



**GE:AR Architecture &
Scope Document**
Version 0.1



Prepared By
Nishanth Saka
Project Manager
[x]cubeLabs

1. Introduction

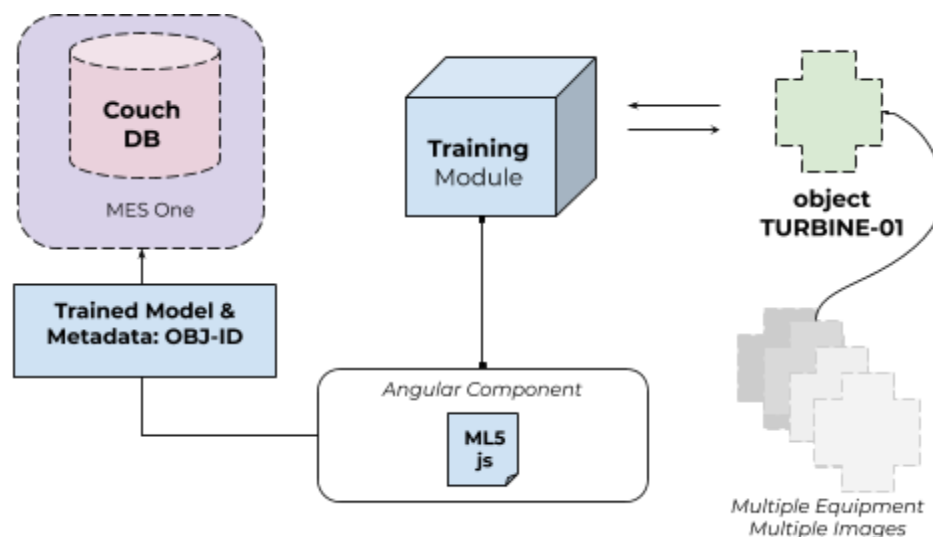
The GE:AR project has the following components...

- Training Module
- Learning Transfer - Trained Model on Cloud (CouchDB)
- Detection Module
- Integration Module

Details regarding each of these modules are given in further sections.

2. Training Model & Learning Transfer:

Below is the architecture that is followed during implementation of Training Model & Learning Transfer processes...



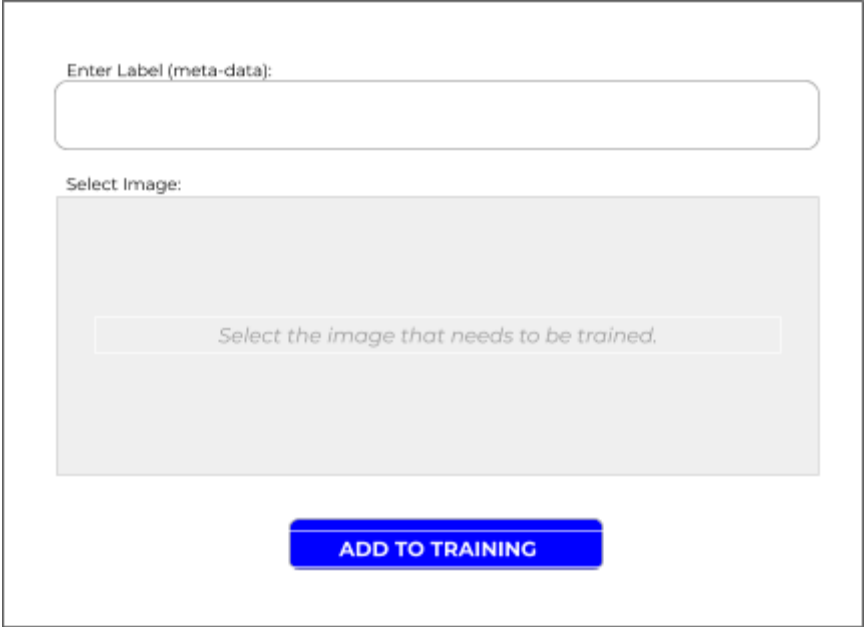
Training Module Architecture

2.1 Object

The object can be any of the equipment that needs to be used for training. For each object, a set of 4 images, in different angles, will be provided to model during training process. The idea is to train for a maximum of 4 such equipment.

