

```
In [1]: import os
import cv2
import numpy as np
from skimage.metrics import structural_similarity as ssim
import matplotlib.pyplot as plt
from scipy.stats import pearsonr, spearmanr
# Load and preprocess images
def load_and_preprocess_images(ref_path, test_path):
    ref_image = cv2.imread(ref_path)
    test_image = cv2.imread(test_path)
    if ref_image is None or test_image is None:
        raise ValueError(f"Error: Unable to load images from {ref_path} or {test_path}.")
    # Resize test image to match reference image dimensions
    test_image_resized = cv2.resize(test_image, (ref_image.shape[1], ref_image.shape[0]))
    # Convert to grayscale
    ref_gray = cv2.cvtColor(ref_image, cv2.COLOR_BGR2GRAY)
    test_gray = cv2.cvtColor(test_image_resized, cv2.COLOR_BGR2GRAY)

    return ref_gray, test_image_resized, ref_gray, test_gray
```

```
# Compare Images using SSIM and pixel-wise differences
def compare_images(ref_gray, test_gray):
    similarity_index = ssim.reflected_ssims(ref_gray, test_gray, full=True)
    print(f"Structural Similarity Index (SSIM): {similarity_index[0]}")
```

```
# Calculate pixel-wise differences
diff_image = cv2.absdiff(ref_gray, test_gray)
diff_image_normalized = cv2.normalize(diff_image, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)
```

```
return diff_image_normalized, similarity_index
```

```
# Calculate SSIM threshold using statistical method (mean + std deviation)
def determine_ssim_threshold(statistic_scores):
    mean_ssim = np.mean(statistic_scores)
    std_ssim = np.std(statistic_scores)
```

```
# Using mean + standard deviation as threshold
threshold_mean_std_ssim = mean_ssim + std_ssim
```

```
print(f"Threshold for SSIM classification: {threshold_mean_std_ssim * (threshold:.4f)}")
```

```
return threshold
```

```
# Visualize the SSIM score distribution
def visualize_ssim_distribution(partial_ssim_scores, full_ssim_score, threshold=None):
    plt.figure(figsize=(12, 8))
    plt.hist(partial_ssim_scores, bins=20, alpha=0.7, color='red', label="Partial Insertion SSIMs")
    plt.hist(full_ssim_score, bins=20, alpha=0.7, color='blue', linestyle='dashed', linewidth=2, label="Mean SSIM (Partial Insertion)")
    plt.xlabel("SSIM Score Distribution for Partially Inserted Magnets")
    plt.ylabel("Frequency")
    plt.legend()
    plt.title("SSIM Score Distribution for Partially Inserted Magnets")
    plt.show()
```

```
# 3D Histogram
fig = plt.figure(figsize=(12, 9))
ax = fig.add_subplot(111, projection='3d')
hist, bins = np.histogram(partial_ssim_scores, max=partial_ssim_scores), 100
bincenters = (bins[1:] + bins[:-1]) / 2
ax.bar3d(bincenters, hist, bins[1:], width=0.7, color='red', alpha=0.7)
ax.set_xlabel("SSIM Score")
ax.set_ylabel("Frequency")
ax.set_zlabel("Density")
ax.legend()
plt.show()
```

```
# If threshold is not None:
    ax.axvline(x=threshold, color='purple', linestyle='dashed', linewidth=2, label=f"Threshold = {threshold:.4f}")
```

```
ax.set_xlabel("SSIM Score")
ax.set_ylabel("Frequency")
ax.set_zlabel("Density")
ax.legend()
plt.show()
```

```
# Surface roughness analysis
def analyze_surface_depth_map():
    disparity_map = np.load(depth_map)
    print(f"Surface Roughness (Std Dev): {surface_roughness:.4f}")
    return surface_roughness
```

```
# Feature matching
def feature_matching(ref_gray, test_gray):
    orb = cv2.ORB_create()
    keypoints_ref, descriptors_ref = orb.detectAndCompute(ref_gray, None)
    keypoints_test, descriptors_test = orb.detectAndCompute(test_gray, None)
    bf_matcher = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
    matches = bf_matcher.match(descriptors_ref, descriptors_test)
    matches = sorted(matches, key=lambda x: x.distance)
    return matches, keypoints_ref, keypoints_test
```

```
# Depth analysis
def depth_analysis(ref_gray, test_gray):
    disparity_map = cv2.absdiff(ref_gray, test_gray)
