Real-Part-Detection

This report outlines the key steps we will undertake in this project.

Data Gathering

To address our initial challenge of limited dataset availability, we will first gather a dataset comprising real-world plant components through our existing resources. This dataset will primarily consist of manually captured images of each component in the real-world setting, as opposed to schematic representations.

Next, we will proceed with the data labeling process, a critical step in training our neural network. The objective is to provide a comprehensive description of each component within the images. Additionally, we will demonstrate each component by drawing bounding rectangles around them to precisely determine their location. It's important to note that in the future, we aim to refine this process to identify the exact shape of objects rather than relying solely on rectangles. Furthermore, we will include supplementary information for each element, aiding in its subsequent identification on the P&ID. These features will be instrumental in distinguishing between components within the same category, such as differentiating between two PDCVs based on their distinct functions (e.g., cooling gas vs. process gas).

The annotation process will involve the following steps:

- Utilizing annotation software tools or platforms designed for image labeling, including options like LabelImg, RectLabel, or open-source alternatives such as VGG Image Annotator (VIA).
- Establishing a list of classes or labels for the components under consideration, such as "PDCV Valve," "Control Panel," or "Piping."
- Manually creating bounding boxes around the target components within each image and associating each box with the appropriate class label, providing clear identification of the component.
- Ensuring thorough verification and validation of the annotations to maintain accuracy and data integrity.
- Categorizing the labeled dataset into training, validation, and testing subsets, each serving a specific purpose in model development.
- Employing data augmentation techniques to enhance the diversity of the training dataset, including random rotations, flips, and brightness adjustments.
- we will organize the labeled data in a format that aligns with the requirements of our chosen machine learning framework, such as TensorFlow or PyTorch. Common formats, including TFRecord, COCO format, or structured folder organization by class, will be considered for efficient data utilization.

Training our Network

Moving forward, we will focus on training a convolutional neural network (CNN) to recognize the components within the images. Given the initial constraint of limited dataset size, we will leverage the transfer learning capability of CNNs. This approach involves fine-tuning a pre-trained CNN by retaining its early layers and retraining the last layer (or a few last layers) to adapt to our specific recognition task.

The training process will encompass these steps:

- Selection of a pre-trained CNN model, such as VGG, ResNet, or Inception, known for their ability to capture general image features.
- Modification of the pre-trained model by replacing its output layer to match the number of classes or components we aim to recognize.
- Implementation of transfer learning, wherein the modified model will be retrained using our collected dataset, even if it is relatively small. Data augmentation techniques will be applied to generate additional training samples.
- Fine-tuning of the model parameters, primarily updating the weights in the convolutional layers while keeping the earlier layers (learned from the pre-trained model) relatively stable.
- Ongoing evaluation of the model's performance using validation and testing datasets, with refinements as needed.

Final Step: Matching with P&Id

As we proceed, our final challenge will be to precisely locate components on the P&ID. We will initially explore a simple model that utilizes text matching to the P&ID content. This step will involve the following process:

- Extraction of text from the P&ID diagrams using tools like pytesseract or Kraken, which excel in text identification within images.
- Querying our integrated P&ID database to retrieve text data related to components, including labels, names, or descriptions.
- Implementation of a text-matching algorithm that compares the extracted text from the P&ID diagrams with the text data from the database, employing techniques such as fuzzy string matching or exact string matching.
- Establishment of a confidence threshold to validate text matches, ensuring a high degree of similarity between the extracted text and the database information.
- Integration of spatial information, when necessary, to pinpoint the location of the corresponding component on the P&ID diagram.
- Rigorous verification and testing of the text matching process using diverse P&ID diagrams with varying text attributes.

In conclusion, please note that this project is in its initial stages, and adjustments to the described processes may occur as we progress and gain insights.