MOTOR ASSEMBLY MONITORING THROUGH IMAGE ANALYSIS

PROJECT ID-IN02

Team Members-

1)**Arpan Kumar**

4th year ECE

Academy of technology

2) Ayush Kumar

4th year EEE

Academy of technology

Team Mentor-

Mr. 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.

Work products and deliverables

- **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.

  Input folder name of images-**new_images**

  reference image name-**magnet_insertion-proper.jpg**

- **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

User Manual

1. Installation Requirements:

- Ensure Python is installed on your system.
- Install Jupyter Notebook.
- Install necessary libraries :

1)OpenCV 3)scikit-image
2)JSON 4)Matplotlib
5)SciPy-stats

  - For using library have to write pip install with given libraries.

2 . Running the Application:

Execution:

- Use the JSON config file for giving the values to two dictionary keys where keys are' ref file': "**magnet_insertion-proper.jpg**" and' test dir': "**new_images**"
- Run the main() function in the provided script.

Output:

- A verdict saying whether the materials in the image is partially inserted or fully inserted and to what percentage (approximation) along with the scores and threshold.
- Additionally also view the results, including SSIM scores, surface roughness values, and insertion percentages.

Technical Manual

Datasets:

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

Software Libraries Used:

- **OpenCV**: For Image processing and feature matching.
- **scikit-image**: For SSIM calculation and surface roughness analysis .
- **Matplotlib**: For Visualization of SSIM score distributions and analysis results.
- **SciPy-stats**: For provides a range of statistical functions.
- **JSON:** For giving the values for dictionary keys(values for reference and test images)

GitHub Repository Structure

- Reference image path: **magnet_insertion-proper.jpg**
- Input folder path: **new_images**
- JSON file**: config.json**
- Python code: **main.py**
- Code pdf: **main.pdf**
- Output image folder**: output images.zip**
- Project report: **MOTOR ASSEMBLY MONITORING THROUGH IMAGE ANALYSIS.pdf**
- Presentation**: MOTOR ASSEMBLY MONITORING THROUGH IMAGE ANALYSIS.pptx**

