# PROJECT REPORT

## MOTOR ASSEMBLY MONITORING THROUGH IMAGE ANALYSIS


**Team Members-**

1)**Arpan Kumar**

4th year ECE

Academy of technology

2) **Ayush Kumar**

4th year EEE

Academy of technology

**Team Mentor-**
**Anik Sardar**

Manager: Sensor QC & Calibration

Prophecy Sensorlytics India PVT.LTD


### PROBLEM STATEMENT:

The project focuses on developing an image analysis system to monitor the correct insertion of magnets into the rotor slots of a motor during assembly. Additionally, it includes a 3D surface scanning component to evaluate the surface roughness and depth variability of the motor post-assembly. The goal is to leverage advanced computer vision techniques to automate quality control processes, thereby reducing the reliance on manual inspections and enhancing overall production efficiency.

### WORK APPROACH:

1. **Data Collection and Preprocessing:**
   - Gather stereo and RGB images of motor assemblies, including both properly placed and misaligned magnets.
   - Convert RGB images to grayscale and preprocess them for noise reduction.
2. **Image Comparison Algorithms:**
   - Develop algorithms using Structural Similarity Index (SSIM) to compare test images against reference images.
   - Implement pixel-wise difference calculations to quantify deviations.
3. **Depth Analysis Using Stereo Images:**
   - Generate disparity maps from stereo images to analyse depth discrepancies.
   - Utilize depth analysis techniques to detect improperly inserted magnets.
4. **3D Surface Scanning and Roughness Analysis:**

- Extract 3D data from stereo images to create a point cloud representation of the motor surface.
- Analyse surface roughness and depth variability using statistical methods.

5. **Decision-Making System Development:**
    - Create a threshold-based system to automatically flag assembly or surface quality issues based on analysis metrics.
    - Analysed SSIM scores, insertion percentages, and surface roughness to make decisions on assembly quality.

6. **Reporting and Visualization:**
    - Generated reports using matplotlib with visualizations to display the results.
    - Created quality control reports indicating any assembly or surface issues detected.

**<u>Work Products and Deliverables:</u>**

- **Software Programs**: Python scripts implementing image analysis, feature matching, depth analysis, and 3D surface scanning.
- **Datasets**: Collection of stereo and RGB images showing different magnet alignment conditions.
- **GitHub Repository**: A repository containing all source code, datasets, and documentations are present in this link:

  https://github.com/arpankumar2520/IDEAS_TIH_Project/tree/my-new-branch

**<u>User Manual:</u>**

1. **Installation Requirements:**
    - Ensure Python is installed on your system.
    - Install Jupyter Notebook.
    - Install necessary libraries using the command:

      pip install opencv-python scikit-image matplotlib

2. **Running the Application:**

   **Execution**:

   - Run the main () function in the provided script.
   - Specify the path for the reference image and the folder containing test images.

   **Output**:

   - View the results, including SSIM scores, surface roughness values, and insertion percentages.
   - Check the automatically generated reports indicating quality control issues.

**<u>Technical Manual:</u>**

**Datasets:**

- **Captured Images**: Stereo and RGB images of motor assemblies for analysis.

**Software Libraries Used:**

- **OpenCV**: Image processing and feature matching.
- **scikit-image**: SSIM calculation and surface roughness analysis.
- **Matplotlib**: Visualization of SSIM score distributions and analysis results.

**Program Documentation:**

- **load_and_preprocess_images**: Loads images, resizes the test image to the reference image's dimensions, and converts images to grayscale.
- **compare_images**: Calculates SSIM between reference and test images.
- **determine_ssim_threshold**: Determines the SSIM threshold for classifying assembly accuracy.

- **analyze_surface**: Calculates surface roughness using standard deviation of the depth map.
- **feature_matching**: Matches features between the reference and test images using ORB.
- **depth_analysis**: Generates disparity map for depth analysis.
- **decision_making**: Flags quality control issues based on thresholds for SSIM, insertion, and roughness.

**GitHub Repository Structure**

- Reference image path: **magnet_insertion-proper.jpg**
- Input folder path: **new_images**
- Python code: **Project_program.ipynb**
- Code pdf: **Project_program.pdf**
- Reason for selecting or rejecting the threshold: **Selecting_threshold.docx**
- Output image folder**: output images.zip**
- Project report: **MOTOR ASSEMBLY MONITORING THROUGH IMAGE ANALYSIS.pdf**
- Presentation**: MOTOR ASSEMBLY MONITORING THROUGH IMAGE ANALYSIS.pptx**

# Relevance of the project to industrial applications

- The project "Motor Assembly Monitoring through Image Analysis" is highly relevant to industrial applications, particularly in the manufacturing sector. Here are some key aspects of its relevance:

1. **Quality Assurance**: By employing image analysis techniques to monitor motor assembly, manufacturers can ensure that each component is correctly positioned and fully inserted. This real-time monitoring reduces defects and improves overall product quality, essential for maintaining high standards in industrial applications.
2. **Increased Efficiency**: Automating the inspection process through image analysis significantly speeds up quality control compared to manual inspections. This

efficiency allows for faster production cycles and the ability to identify and rectify issues immediately, minimizing downtime.

3. **Cost Reduction**: Detecting assembly issues early in the production process helps reduce costs associated with rework, waste, and product recalls. This project can lead to substantial savings for manufacturers by minimizing defects and improving yield rates.
4. **Enhanced Data Analytics**: The integration of image analysis with data analytics enables manufacturers to gather valuable insights regarding assembly processes. Analyzing metrics such as SSIM, insertion percentages, and surface roughness can lead to continuous improvement initiatives and better decision-making.
5. **Flexibility in Production**: The ability to adapt image analysis techniques to various types of motor assemblies makes this approach versatile. Manufacturers can implement these methods across different product lines, enhancing flexibility and responsiveness to market demands.
6. **Compliance and Safety**: In industries where safety is critical, such as automotive or aerospace, ensuring the integrity of motor assemblies is paramount. Image analysis provides an objective assessment, helping companies comply with industry regulations and safety standards.
7. **Reduced Human Error**: Automated inspection reduces reliance on human judgment, which can be prone to error, fatigue, or oversight. This leads to more consistent and reliable quality assessments.

# Conclusion:

The program implements a comprehensive monitoring system for motor assembly through image analysis, focusing on magnet insertion quality. It begins by loading and preprocessing reference and test images, then compares them using the Structural Similarity Index (SSIM) and pixel-wise differences to evaluate assembly integrity. The system calculates various metrics, including surface roughness, magnet insertion percentage, and alignment scores, to assess the quality of assembly. Feature matching and depth analysis further enhance the understanding of discrepancies between the images. The decision-making module interprets these metrics against predefined thresholds to identify potential assembly issues, such as low SSIM values, incomplete magnet insertion, or high surface roughness. Finally, the program visualizes results and SSIM score distributions, enabling clear assessment of magnet insertion quality. This approach enhances quality control in manufacturing processes, ensuring that motor assemblies meet stringent standards for performance and reliability.

# References:

- **Used tutorials:** OpenCV, Matplotlib, scikit, python
- **YouTube-** learned different thresholding techniques and Image Comparison Algorithms