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 use classification to identify if the product is missing its label on the line of assembly. A classification model can take a product image and decide one of two labels, "label present" or "label missing" in this binary decision. This approach finds fit because the products are all very similar with respect to visual appearance, and the presence of the label is the only thing that will distinguish the image. Hence, much less complexity is required compared to detection or segmentation methods.

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 debug the poorly performing model trained on 1000 images, I would design a simple experimental approach to nail down where the problem lies. First, I would look at the curves of training vs. validation loss to check for underfitting or overfitting. Then, I would look into a sample of the training images compared with new factory images. I would also check the label distribution of the dataset to confirm whether the products labeled and unlabeled are balanced or not, as any imbalance on this front could bias prediction. I would then continue testing the trained model on the new images, identifying particular errors by means of confusion matrices. Data augmentation and fine-tuning could be further experimented with if needed to resolve the issues and to enhance generalization.

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: The accuracy in this case is not the correct metric, since missing one out of every ten actual defective items signifies a false negative rate that is considered prejudicial in the hands of quality controllers, i.e., those for whom defect detection is of utmost importance. On the other hand, I would focus on recall, the measure of how many true defective products have been detected, and hence directly addresses the issue of undetectable defects. I would also consider the F1-score to weigh the balance between precision and recall so that the model does not admit too many false negatives and false positives. The confusion matrix will allow the quantification of some errors, whereas the ROC-AUC could have been considered for assessing the model's capability to differentiate defective from non-defective products at different thresholds. These metrics align more to the real risk of defective products reaching a customer.

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: Before they are keeping into the dataset, objects that are blurry or partially visible should be given critical consideration. I would keep those images if the operative condition they depict is truly what happens along the assembly line, as these could reinforce the model's generality for similar, marginal cases in production. It, however, should be noted that very many such images may thereby add noise to the dataset and diminish the model's performance on clear images. Hence, I would weigh the trade-offs between realism and quality of the dataset. Some blurry images will allow the model to generalize better to imperfect situations, but too many could become a hindrance to accuracy. Thus, I would include some guidelines on how to annotate these images and limit their share within the dataset. The other option would be to either preprocess these images or augment the clear images with simulated blur so as to enrich the dataset without compromising on quality for the real edge cases.