**Homework\_2 Image Processing**

*Group Partner: Abdul Malik (2211011098)*

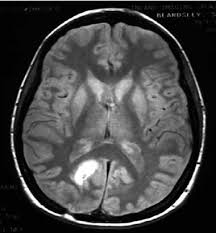
*Submitted by: Abdullah Rihawi (2211011093)*

Figure 1 The Image Used in Homework

**Histogram Equalization:**

Image processing uses Histogram Equalization (HE) to improve contrast by modifying pixel intensity distributions. The method redistributes pixel brightness values across the entire intensity spectrum to display more image details in dark or bright regions.

The image details become more visible through Histogram equalization because this technique modifies contrast distribution to improve scene visibility in scenes with low illumination or intense glare conditions. Medical imaging involves widespread use of this technique which helps show significant structures not visible without such enhancement. Remote sensing technology along with satellite imagery benefits from histogram equalization because it enhances visibility of terrain details and weather patterns. The method proves essential for biometric systems which depend on fingerprint recognition because it effectively enhances the visibility of ridge details. Image clarity and interpretability in different domains improve because pixel intensities reach their complete dynamic range during stretching processes.

A close-up of a brain scan

AI-generated content may be incorrect.A close-up of a computer screen

AI-generated content may be incorrect.Code Covering Basic Histogram Equalization:

Explanation:

The given code consists of a function that takes image\_path as a parameter, this parameter is set to make it possible to read the image using the imread() function. Then, the grasped picture is ensured that it is in black and white format (grey) using the tgb2gray() function. Now, we are ready to apply Histogram Equalization, so we use the histeq() function and we plot the results.

**Local Histogram Equalization:**

Local Histogram Equalization serves as a method in image processing that utilizes contrast-enhancing histogram equalization to individual regions of an image instead of implementing it across the whole image area simultaneously. The method adapts contrast enhancement based on various image sections which makes it efficient for images containing diverse lighting areas. The image enhancement provided by local histogram equalization occurs at the level of small image areas rather than through system-wide operations. This method proves most effective when applied to images with inconsistent lighting because it corrects both excessively bright and excessively dark regions.

The approach implements histogram equalization on individual tiles which divides the image into non-overlapping parts. The method finds extensive application in medical imaging for diagnostic improvement and for surveillance systems that need enhanced image visibility under diverse lighting conditions. Maintaining local particularities stops image enhancement from getting excessive while simultaneously improving the contrast adjustment of multilevel visual content.

Code Covering Local Histogram Equalization:

**A screenshot of a computer

AI-generated content may be incorrect.**

**A close-up of a brain scan

AI-generated content may be incorrect.**

Explanation

The local\_hist\_eq function implements Adaptive Histogram Equalization (AHE) for Local Histogram Equalization of images. The method strengthens the image contrasts in limited areas instead of performing a universal histogram adjustment

**Adaptive Histogram Equalization:**

The Adaptive Histogram Equalization (AHE) functions as an image processing solution that enhances contrast by creating dedicated histograms for image subsections before adjusting their lightness values through targeted distribution. The local operations of the AHE contrast enhancement technique differ from standard histogram equalization global methods by enabling better contrast detection in areas with diverse illumination effects. A local processing method provides exceptional advantages for images that display mixed lighting levels since it discloses hidden details in dim and bright areas. The essential advantage becomes evident because this method strengthens edge definitions within different image regions.

Code Covering Adaptive Histogram Equalization

A screenshot of a computer program

AI-generated content may be incorrect.

A close-up of a brain scan

AI-generated content may be incorrect.

Explanation:

The adaptive\_hist\_eq function processes a specified image path to read an image before converting its color mode to grayscale then it applies adaptive histogram equalization through adaptive\_hist\_eq to improve local contrast which enables visual comparison between original grayscale and AHE-enhanced imagery in a figure window.

**Contrast-Limited Adaptive Histogram Equalization (CLAHE):**

CLAHE operates as an image enhancement method that divides input pictures into blocks to perform local contrast adjustments with noise control through restricted histogram equalization parameters followed by tile blending. CLAHE technology strengthens both low-contrast and high-contrast image features in medical imaging while retaining the clarity of details along with low-noise performance across different lighting conditions. This technique proves optimal for medical imaging and low-light photography because it avoids the noisy artifacts that harm contrast enhancement using conventional adaptive methods.

Code Covering CLAHE

A close-up of a computer code

AI-generated content may be incorrect.

A close-up of a brain scan

AI-generated content may be incorrect.

Explanation:

The clahe\_hist\_eq MATLAB function processes image file paths that read them before converting them to grayscale and applying Contrast Limited Adaptive Histogram Equalization (CLAHE) with a low clip limit of 0.01 for contrast enhancement and noise suppression. The function presents both images in a figure that shows the original grayscale version and the CLAHE-enhanced version next to each other for comparative viewing.