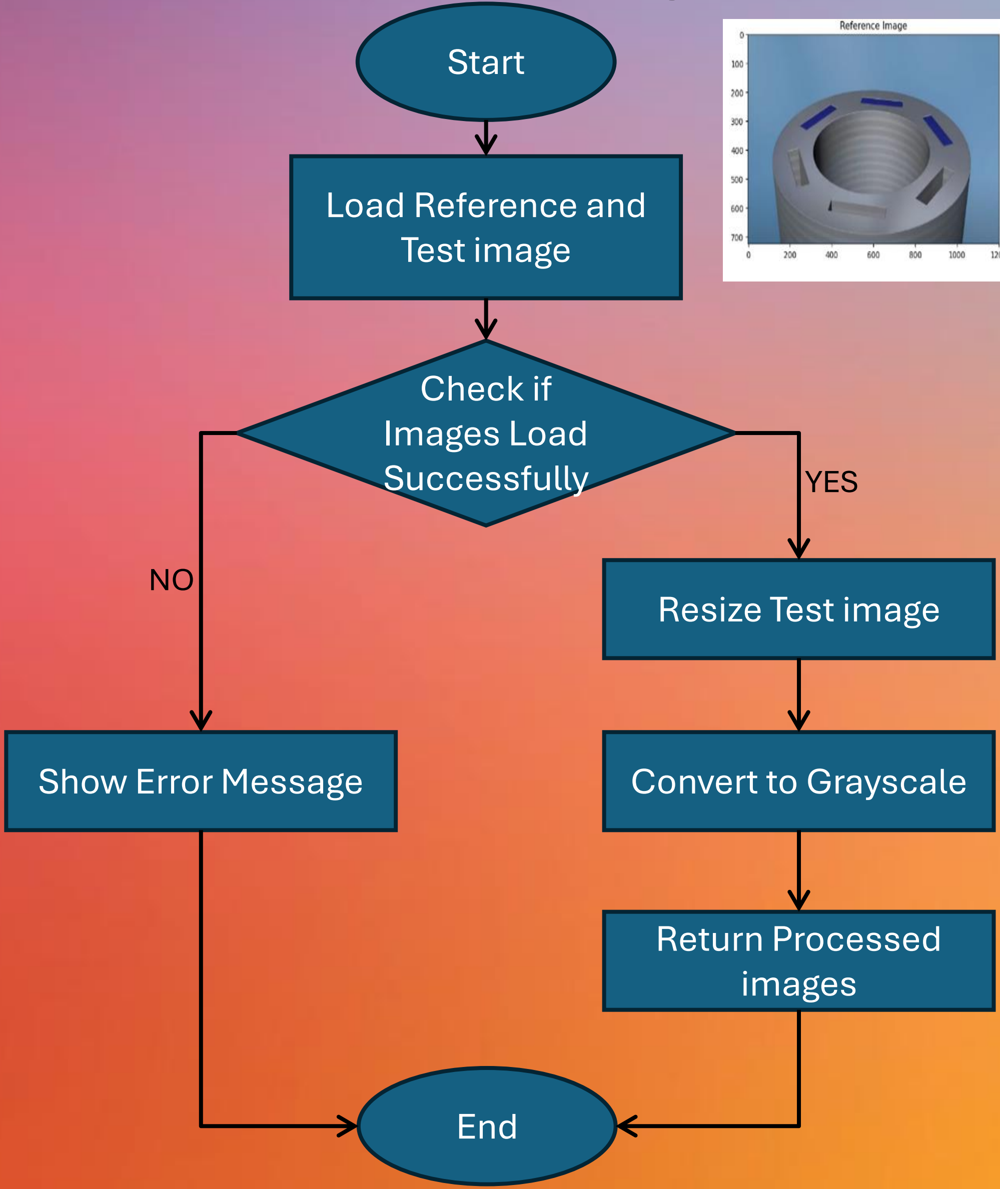
Program Documentation:

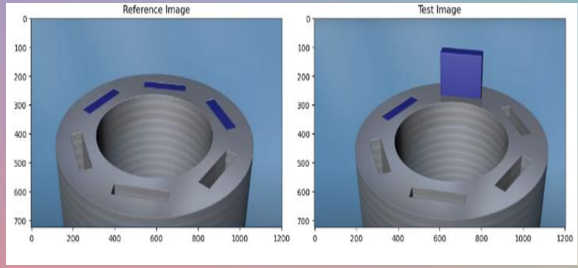
list of functions are used:

- **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.

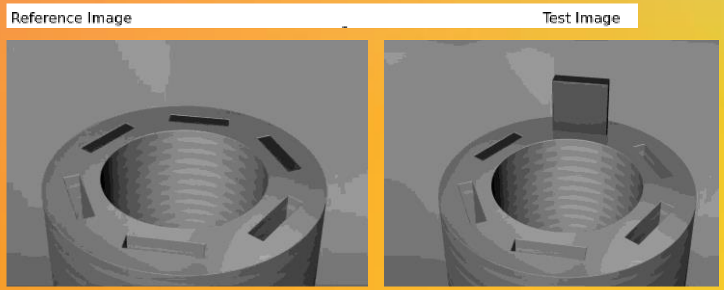
Analysis of Program:

- **Load and preprocess images:** Loads images, resizes the test image to the reference image's.

