# Automated Fiber Optic Inspection System

## Implementation and Advancement of DO2MR and LEI Algorithms

**Based on:** "Automated Inspection of Defects in Optical Fiber Connector End Face Using Novel Morphology Approaches"

Authors: Shuang Mei, Yudan Wang, Guojun Wen, Yang Hu Published: Sensors 2018

# Architecture

main.py

- config_loader (.json): Loads configuration parameters from a JSON file
- calibration.py: Determines the physical scale um/pixel

image_processing.py:

- Preprocessing
- Localization
- Zone Generation
- Defect Detection

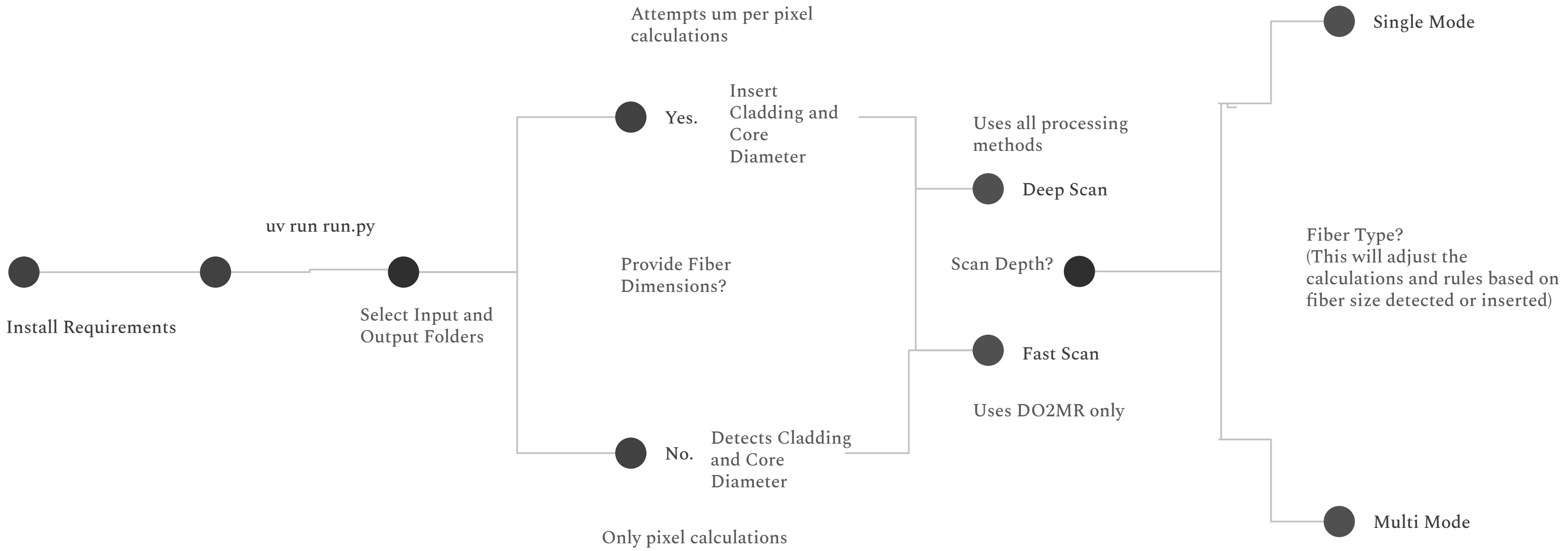
analysis.py:

- Characterization
- Classification
- Applying Pass/Fail Rules

reporting.py:

- Annotated Images
- CSV Reports
- Polar Histograms

# Initialization



# Preprocessing

**Load Image**

```
cv2.imread()
```

**Convert to Grayscale**

```
cv2.cvtColor(image,
cv2.COLOR_BGR2GRAY)
```

**Denoise:** To minimize false detections caused by noise introduced during image acquisition, apply a smoothing filter

```
cv2.GaussianBlur(gray_image,
(kernel_size, kernel_size), 0)
```

