

PhotoGAUGE ML/CV Engineer Interview Projects

Thank you for interviewing with PhotoGAUGE. We would like you to work on these **2 projects** and return your solutions by the date indicated in the email.. Solutions should be compiled in the form of a technical report and sent to us as a PDF.

You may use any software available to you that you are **legally** allowed to use or reuse any code you can find online or anywhere else. However, please cite all software used and other sources wherever necessary and follow all other established rules of scientific writing.

Feel free to ping us anytime at careers@photogauge.com with “Application Engineer: Projects” as the Subject, if you have any questions. Please note that these questions and data belong to PhotoGAUGE and should not be shared with anyone else. The purpose of this interview is not just to seek your solutions, but also to understand your technical knowledge, problem-solving skills, initiative, professionalism and communication skills. And some of the questions may appear incomplete or missing information needed to answer them, and we expect you to contact us with your questions.

Project #1: Build a tag inspection system for a fulfillment center

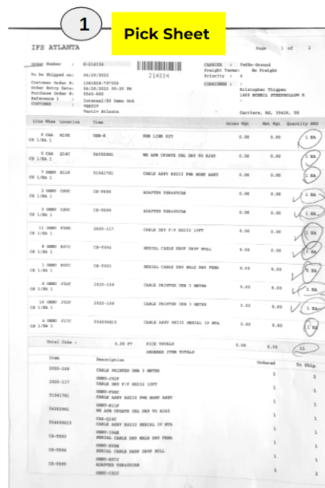
Background: A fulfillment center acts as a pseudo-warehouse for many OEMs and ships parts based on purchase orders (PO) from its customers. One such fulfillment center has approached PhotoGAUGE for building an optical verification system for their outbound parts, i.e. if the parts being shipped are as per the PO or not. Currently, they carry out this task manually as shown below.

- A PO is selected from their master daily queue (in a FIFO manner) and its corresponding pick-sheet is created—the pick-sheet contains all parts that are

required for this specific PO along with part numbers and quantities.

- Pickers collect all the parts as per the pick-sheet and place them on a verification table
- A person from their outbound quality verifies all the picked parts against the pick-sheet by reading tags on each packet. If all parts are as per the sheet, the shipment is packed and sent out or else, the shipment is held until the discrepancies are sorted out.

Current process



2
Pickers collect the items per the sheet, place them on the verification table



3 Manual verification against the pick sheet



4 Dispatch



Problem statement

The customer wants to replace the manual verification – your task is to build a camera-based system that can be employed for the final verification.

Deliverables

1. Submit a working code that takes an image of multiple parts lying on a verification table, each part with a tag on it, reads the information on it and compiles it to a csv file. You will find all the relevant data [here](#). This link contains photos and videos of various tags from the customer
2. Use the csv file from above and propose a way to compare it with the pick sheet.
3. How will you handle false negatives, i.e. tags that were missed?

Project #2: Build an optical system for measuring length of pipes

Background: One of PhotoGAUGE's customers is a major supplier of pipes to the oil and gas industries. Individual joints, which are 40'-50' long, are threaded together to form a continuous pipe that is inserted into a drill hole. Before being delivered to a customer, these joints are stored in an intermediate station. When a customer requests a certain length of pipe, the joints are brought out from storage and laid out in a single layer on stands (see aerial shot below), measured manually using a tape measure or a laser tool. A group may contain anywhere from 20 to 300 pipes.



Problem statement

Your task is to develop an optical system for measuring the length of each pipe in a group. PhotoGAUGE uses drones to photograph each group and you can find 3 such shoots [here](#)

Deliverables

1. Submit a working code that takes a frame of pipes as an input and outputs the end points of each pipe in a robust manner.
2. As can be seen from the drone shoots, there is significant overlap between two consecutive frames, which may lead to duplicate endpoints for a set of pipes in the two frames. How will you handle this and ensure that each pipe has only two endpoints?