Machine Learning Engineer Test:
Computer Vision and Object Detection

Objective

The goal of this test is to assess your skills in computer vision and object detection, with a specific focus on detecting room walls and identifying rooms in architectural blueprints or pre-construction plans.

This test is designed to evaluate your practical skills in applying advanced computer vision techniques to a specialized domain and your ability to integrate machine learning models into a simple API server for real-world applications. Good luck!

Duration

2 hour ~ 3 hours

Tasks

1) Room Wall Detection (required)

You will be provided with a dataset containing images of architectural blueprints or pre-construction plans. Develop a model or algorithm to detect the walls within these blueprints and annotate the detected walls in the images, clearly marking their boundaries.

2) Room Detection (optional)

Using the detected walls from the previous task, identify distinct rooms within the blueprints, annotating each room, ensuring that the boundaries are clearly defined and each room is distinctly marked.

3) API Server for Inference (required)

Develop a basic API server that can run inference using your model. The API should accept an image as input and return the image with annotated walls and identified rooms.

Recommended Libraries and Tools

PyTorch	Useful for developing machine learning models.
OpenCV	Essential for image processing tasks.
Scikit-image	For image processing and computer vision tasks.
Scikit-learn	For implementing machine learning algorithms.
Matplotlib/Seaborn	For visualizing the results.
Pandas/Numpy	For data manipulation and numerical operations.
Flask/FastAPI	For creating the API server to run inference.

Project Submission

Fork the provided base repo with test data and example requests, develop your project, and provide access to the following GitHub users:

- vhaine-tb
- gabrielreis-tb

Sample project to Fork

https://github.com/TrueBuiltSoftware/ml-eng-test/blob/main/README.md

Format of Submission

- Include a README file in your repository with detailed instructions on how to set up and run the API server.
- Use docker to containerize your application, ensuring consistency across different environments.
- Make sure to add the trained model and any additional files/libraries required to run the code.
- Provide a sample CURL/fetch/script to test the API inference
- Ensure your code is well-commented and documented to explain your approach and methodology.

Evaluation Criteria

- **Accuracy of Wall Detection:** The effectiveness of your algorithm in accurately detecting walls within the blueprints.
- **Room Detection Performance:** The ability to correctly identify and annotate rooms based on the detected walls.
- **API Server Functionality:** The API server should be functional, easy to set up, and capable of performing inference as expected.
- **Code Quality:** Clarity, organization, documentation, and adherence to best practices in coding and machine learning.
- **Result Presentation:** Clear visualization and annotation of detected walls and, optionally, rooms, along with a concise report of your findings.