**Contrast Limited Adaptive Histogram Equalization - Correct uneven illumination**

```
clahe = cv2.createCLAHE(
   clipLimit=2.0,
   tileGridSize=(8, 8)
)
enhanced = clahe.apply(gray_image)
```

**Gaussian Blur - Reduce noise while preserving edges**

```
blurred = cv2.GaussianBlur(
   enhanced,
   ksize=(5, 5),
   sigmaX=0
)
```

# Zoning

**Automatic Circle Detection**

`cv2.HoughCircles()`

**Define and Display Core/Cladding Regions**

`cv2.circle()`

**Display Diameters:**

**Calculates the diameter (radius * 2) in pixels**

`cv2.putText()`

image_processing.py receives the preprocessed image and a mask for the specific zone being inspected (e.g., the core). It creates a working image where everything outside the current zone is blacked out.

↓

Applies medianBlur for the "Core" and a bilateralFilter for the "Cladding" to enhance the image in a way that is best suited for each region

↓

For each algorithm that runs, its output is a contribution to a confidence_map

## Hough Circle Transform

- Defects on edges
- Elliptical distortion
- Partial occlusion

```
def locate_fiber_structure(processed_image, ...):
    # Detect circles using Hough Transform
    circles = cv2.HoughCircles(
        processed_image,
        cv2.HOUGH_GRADIENT,
        dp=1.2,            # Accumulator resolution
        minDist=min_dist,  # Min distance between centers
        param1=70,         # Canny edge threshold
        param2=30,         # Accumulator threshold
        minRadius=min_r,
        maxRadius=max_r
    )

    # Validate and refine circles
    if use_circle_fit:
        # Use least-squares circle fitting
        refined_circle = circle_fit.hyper_fit(edge_points)
```

## Zone Mask Generation

```
def generate_zone_masks(shape, localization_data, ...):
    # Create distance map from center
    Y, X = np.ogrid[:height, :width]
    center_x, center_y = localization_data["cladding_center_xy"]
    dist_sq = (X - center_x)**2 + (Y - center_y)**2

    # Core Zone (0 to core radius)
    core_mask = (dist_sq < core_radius_px**2)

    # Cladding Zone (core to cladding radius)
    cladding_mask = (
        (dist_sq >= core_radius_px**2) &
        (dist_sq < cladding_radius_px**2)
    )

    # Adhesive Zone (1.0x to 1.15x cladding)
    adhesive_mask = (
        (dist_sq >= cladding_radius_px**2) &
        (dist_sq < (1.15 * cladding_radius_px)**2)
    )
```

# Processing Algorithms

do2mr:  Defects create local discontinuities that are amplified by the min-max difference (Based on research paper)

multiscale_do2mr: The DO2MR algorithm is run multiple times on scaled-down and scaled-up versions of the image. This helps detect defects of different sizes. The results are combined and add weight to the map.

morph_gradient: Calculates the morphological gradient (dilation minus erosion) of the image, which highlights edges and textures. The result is thresholded and added to the confidence map.

black_hat: Performs a black-hat transform, which reveals dark features on bright backgrounds. The result is thresholded and contributes to the map.

gabor: A bank of Gabor filters (filters for specific frequencies and orientations) is applied to the image for detecting texture anomalies. The maximum response across all filter orientations is taken and thresholded.

lbp (Local Binary Pattern): An LBP operator is applied to analyze texture. It classifies pixels based on their local neighborhood for finding texture defects that don't have sharp edges.

lei_advanced / skeletonization: Uses techniques Canny edge detection followed by "thinning" (skeletonization) to identify long, thin lines.

advanced_scratch: Uses  fusion of methods, including gradient analysis, Gabor filters, and Hessian matrix analysis (which finds "ridgelines" in the image intensity landscape) to create scratch mask.

wavelet: A Wavelet Transform decomposes the image into different frequency bands. High-frequency detail coefficients often correspond to defects. Their magnitude is calculated and thresholded.