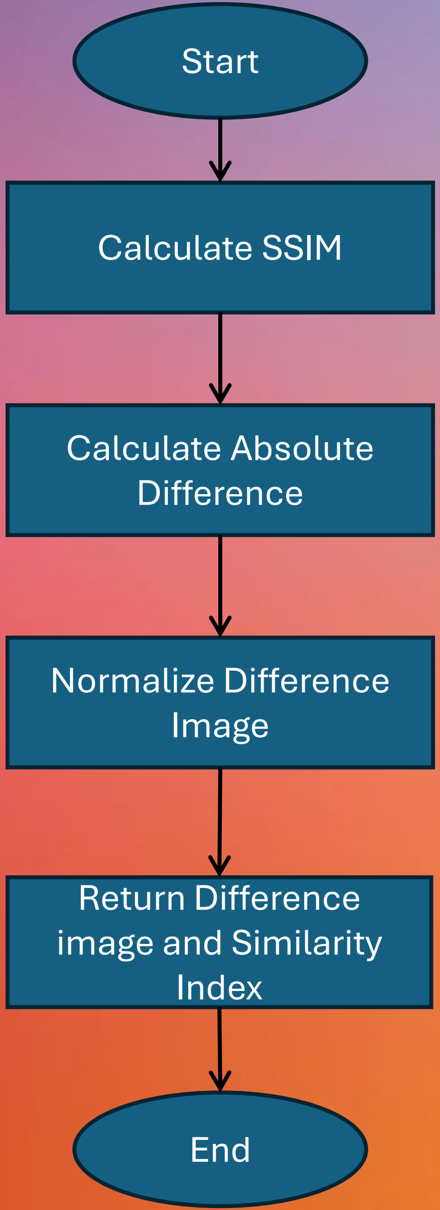




The function first loads the reference and test images using `cv2.imread`. If the loading process fails, it displays an error message and terminates. Upon successful loading, the test image is resized to match the dimensions of the reference image using `cv2.resize`. Both images are then converted to grayscale using `cv2.cvtColor`. Finally, the function returns the original reference image, the resized test image, and their grayscale versions, as shown here . We used grayscale images because it have single intensity value per pixel and reduces complexity and easy to apply SSIM , Feature matching algorithm.



- **Comparing images:** Calculates SSIM between reference and test images.



We compute the similarity index (SSIM) between the reference and test grayscale images to assess structural similarity. The SSIM score, ranging from -1 to 1, measures how closely the two images align, with a score of 1 indicating identical structures. High SSIM values suggest proper alignment, while lower values indicate potential alignment issues. For example, the SSIM for a fully inserted magnet is 1.0000, whereas the test images show an SSIM of approximately 0.8689, indicating alignment problems. Additionally, we calculate the absolute pixel-wise difference using `cv2.absdiff` and normalize this difference to a 0-255 range using `cv2.normalize`. The normalized difference image highlights mismatches and reveals misalignments or defects that SSIM alone may not capture. In this case, all images show magnet misalignment.

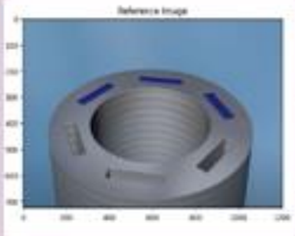
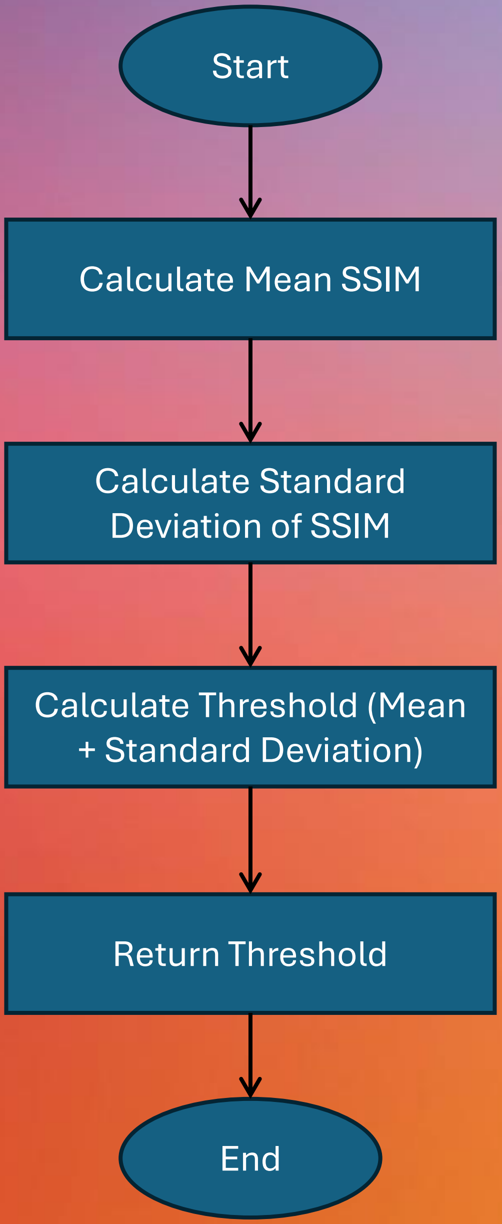
Reference Image	Test Image	SSIM	Pixel-wise Difference
		0.8765	
		0.8701	
		0.8669	
		0.8753	
		0.8539	
		0.8708	