2.2 Label (Meta Data)

While training, each image (for a specific equipment) shall be labelled with metadata, preferably 'ObjID'. This trained model shall associated the focal-points of the image with the corresponding metadata.



Enter Label (meta-data):

Select Image:

Select the image that needs to be trained.

ADD TO TRAINING

2.3 Training

Once all the images (for corresponding equipment) are added for training, the ML Model Training process can be started.

2.4 Test

Once the Training process is completed, the ML Model can be tested for accuracy - by checking with reference image and review if appropriate metadata is being displayed.

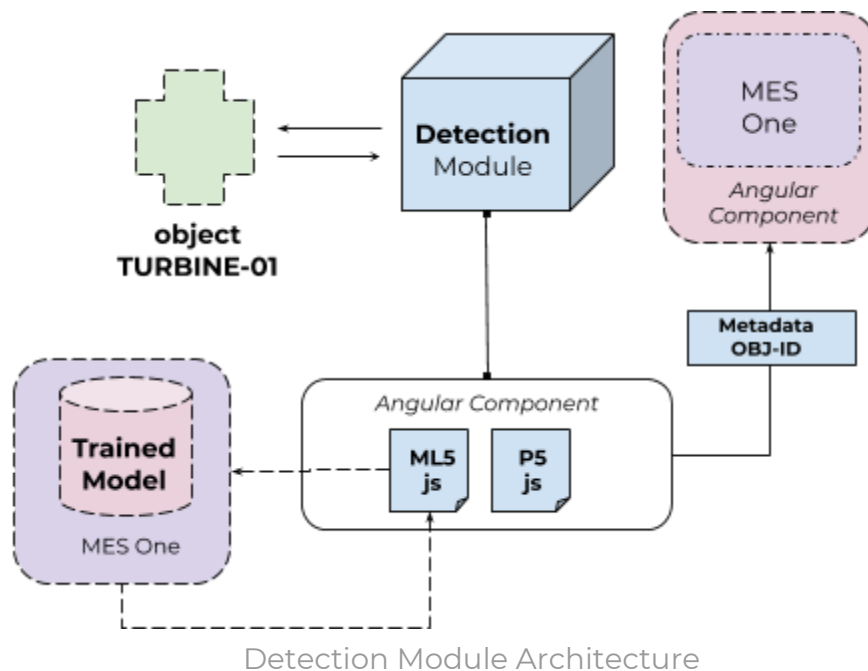
In case the model is providing inaccurate detection, during testing, the ML Model can be retrained, with different values for parameters like epoch, confidence etc., for better results.

2.5 Save/Upload Model

Once the ML Model has been tested, the model can then be saved/upload to CouchDB, along with its metadata.

3. Detection Model & Integration

Below is the architecture that is followed during implementation of Detection Model & Integration...



3.1 Object

The same objects used for training - images of various equipment - shall be used for detection - *refer to section 2.1*.

3.2 Detection

The model for detection shall be retrieved from the Trained ML Model stored in the CouchDB Setup - *refer to section 2.5*

3.3 Metadata

On successfully completing the Detection Process, the corresponding Metadata (ObjID) is then sent as an output parameter from the angular component.

3.4 MES One Integration

The component in MES One Integration shall be programmed (by GE) in order to retrieve the above parameter, from the child component, and perform further actions.

4. Need Clarifications

#	Item	Remarks
1	3 Equipment Images for Training - each Equipment should have 4 Images in different angles, and corresponding Metadata (ObjID).	Siva (GE)
2	CouchDB Setup/API Details for ML Model Upload & Access.	Siva (GE)
3	<p>With respect to Section 3.4, the child component (which will be detecting the images for ObjID) will send the Metadata to its parent component (MES One). Would the parent component call an API and then send data like Instructions to the child component? If yes, the parent component should also send the corresponding image's url (used during model training) as an additional parameter to the child component.</p> <p>OR</p> <p>Should the child component directly hit the corresponding API and get instructions data & the url for corresponding image?</p>	Eswar (GE)

5. Delivery

#	Item	Date
1	Interim Build & Demo - Training Model & Learning Transfer	13 Jan, 2020
2	Final Build & Delivery - Training Model & Learning Transfer + Detection Model & Integration	17 Jan 2020