**Assignment 13**

**Name: -**

**Enrollment No: -**

**E-mail: -**

1. **Introduction**

In this assignment, the task was to build an anomaly detection system using Teachable Machine’s Image Project, train it on a manufactured product not used in the Inspectors Ally example, and deploy the trained model into a Streamlit web application. The app should classify products as either Normal or Defective, and include a real-time camera-based interface for bonus points.

1. **Product Used**

* Product Name: Ball Bearing
* Why Chosen: It is a commonly manufactured component and was not used in the Inspectors Ally demo.
* Image Classes:
  + Normal Ball Bearing
  + Defective Ball Bearing

1. **Dataset Collection and Preparation**

* Images Collected: ~60 per class (Normal & Defective)
* Method: Images were manually sourced and labelled directly in Teachable Machine.
* Preprocessing: Handled automatically by Teachable Machine during upload and training.
* Labelling: Done within Teachable Machine’s interface with class names clearly defined.

1. **Model Training – Teachable Machine**

* Platform Used: Teachable Machine – Image Project
* Model Type: Image Classification
* Model Hyperparameters Tuned:
  1. Epochs: 50
  2. Batch Size: 16
  3. Learning Rate: 0.001
* Export Format:
  1. Initially exported as TensorFlow .h5
  2. Converted to ONNX (model. onnx) for compatibility with Streamlit Cloud

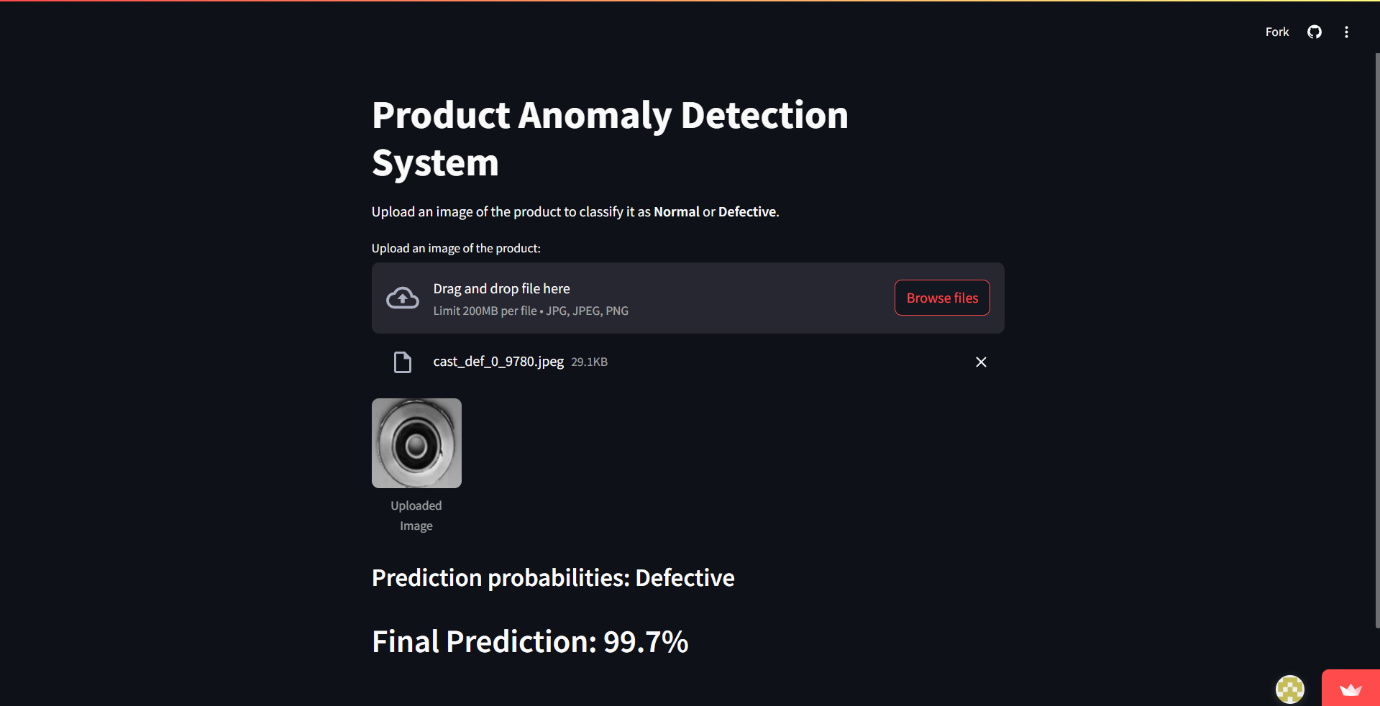
1. **Streamlit App Implementation**

Features:

* Upload image to classify Ball Bearing as Normal or Defective.
* Display confidence percentage for both classes.
* Show uploaded image in the UI.
* Clear label output with color-coded results.
* Interface design inspired by *Inspectors Ally*.

Bonus Implemented:

1. Real-time anomaly detection using live webcam (streamlit-webrtc)
2. System detects anomalies without the need to upload images manually
3. **Output Screenshot**



1. **Deployment Links**
2. GitHub Repository: <https://github.com/Hitesh4011/Intel-week-13>
3. Streamlit App: <https://intel-week-13-assignment.streamlit.app>
4. **Conclusion**

This project demonstrates an end-to-end implementation of an anomaly detection system using:

* Teachable Machine for training,
* ONNX for portable model format,
* Streamlit for interactive deployment.

By extending the assignment with a real-time camera interface, this solution enhances usability and supports fast visual inspection in manufacturing.