Fig: Displaying SSIM

- **Determining the ssim threshold:** Determines the SSIM threshold for classifying assembly accuracy.



We compute the mean and standard deviation of the SSIM scores using `np.mean` and `np.std`, then calculate the threshold by adding the mean and standard deviation, returning the resulting threshold.

The reason for using the mean + standard deviation is that it provides a reliable, adaptive similarity level, avoiding overly strict or lenient arbitrary thresholds. This approach improves classification accuracy in quality control by accommodating acceptable variations. The mean plus standard deviation is chosen because it closely aligns with the mean SSIM values, unlike higher thresholds, which tend to differ significantly.

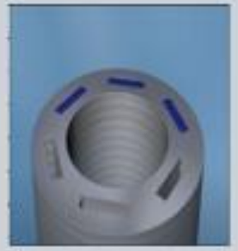
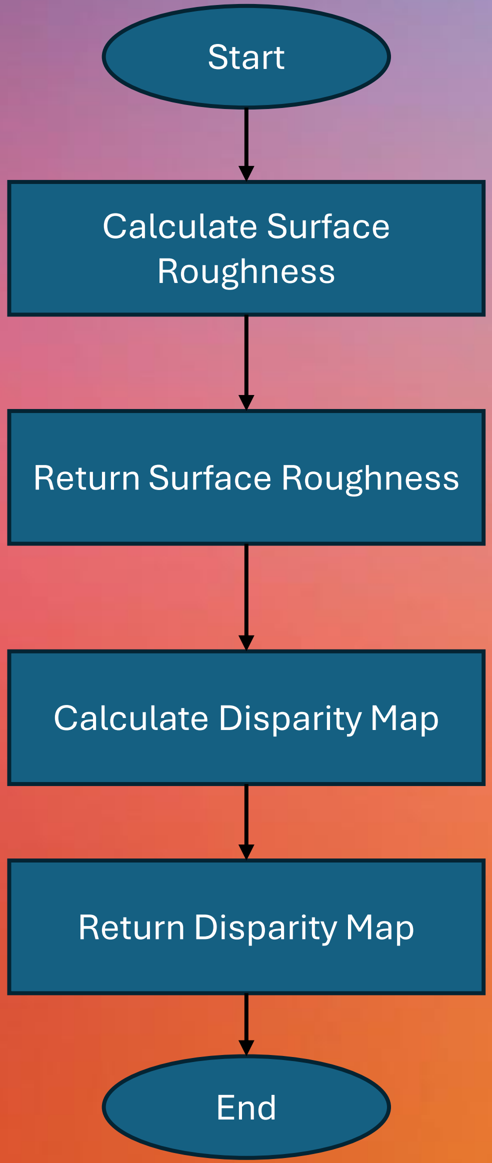
	Test Image	SSIM	Threshold(mean + deviation)	Threshold(mean +2 * deviation)	Threshold (mean +3 * deviation)
		0.8765	0.9286	0.9729	1.0172
		0.8701	1.0000	1.0649	1.1299
		0.8669	0.9326	0.9799	1.0271
		0.8753	0.9388	0.9892	1.0396
		0.8539	0.9564	1.0166	1.0768
		0.8708	0.9459	1.0007	1.0555

Fig: Comparing different SSIM thresholding

- **Analysing surface and Depth :** Calculates surface roughness using standard deviation of the depth map. Generates disparity map for depth analysis.