# Region-Based Defects (DO2MR Method)

**Morphological Filtering** (**Core of DO2MR**):

- A copy of the zone image is dilated (making bright features larger) to create an I_max image.
- Another copy is eroded (making dark features larger) to create an I_min image.
- The residual image is calculated by subtracting the eroded image from the dilated image (I_residual = I_max - I_min). This dramatically highlights areas of local intensity change, which correspond to potential defects.
- The residual image is smoothed with a 3x3 Median Blur to reduce noise.

```
Min-Max Filtering
Imin(x,y) = min Is(x,y) for (x,y) in neighborhood
Imax(x,y) = max Is(x,y) for (x,y) in neighborhood
Residual Calculation
Ir(x,y) = Imax(x,y) - Imin(x,y)
```

**Sigma Thresholding**:

- The mean (μ) and standard deviation (σ) of the pixel intensities within the *active* part of the filtered residual image are calculated.
- A threshold is calculated using the formula: $T=\mu+\gamma\cdot\sigma$, where gamma is a sensitivity parameter from the config (e.g., 1.5).
- Any pixel in the residual image with a value above this threshold T is considered part of a potential defect and is set to 255 (white). All other pixels are set to 0 (black).

```
Statistical Thresholding
μ = mean(Ir)
σ = std(Ir)
Threshold = μ + γ·σ
Binary Decision
IB(x,y) = 255 if Ir(x,y) > Threshold
          0 otherwise
```

# Region-Based Defects (DO2MR Method)

Original Image    Max Filter    Min
Filter

| A | → | B | → | C |

↓ ↓

Residual = B - C

↓

Statistical Analysis

↓

Thresholding

↓

Binary Mask

# Scratch Defects (LEI Method)

**Image Enhancement:** To make low-contrast scratches more visible, the first step is to enhance the image using histogram equalization

`cv2.equalizeHist(denoised_gray_image)`

**Scratch Searching:** This process involves applying specially designed linear filters at multiple orientations (e.g., every 15 degrees) to detect scratches at any angle. Each filter application produces a "response map" where the response is high if the filter aligns with a scratch

`np.array()` to create the custom linear kernels for each orientation

`cv2.filter2D()`: Apply each kernel to the enhanced image to generate the corresponding response map

**Scratch Segmentation** a threshold to each individual response map to create a binary image highlighting potential scratch segments at that specific orientation

`cv2.threshold()`

**Result Synthesization** Combine all the individual binary scratch maps into a single, comprehensive map. The paper specifies using a logical OR operation for this synthesis

`cv2.bitwise_or(map1, map2)`

Region-Specific Analysis

`cv2.bitwise_and()` to isolate the defects that fall within each zone.

`cv2.findContours()` and `cv2.contourArea()` on the resulting images to count the number of defects and measure their features

Linear Enhancement Inspector

Scratch Strength Calculation:

$$s\theta(x,y) = 2\cdot fr\_\theta(x,y) - fg\_\theta(x,y)$$

$fr\_\theta$: Average intensity along red (center) branch

$fg\_\theta$: Average intensity along gray (parallel) branches

$\theta$: Orientation angle (0° to 180°, step 15°)

# Confidence Map

A floating-point image called confidence_map, initially all zeros, is created.

As each algorithms produces a binary mask, the system looks at the algorithm_weights.
For every pixel a given algorithm has marked as a defect, the corresponding pixel in the confidence_map is increased by that algorithm's weight.

After all algorithms have "voted" on the confidence map, a final decision is made.

1. **Adaptive Threshold**: The system calculates an adaptive_threshold_val based on the image's statistics (mean, standard deviation)
2. **Thresholding the Map**: The confidence_map is thresholded. Any pixel with a confidence score above the adaptive threshold is considered a high-confidence defect. A second, lower threshold is used to identify medium-confidence defects.
3. **Validation**: The resulting mask undergoes a final validate_defect_mask step. For each potential defect, this function analyzes its contrast against the immediate surrounding background. Defects with very low contrast are discarded as likely false positives.

