Conceptual Design Report

McDefect Solutions

Model Selection

The solution that we plan to offer involves a 2-fold process. Firstly, we need to detect and segment out the components or parts from an image (or real time streaming video) for which defects are to be analysed. We plan to achieve this by training an Object Detection model on our custom dataset of component images, which we procure from the companies that will use our solution.

Next, we need to either detect defects or classify them into certain categories. To address this requirement, we plan to use a pre-trained Deep Learning model, and fine-tune it using transfer learning based on the defects dataset specific to the requirements of every company that we partner with.

Component / Part Detection

For detecting and segmenting out relevant components (by creating bounding boxes) from in image (or video), we choose to fine-tune the **RetinaNet** (with **ResNet50** Backbone) architecture on our custom dataset(s). This will be used with our OpenCV pipeline used to preprocess and ingest images (or relevant frames from a video) into the defect detection/classification model.

This is a transfer learning based approach, where we instantiate & fine-tune the RetinaNet for catering to the requirements of every different industry that we plan to collaborate with, based on the component images that we obtain from the industries. For the demo, we intend to fine-tune the model for detecting pump impellers in an image.

Ref: *Focal Loss for Dense Object Detection*

Defect Detection / Classification

For the core task of defect detection/classification, we select the **VGG16** architecture (since it was the better performing model) as our base model for the transfer learning process. Based on the requirements of the industries, fully-connected layer(s) can be added to the top of this network with the necessary number of classes to be predicted. Now, this model can be fine-tuned on the industry-specific component image dataset.

We assume that the prediction classes are as follows:

- **Defect Detection:** 2-class classification *[Defective / OK]*
- **Defect Classification:** Multi-class classification *[Types of defects]*

Ref: *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION*

Dataset Selection

To test our concept and prove the effectiveness of transfer learning in such industry-specific applications, we plan to use the following datasets:

Defect Detection

Casting Product Image Data [from Kaggle]

Defect Classification

Metal surface defects dataset [from Kaggle]

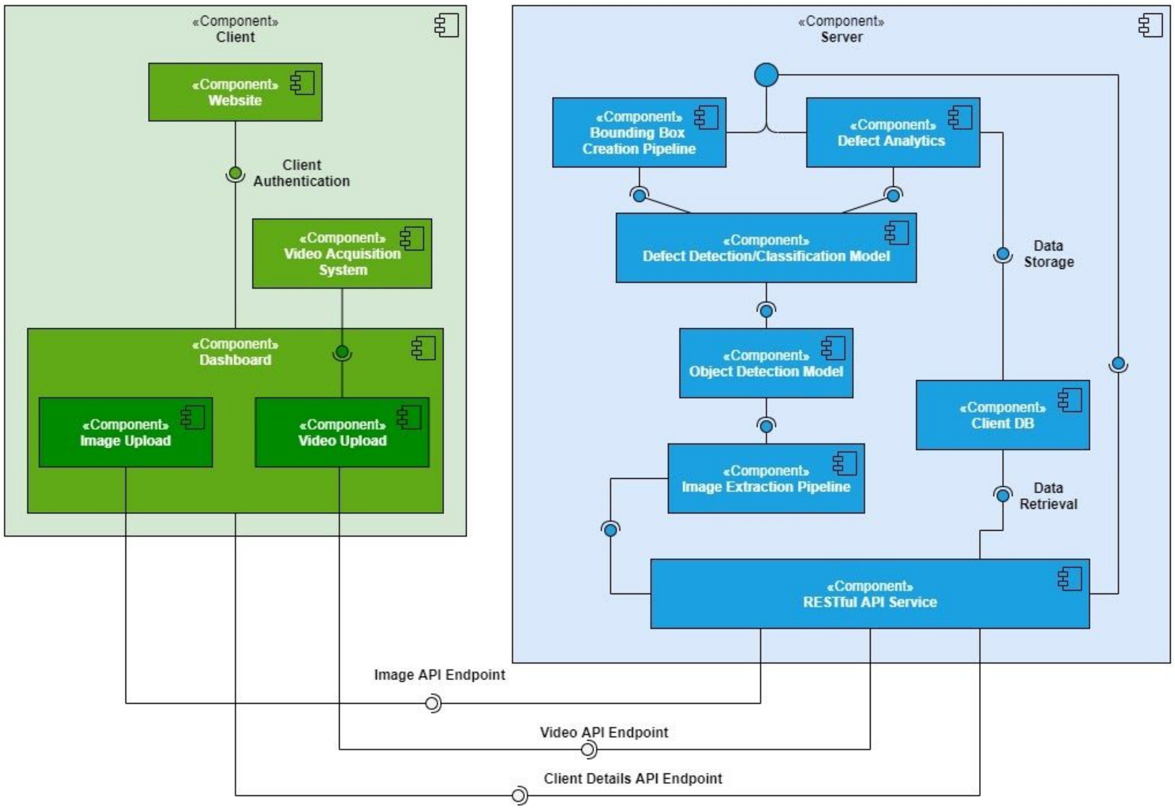
Note: *The same datasets would be used to train the YOLO object detection models to recognize and segment the relevant components form an image (or video) of the assembly line.*

Technology Choices

Operating System	Ubuntu 18.04 & Windows 10
AI Algorithm(s)	CNN-based Deep Neural Networks
Python DL & CV Framework(s)	Keras, OpenCV
Major Python Libraries	Numpy, Pillow, Unittest, Pydoc
Frontend & Website	HTML5, CSS3, JavaScript
Backend	FastAPI (Python)
Integration & Deployment	Docker, Localhost, Github Pages

High Level UML Diagrams

Component Diagram



Note: *Owing to the time & resource constraints posed as a result of this being a 10-week long course project, we plan to only implement the most crucial components for this project, that are sufficient to present an MVP. Following are the components that we do NOT plan to implement as a part of the Course Project, unless we find additional time:*

- *Client Authentication (we plan to have dummy authentication for the demo)*
- *Client DB*
- *Video Acquisition System (because it is industry-specific & hardware intensive)*
- *Defect Analytics (a very basic version will be implemented as a part of the project)*

Team

Tezan Sahu	170100035
Suchit Sharma	170040041
Chinmay VG	193109018
Chinmay Naik	203190016
Ayush Pandey	193300007