We calculate the disparity map by finding the absolute pixel-wise difference between the reference and test grayscale images using `cv2.absdiff` and return the result. Surface roughness is then computed by calculating the standard deviation of the disparity map, which quantifies the variation in pixel values and provides insight into texture consistency. This analysis helps in detecting defects, ensuring quality, and supporting smooth operation in assembled parts. In this case, the approximated surface roughness is 22.3971, which is used to assess whether the magnet is properly inserted.

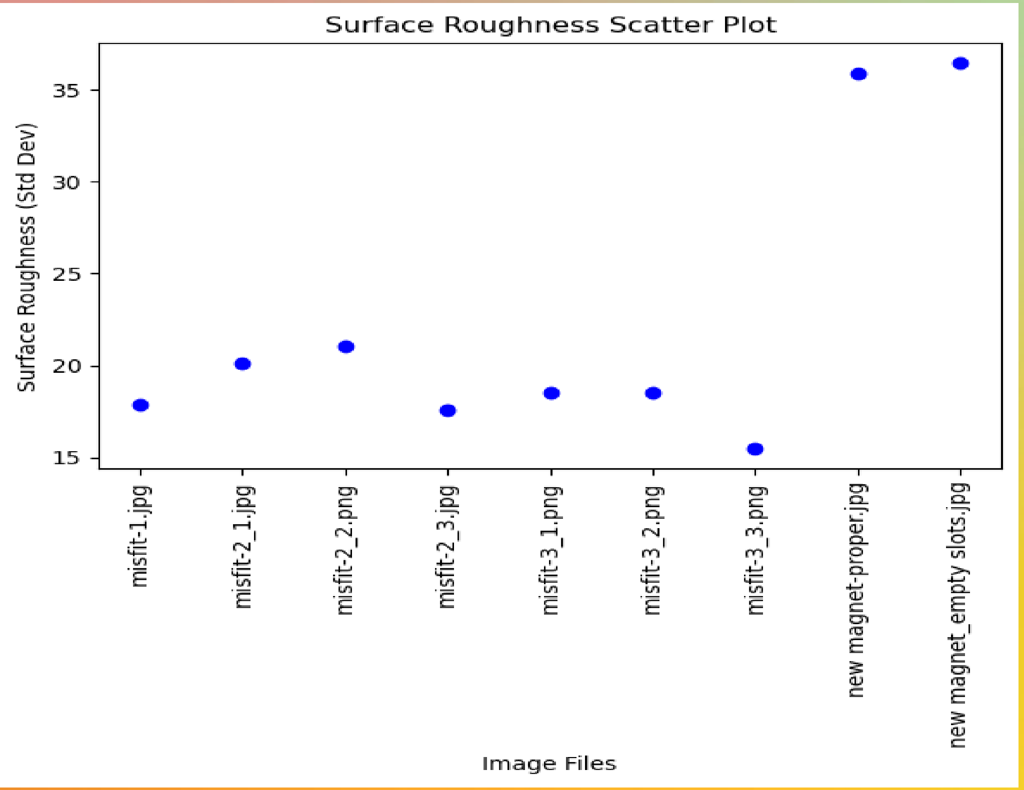
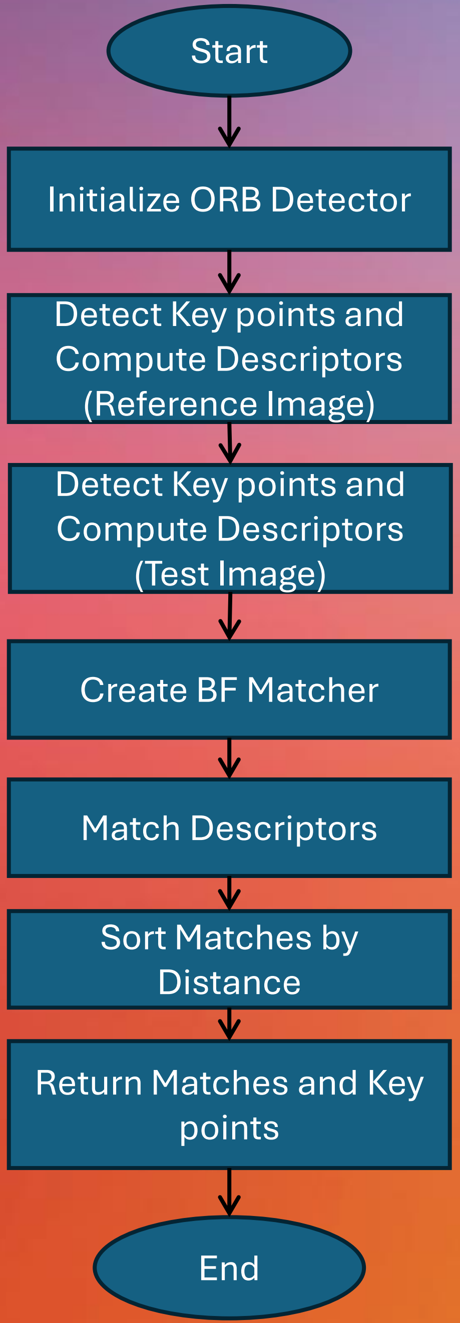