# Pass/Fail Rules

After all defects have been detected and characterized, the apply_pass_fail_rules function in analysis.py is called. It then calls get_zone_definitions(fiber_type_key) to load the corresponding list of zones and their specific pass_fail_rules from the configuration file.

Rules for Single-Mode Fiber:

- Core Zone:
  - max_scratches: 0
  - max_defects: 0
  - max_defect_size_um: 3.0
- Cladding Zone:
  - max_scratches: 5
  - max_defects: 5
  - max_defect_size_um: 10.0
- Adhesive Zone:
  - max_defects: "unlimited"
  - max_defect_size_um: 50.0
- Contact Zone:
  - max_defects: "unlimited"
  - max_defect_size_um: 100.0

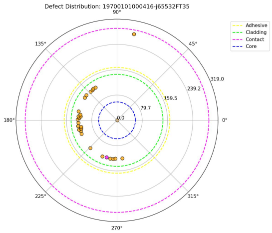
if the scratch count exceeds max_scratches it's a FAIL.

if the pit/dig count exceeds max_defects it's a FAIL

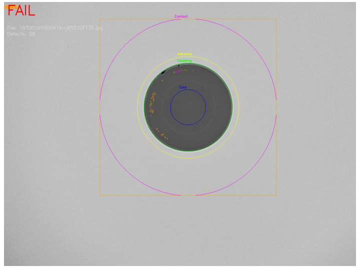
For each individual defect in the zone and checks if its size (length_um) exceeds max_defect_size_um it′'s a FAIL

Any rule violation immediately sets the overall_status to "FAIL" and appends a descriptive reason to a list.

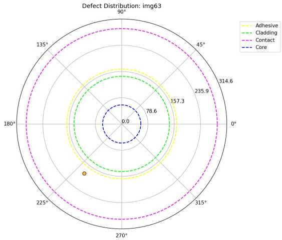
# Results



Defect Distribution: 19700101000416-J65532FT35
90°
45°
0°
315°
270°
225°
180°
135°
319.0
239.2
159.5
79.7
0.0
Adhesive
Cladding
Contact
Core

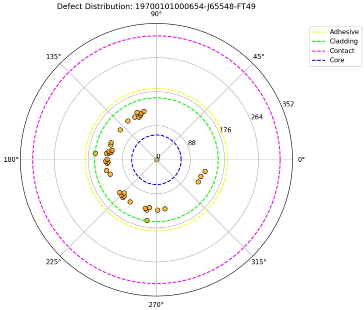



FAIL
Contact
Adhesive
Cladding
Core

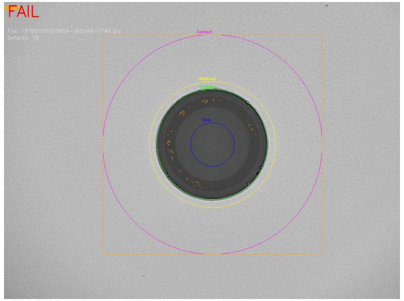



Defect Distribution: img63
90°
45°
0°
315°
270°
225°
180°
135°
314.6
235.9
157.3
78.6
0.0
Adhesive
Cladding
Contact
Core

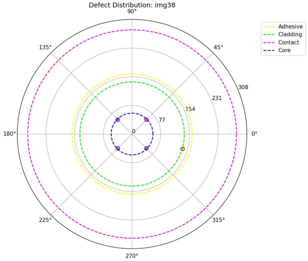

PASS
Contact
Adhesive
Cladding
Core

Defect Distribution: 19700101000654-J65548-FT49
90°
45°
0°
315°
270°
225°
180°
135°
352
264
176
88
0
Adhesive
Cladding
Contact
Core

FAIL
Contact
Core

Defect Distribution: img38
90°
45°
0°
315°
270°
225°
180°
135°
308
231
154
77
0
Adhesive
Cladding
Contact
Core


FAIL
Contact
Adhesive