    return disparity_map
```

```
# Calculate correlation coefficients (Linear and Non-Linear)
def calculate_correlation(ref_gray, test_gray):
    ref_flat = ref_gray.flatten()
    test_flat = test_gray.flatten()
    linear_corr = pearsonr(ref_flat, test_flat)
```

```
    nonlinear_corr = spearmanr(ref_flat, test_flat)
```

```
    print(f"Linear Correlation (Pearson): {linear_corr:.4f}")
    print(f"Non-linear Correlation (Spearman): {nonlinear_corr:.4f}")
```

```
    return linear_corr, nonlinear_corr
```

```
# Magnet insertion percentage
def determine_magnet_insertion(test_gray):
    threshold_mean_std_ssim = determine_ssim_threshold(partial_ssim_scores)
    total_pixels = test_gray.size
    insertion_percentage = (inserted_pixels / total_pixels) * 100
    print(f"MagNet Insertion Percentage: {insertion_percentage:.2f}%")
    return insertion_percentage
```

```
# Alignment score
def calculate_alignment_score(ssim_index):
    alignment_score = ssim_index * 100
    print(f"Alignment Score: {alignment_score:.2f}%")
    return alignment_score
```

```
# Decision-making system
def decision_making_system(insertion_percentage, surface_roughness, alignment_percent):
    threshold_mean_std_ssim = determine_ssim_threshold(partial_ssim_scores)
    threshold_insertion = 0.270
    threshold_roughness = 22.3971
```

```
    if ssim_index < threshold_mean_std_ssim and insertion_percentage < threshold_insertion:
        print(f"Decision: Assembly issue detected (low SSIM and magnet not fully inserted).")
```

```
    elif insertion_percentage < threshold_insertion:
        print(f"Decision: Magnet not fully inserted.")
```

```
    elif surface_roughness > threshold_roughness:
        print(f"Decision: Surface quality issue detected (high surface roughness).")
```

```
    else:
        print(f"Decision: Assembly is acceptable.")
```

```
# Main execution function
def main():
    input_folder = "missfit_1-image_proper.jpg" # Path to the fully inserted magnet image
    input_folder = "new_images" # Folder containing test images (partially inserted magnets)
```

```
    # Load to store SSIM scores
    full_ssim_scores = []
    partial_ssim_scores = []

    for img in images_in_the_input_folder:
        test_image_name = img.name
        full_ssim_scores.append(ssim.reflected_ssims(ref_gray, test_image_name))

        # Process each image
        for test_image_name in images_in_the_input_folder:
            test_image_name = img.name
            print(f"Processing: {test_image_name}")

            # Load and preprocess images
            ref_gray, test_image, ref_gray, test_gray = load_and_preprocess_images(ref_path, test_path)

            # Compare Images
            diff_image_normalized, ssim_index = compare_images(ref_gray, test_gray)

            # Collect SSIM score for further analysis
            if f"full_{test_image_name}" in test_image_name.lower():
                full_ssim_scores.append(ssim_index)
            else:
                partial_ssim_scores.append(ssim_index)
```

```
    # Calculate and display the threshold for the current image
    current_threshold = determine_ssim_threshold(partial_ssim_scores)
```

```
    print(f"Current SSIM Threshold for {test_image_name}: {current_threshold:.4f}")
```

```
    # Linear Correlation (Pearson)
    linear_corr = pearsonr(ref_gray, test_gray)
```

```
    # Non-linear Correlation (Spearman)
    nonlinear_corr = spearmanr(ref_gray, test_gray)
```

```
    # Magnet Insertion percentage
    insertion_percentage = determine_magnet_insertion(test_gray)
```

```
    # Alignment score
    alignment_score = calculate_alignment_score(ssim_index)
```

```
    # Decision-making
    decision_making(ssim_index, insertion_percentage, surface_roughness)
```

```
    # Visualize results with larger images
    plt.figure(figsize=(12, 12)) # Larger figure for better visibility