Fig: Surface Roughness values of all images in scatter plot

- **Feature matching:** Matches features between the reference and test images using ORB.



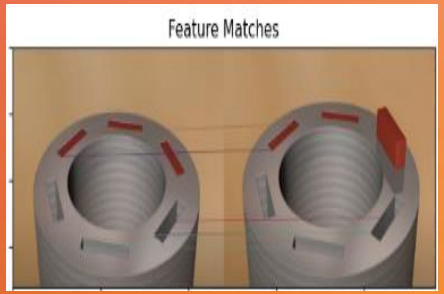


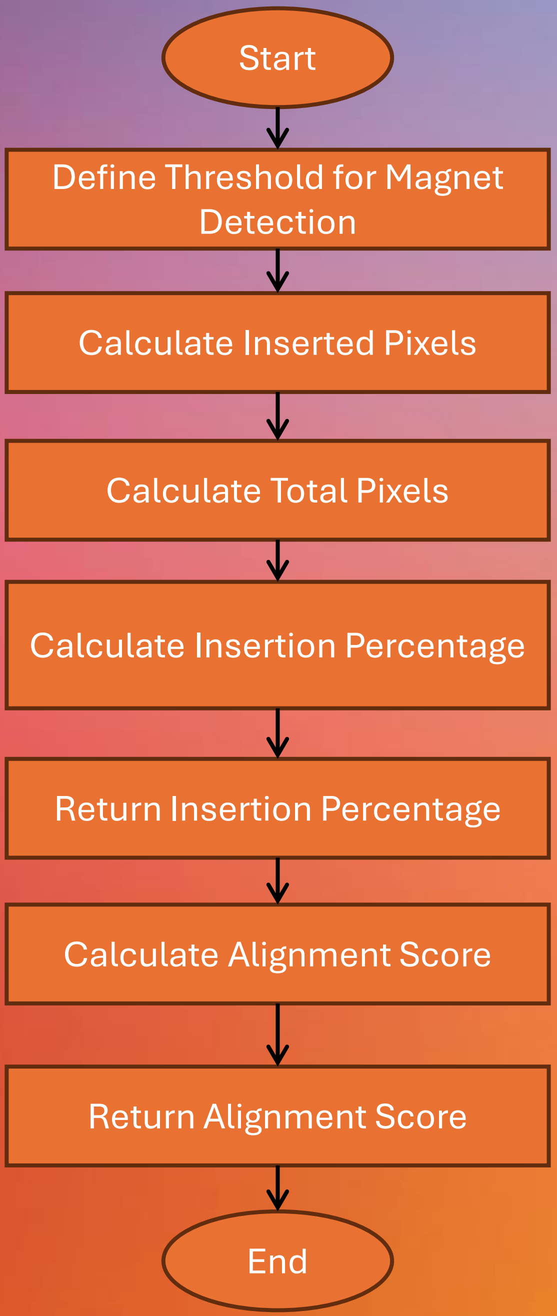
Fig: Feature matching

The ORB (Oriented FAST and Rotated BRIEF) algorithm here is used to match features between a reference image and a test image. Here's how it works: First, ORB detects keypoints in both the reference (`ref_gray`) and test (`test_gray`) grayscale images. Keypoints are distinct points in an image, often at corners or edges, that can be reliably matched across similar images. After detecting these keypoints, ORB then calculates descriptors for them. Descriptors are compact representations that capture information about the local image area around each keypoint, making each keypoint unique and identifiable.

Next, ORB uses a BRIEF (Binary Robust Independent Elementary Features) descriptor, which is both efficient and robust, to match keypoints between images. The code employs a Brute-Force Matcher (`BFMatcher`), which compares descriptors from the reference and test images to find the closest matches between keypoints. These matches are sorted by distance, where a lower distance indicates a stronger similarity between matched keypoints. The code selects the best matches by choosing those with the shortest distances.

Finally, to visualize the matching process, the code draws lines connecting matching keypoints in the reference and test images. This visualization provides a clear indication of how similar or different the two images are based on these aligned keypoints. Using ORB in this way allows the program to assess alignment and similarity beyond pixel-by-pixel comparisons, offering a more sophisticated and feature-based approach to comparing images.

