

Fuzzy Inference Systems to Fine-Tune a Local Eigenvector Image Smoothing Method

1) Reference:

Almutairi, K.; Morillas, S.; Latorre-Carmona, P. Fuzzy Inference Systems to Fine-Tune a Local Eigenvector Image Smoothing Method. *Electronics* **2024**, *13*, 1150.
<https://doi.org/10.3390/electronics13061150>

2) Purpose:

This folder provides two Fuzzy Inference Systems (FIS) for image smoothing:

- **FuzzyEIG1:** Takes the absolute value of standard deviation (SD) of the noisy image as input and generates three outputs (C1, C2, C3) separately for filtering.
- **FuzzyEIG2:** Designed to smooth noisy images based on two local neighbourhood variances (d1 and d2) and generates three outputs (C1, C2, C3) separately for filtering.

3) Functionality:

Takes a noisy image (`ImageR`) as input and returns the smoothed image (`FilteredI`).

4) Parameters:

- **Window size** (integer, e.g., Delta =1 for 3x3)
- **D:** Control parameter for the local eigenvector filter (recommended: $(5/6) * \text{noise level}$ for window size 3x3)
- **FLSParams:** Membership function parameters for the FIS (optimised for noise levels 10, 20, 30) - provided in `SmoothingParams.mat`. Where you can find the smoothing parameters for each noise level.
- **SmoothingCoeffsMatrixNC*:** Pre-computed smoothing coefficients for specific noise levels.

5) Example Usage:

% Inputs:

* ImageR: Noisy image
* ImageO: Original image (used for PSNR calculation)
* Delta: Size of the filtering window (use 1 for 3x3).
* D: Control parameter for filter use $((5/6)*NoiseLevel)$ for the 3x3 window.
* N(noiselevel)C*: Membership function parameters (from SmoothingParams.mat).
In the SmoothingParams.mat, you can find the parameters N10, N20, and N30, where from noise 10, 20, and 30, respectively.

% Outputs:

* FilteredI: Smoothed image.

Please open the function called FuzzyEIG1.m to use the first method and FuzzyEIG2.mat for the second system.

1: FuzzyEIG1:

```
load('ImagesFIS.mat')
load('SmoothingParamsC1.mat')
load('SmoothingParamsC2.mat')
load('SmoothingParamsC2.mat')
```

```
FilteredI = FuzzyEIG1(ImageR, ImageO, N*noiselevelC1, N*noiselevel C2, N*noiselevel C3, 1, (5/6)*noiselevel)
```

2: FuzzyEIG2:

```
load('ImagesFIS.mat')
load('SmoothingParamsC1.mat')
load('SmoothingParamsC2.mat')
load('SmoothingParamsC2.mat')
```

```
FilteredI = FuzzyEIG2(ImageR, ImageO, N*noiselevel C1, N*noiselevel C2, N*noiselevel C3, 1, (5/6)*noiselevel)
```

Notes:

- The code provides optimised parameters for noise levels 10, 20, and 30. For intermediate noise levels, consider averaging or alternative methods, such as linear interpolation, for any noise level using corresponding parameters from the provided files.
- Refer to SmoothingParams.mat and other .mat files for specific pre-computed coefficients.