    plt.subplot(2, 3, 1)
    plt.title("Reference Image")
    plt.imshow(ref_gray)
    plt.subplot(2, 3, 2)
    plt.title("Test Image")
    plt.imshow(test_gray)
    plt.subplot(2, 3, 3)
    plt.title("Pixel-wise Differences")
    plt.imshow(diff_image_normalized)

    plt.subplot(2, 3, 4)
    plt.title("Feature Matches")
    plt.imshow(feature_matches)
    plt.subplot(2, 3, 5)
    plt.title("Disparity Map")
    plt.imshow(disparity_map)
    plt.subplot(2, 3, 6)
    plt.title("Dispersion Map")
    plt.imshow(surface_roughness)

    plt.tight_layout()
    plt.show()
```

```
# Determine the threshold for SSIM classification
ssim_threshold = determine_ssim_threshold(partial_ssim_scores)
```

```
# Compute SSIM score for fully inserted magnet (reference image)
fully_inserted_image = load_and_preprocess_images(ref_path)[2]
```

```
fully_inserted_ssims = ssim.reflected_ssims(fully_inserted_image, test_image)
```

```
# Visualize SSIM score distribution
visualize_ssim_distribution(partial_ssim_scores, full_ssim_scores, ssim_threshold)
```

```
print(f"The SSIM score for the fully inserted magnet falls 'above' if full_ssim_score > ssim_threshold else 'below' the threshold.")
```

```
if __name__ == "__main__":
    main()
```

```
Processing: missfit_1-image_proper.jpg
Structural Similarity Index (SSIM): 0.8701
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_1-image_proper.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: missfit_2-1.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_2-1.jpg: 0.8701
Surface Roughness (Std Dev): 20.0010
Linear Correlation (Pearson): 0.9312
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: missfit_2-2.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_2-2.jpg: 0.8701
Surface Roughness (Std Dev): 20.0010
Linear Correlation (Pearson): 0.9312
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: missfit_2-3.jpg
Structural Similarity Index (SSIM): 0.8608
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_2-3.jpg: 0.8701
Surface Roughness (Std Dev): 20.0010
Linear Correlation (Pearson): 0.9312
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: missfit_3-1.jpg
Structural Similarity Index (SSIM): 0.8703
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_3-1.jpg: 0.8703
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: missfit_3-2.jpg
Structural Similarity Index (SSIM): 0.8703
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_3-2.jpg: 0.8703
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: missfit_3-3.jpg
Structural Similarity Index (SSIM): 0.8703
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for missfit_3-3.jpg: 0.8703
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: new_magnet_proper.jpg
Structural Similarity Index (SSIM): 0.8701
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for new_magnet_proper.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: new_magnet_empty_slots.jpg
Structural Similarity Index (SSIM): 0.8597
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for new_magnet_empty_slots.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: new_magnet_proper.jpg
Structural Similarity Index (SSIM): 0.8701
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for new_magnet_proper.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```

```
Disparity Map
```

```
Dispersion Map
```

```
Processing: new_magnet_empty_slots.jpg
Structural Similarity Index (SSIM): 0.8597
Threshold for SSIM classification: 0.8701
Current SSIM Threshold for new_magnet_empty_slots.jpg: 0.8701
Surface Roughness (Std Dev): 17.8853
Linear Correlation (Pearson): 0.9376
Non-linear Correlation (Spearman): 0.7580
Magnet Insertion Percentage: 1.44%
Alignment Score: 87.01%
Decision: Assembly issue detected (low SSIM and magnet not fully inserted).
```

```
Reference Image
```

```
Test Image
```

```
Pixel-wise Differences
```

```
Feature Matches
```