- **Magnet insertion percentage and alignment score**

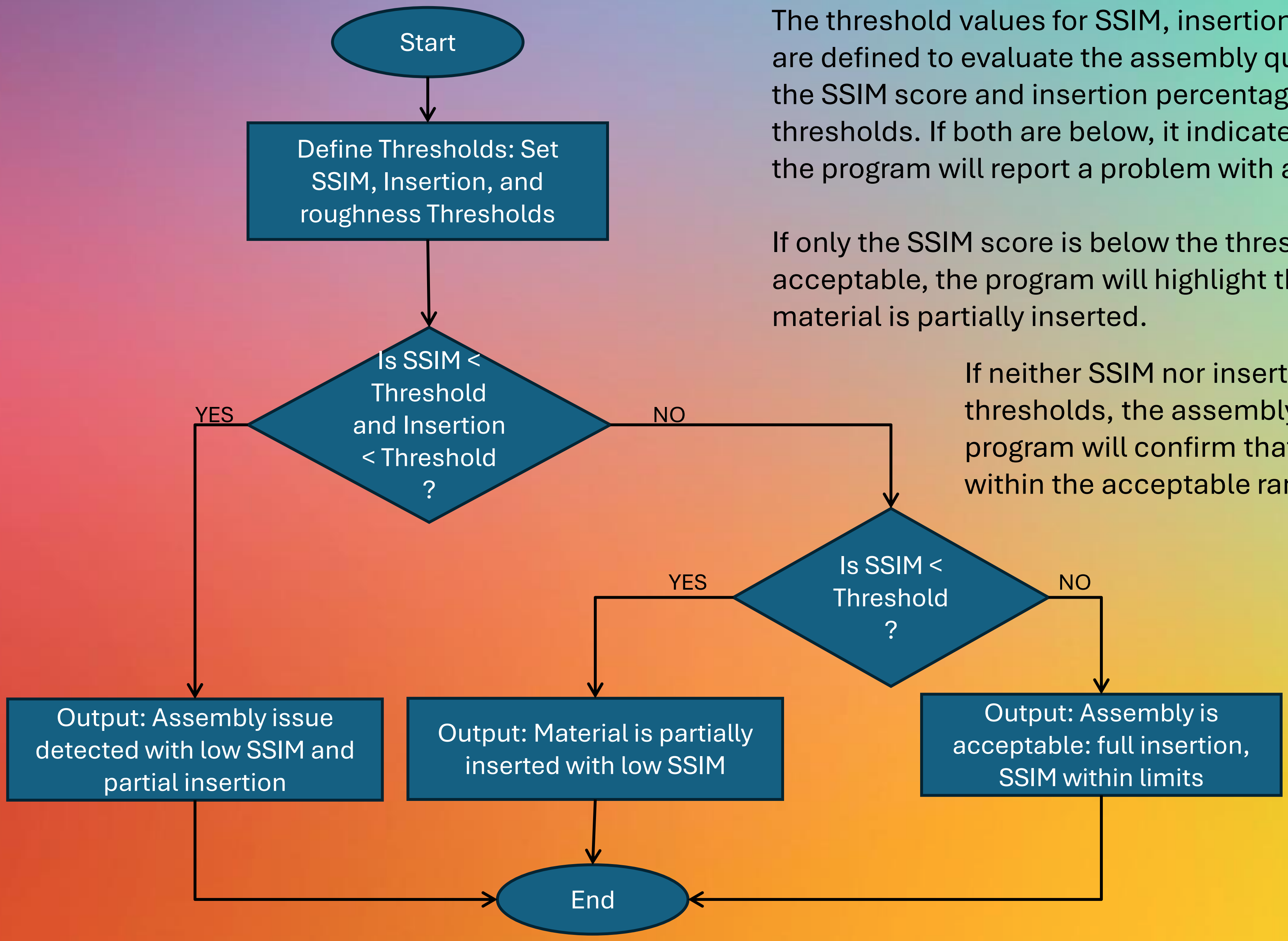


We set a threshold value of 130 to identify pixels corresponding to inserted magnets. By counting the number of pixels below this threshold, we can determine magnet insertion. The total number of pixels in the test grayscale image is calculated, and the percentage of inserted pixels is computed by dividing the inserted pixel count by the total pixel count and multiplying by 100. This function returns the calculated magnet insertion percentage, which helps verify proper magnet insertion in the assembly and informs decision-making. Additionally, the alignment score is calculated using the Structural Similarity Index (SSIM), which reflects how closely the test image matches the reference in terms of structure, texture, and other characteristics.

Reference Image	Test Image	Magnet insertion percentage	Alignment Score in %
magnet_insertion-proper.jpg 	Processing: misfit-1.jpg	45.58%	87.01%
	Processing: misfit-2_1.jpg	46.26%	86.08%
	Processing: misfit-2_2.png	55.04%	85.39%
	Processing: misfit-2_3.jpg	46.22%	87.08%
	Processing: misfit-3_2.png	48.56%	86.69%
	Processing: misfit-3_3.png	46.80%	87.65%

Fig: Values of magnet insertion and alignment score.

Decision making: Quality control issues based on thresholds for SSIM, insertion, and roughness.



The threshold values for SSIM, insertion percentage, and surface roughness are defined to evaluate the assembly quality. The program first checks if both the SSIM score and insertion percentage are below their respective thresholds. If both are below, it indicates a significant assembly issue, and the program will report a problem with alignment and insertion.

If only the SSIM score is below the threshold, but the insertion percentage is acceptable, the program will highlight the alignment issue, noting that the material is partially inserted.

If neither SSIM nor insertion percentage falls below their thresholds, the assembly is deemed acceptable, and the program will confirm that both alignment and insertion are within the acceptable range.

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. **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.

5. **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.

6. **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

THANK YOU