting, viewing angles, and product variations. If not, the model may be facing a domain shift, so I'd test it on a small hold-out set of new factory images and visualize where it fails. I'd also carefully review annotations to make sure labels are consistent and bounding boxes are accurate, since label noise can reduce performance. Next, I'd compare training and validation losses to see if the model is overfitting, and, if needed, add data augmentation to improve generalization. Finally, if the issue persists, I'd fine-tune the model with a small number of newly annotated factory images to see if the performance gap can be closed.

Q3: Accuracy vs Real Risk

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

Answer :

Accuracy is not the right metric here because it hides the real risk of missing defective products. In a factory setting, even a single missed defective product can have big consequences, so false negatives matter much more than overall accuracy. Instead, I would focus on recall (or sensitivity) for the defective class, since it measures how many defective items the model actually catches. I'd also look at the confusion matrix and track metrics like precision vs. recall trade-offs to balance catching defects with minimizing false alarms. This way, the evaluation aligns with the business goal of ensuring no defective product slips through, even if it slightly reduces overall accuracy.

Q4: Annotation Edge Cases

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

Answer:

Blurry or partially visible objects should not be removed entirely, because they reflect real-world conditions that the model may encounter on the assembly line. However, they should be annotated carefully and consistently, since poor or inconsistent labels can confuse the model. The trade-off is between data realism (keeping difficult cases so the model generalizes) and data clarity (removing extreme noise that adds no learning value). If the blur is so severe that even a human cannot recognize the object, it may be better to exclude those samples. On the other hand, partial visibility is valuable to include, since products on a line are often occluded or tilted. Balancing these cases ensures the model learns robustness without being trained on meaningless noise.