**Comparisons of Histogram Equalization Techniques**

Basic Histogram Equalization (HE) works across the entire image through a process of evenly spacing pixel values. The global method remains simple to execute thus it proves useful for images with restricted dynamic range values. The main drawback of this method includes both the increased noise levels in homogenous sections and its inability to handle localized illumination variations. Such uneven enhancement in the image results from over-processing along with areas that remain under-adjusted.

AHE addresses these problems through a-dividing-image-into-tiles technique before performing histogram equalization operations individually within each tile. Within AHE there exists an inherent issue that produces large-scale noise boost generation which causes a fake appearance of square elements in concentrated image sections.

The CLAHE method extends AHE through the addition of clipping limitations to the histogram process that determines the cumulative distribution function. The clipping mechanism operates to decrease the amount of contrast enhancement that occurs in homogeneous image regions thus controlling noise amplification levels. USA provides an optimal trade-off of image contrast elevation with reduced artifacts thus serving regressively important in systems which require both detail maintenance and artifact minimization. The visual analysis exhibits HE delivers a basic contrast increase though it results in noise artifacts and AHE enhances local contrast while creating visible noise and CLAHE delivers the best combination of local detail optimization with minimized noise leading to an improved image quality.

A close-up of a brain scan

AI-generated content may be incorrect.

**Convolution and Correlation Applications**

The processing of images significantly depends on convolution and correlation as they bring separate functional qualities to each operation. The flipping operation of the kernel enables convolution to achieve tasks such as blurring and sharpening in addition to edge detection. The choice of kernel structure enables us to adjust image frequencies allowing either reduction of noise or accentuation of edges. Advancements in complex tasks such as image pre-processing depend on convolution but also rely on this technique for feature extraction in neural networks and the application of filters to modify image attributes.

The pattern recognition and template matching functions are where correlation finds its strength. The ability to detect specific image patterns across entire visuals allows this operation to support object detection and motion analysis tasks. The exact matches between template patterns and imaging elements become possible through template comparison which succeeds in identifying occurrences even within distorted or rotated patterns. The ability discovered through this method remains effective with signals in addition to images which enable signal processing operations that detect signal similarities.

The sliding kernel process for images uses different application rules to generate contrasting results. Scientists choose convolution over correlation based on real-time needs since convolution excels in blocking tasks and extracting features, but correlation better detect templates along with distinctive patterns.

The process of sliding kernel operation between convolution and correlation both starts identically but diverges because convolution flips its kernel before movement occurs. The kernel flipping operation that occurs during convolution makes this tool the best choice for linear filter work such as image blurring because it maintains shift-invariant qualities needed in CNNs while deriving derivative functions. In correlation operations, the flipping step is omitted because it directly locates kernel instances within images which leads to superior template-matching capabilities as well as pattern recognition and image registration functions. Each operation produces different outcomes due to the flipping step leading convolution to be commutative but correlation does not make them suitable for different image processing needs.

Code Covering Convolution and Correlation

A close-up of a brain scan

AI-generated content may be incorrect.A close-up of several images of a brain

AI-generated content may be incorrect.A close-up of a brain scan

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.

Explanation

The function convolution\_correlation starts its operation by receiving image file path information together with kernel matrix data. The code compares the kernel with pre-defined collections of sharpening, edge detection and blurring kernels to determine its purpose before generating an appropriate kernel\_name. After image reading and grayscale conversion, the system first processes image convolution with conv2 before executing filter2 correlation to preserve input dimensions using the 'same' option. The function produces three subplots in the figure which include the original grayscale image along with its convolution result converted to uint8 display and the correlation result adjusted to uint8 visualization.

**Border Padding with Different Methods:**

Border padding in image processing addresses the issue of kernel operations extending beyond image boundaries by adding virtual pixels, using methods like zero, replicate, or symmetric padding, to maintain image dimensions and prevent boundary artifacts; this is crucial for accurate convolution and correlation, ensuring that edge pixels are processed correctly, information isn't lost, and features near the image edges are accurately detected, ultimately improving the overall quality and reliability of image processing results.

A close-up of a screen

AI-generated content may be incorrect.Code Covering Border Padding

A close-up of a brain mri

AI-generated content may be incorrect.

Explanation

The border\_padding function demonstrates three common border padding techniques applied to a grayscale image. It takes an image\_path as input, reads the image, and converts it to grayscale. Then, it uses the padarray function to create three padded versions of the image: padded\_zero (zero padding), padded\_replicate (replicate padding), and padded\_symmetric (symmetric padding), each extending the image by 10 pixels on all sides. Finally, it creates a figure with four subplots, displaying the original grayscale image and the three padded versions side-by-side, each with a descriptive title indicating the padding method used. This allows for a visual comparison of the effects of different border padding techniques on the image.