**Project Report: Adaptive Histogram Equalization GUI in MATLAB**

**1. Introduction**

Image enhancement is a fundamental step in image processing that aims to improve the visual appearance of an image or to convert the image to a form better suited for analysis. Histogram equalization is a common technique that enhances the contrast of an image by spreading out the most frequent intensity values.

This project focuses on **Adaptive Histogram Equalization (AHE)** using a **MATLAB GUI**. Unlike global histogram equalization, AHE works on small regions of the image (tiles), making it suitable for enhancing local contrast, especially in images with varying illumination. This project allows users to interactively load an image and apply local histogram equalization using window sizes of **8×8**, **16×16**, or **32×32**, visualizing both the original and enhanced results along with their histograms.

**2. Objective**

* To design a GUI in MATLAB for visualizing adaptive histogram equalization.
* To enable selection of window sizes dynamically using a slider.
* To enhance image contrast in a locally adaptive manner.
* To compare the original and enhanced histograms side-by-side.

**3. Theoretical Background**

**3.1 Histogram Equalization**

Histogram equalization aims to redistribute the intensity values of pixels so that they span a wider range, enhancing global contrast. However, global histogram equalization may not work well on images with local brightness variations.

**3.2 Adaptive Histogram Equalization (AHE)**

AHE divides the image into non-overlapping tiles (e.g., 8x8, 16x16) and applies histogram equalization to each tile individually. This boosts local contrast in areas of the image and is particularly helpful when the image contains both dark and bright regions.

**3.3 Clip Limit (CLAHE)**

A variant of AHE is **Contrast Limited AHE (CLAHE)**, which avoids amplifying noise by clipping the histogram before equalization. This project uses CLAHE through MATLAB’s adapthisteq function.

**4. GUI Design in MATLAB**

The GUI was created using MATLAB's uicontrol and axes functions. It consists of:

* **Four Display Axes**:
  + Original Image
  + Original Histogram
  + Enhanced Image
  + Enhanced Histogram
* **Slider**:
  + Allows selection of window size: 8x8, 16x16, or 32x32.
* **Button**:
  + Used to load an input image from disk.
* **Text Label**:
  + Displays the current window size selected via the slider.

**5. Code Explanation**

**5.1 Main Function**

function trialGUI

This defines the main function that creates the GUI window and its components.

**5.2 Axes and Controls**

ax1 = axes(...); % Original image

ax2 = axes(...); % Original histogram

ax3 = axes(...); % Enhanced image

ax4 = axes(...); % Enhanced histogram

These define the display regions for images and histograms.

btn = uicontrol('Style','pushbutton', ...);

sld = uicontrol('Style','slider', ...);

lbl = uicontrol('Style','text', ...);

These create the button, slider, and label UI elements.

**5.3 loadImage Callback**

Triggered when the user clicks "Input Image". It:

* Loads the image and converts it to grayscale if necessary.
* Displays the original image and its histogram.
* Calls sliderChanged to perform enhancement.

**5.4 sliderChanged Callback**

Called when the slider is moved. It:

* Reads current slider value to determine tile size.
* Computes number of tiles as [floor(rows/win), floor(cols/win)].
* Applies adapthisteq with the selected parameters.
* Updates the enhanced image and histogram displays.

enhanced = adapthisteq(data.img, 'NumTiles', numTiles, 'ClipLimit', 0.01);

This is the key line where adaptive histogram equalization is applied.

**6. Visual Behavior of Window Sizes**

* **8×8**: Enhances very local contrast. Might produce grainy textures due to over-enhancement of noise.
* **16×16**: Balanced enhancement — good compromise between global and local effects.
* **32×32**: Enhances broader regions. May underperform in very dark or bright small regions.

**7. Improvements Added**

* **Presmoothing (Optional)**: Before histogram equalization, Gaussian blur or other filters can be applied to reduce noise.
* **ClipLimit Tuning**: User can experiment with ClipLimit to avoid graininess. Smaller values reduce contrast but prevent noise boosting.

**8. Results**

The GUI performs successfully:

* Loads and displays grayscale images.
* Shows original and enhanced histograms.
* Responds to user interaction with the slider.
* Provides real-time visual feedback of how tile size affects local contrast.

**9. Conclusion**

This project successfully implements a GUI for adaptive histogram equalization in MATLAB. It demonstrates the power of localized contrast enhancement and gives the user control over the granularity of enhancement. The project also serves as a hands-on way to understand how tile size and contrast limits influence the visual outcome.

**10. Future Work**

* Allow runtime adjustment of ClipLimit.
* Add option for color image enhancement (channel-wise).
* Include histogram normalization or log-scale display.