QUESTION and ANSWERS

Q1- **Choosing the Right Approach:**

I would use **object detection** to identify whether a product is missing its label, because the products are visually similar and the key difference is the **presence or absence of a localized label**. Detection allows the model to focus on the specific region where the label should appear, rather than relying on global image features. Classification would struggle if the product occupies different positions or if multiple products appear in one frame. Segmentation is not necessary since pixel-level accuracy is not required for a yes/no decision.
If detection does not perform well, my fallback would be a **two-stage pipeline**: first crop the expected label area using heuristics or template alignment, then apply **binary classification** to determine label presence.

Q2: Debugging a Poorly Performing Model

First, I would compare training vs validation metrics (loss, mAP, precision, recall) to check for overfitting. Then, I would visually inspect prediction outputs on factory images to see whether failures are due to lighting, motion blur, camera angle, or glove type variation. I would also validate annotation quality in Roboflow, ensuring bounding boxes are tight and labels are consistent. Another test would be running inference on a small subset of training images to confirm the model can at least memorize seen data. Finally, I would retrain with augmented data (blur, brightness, rotation) that better matches the factory environment.

Q3: Accuracy vs Real Risk

Accuracy alone is not the right metric in this case because missing a defective (no-glove) product has a higher real-world cost than a false alarm. In glove detection, false negatives are more dangerous than false positives. I would focus more on recall for the “no-glove” class, ensuring defective cases are rarely missed. Metrics like **precision, recall, F1-score, and confusion matrix** provide better insight into real-world performance. In safety-critical systems, maximizing recall is often more important than overall accuracy.

Q4: Annotation Edge Cases

Blurry or partially visible objects should generally be **kept in the dataset** because they represent real-world conditions encountered on the assembly line. Excluding them would make the model perform well only on clean, ideal images. However, these samples must be **labeled consistently** to avoid confusing the model. The trade-off is between **realism and**

noise: including them improves robustness, but too many low-quality samples can slow training or reduce precision. A balanced and well-curated dataset leads to better generalization.