

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
|  | **Image Enhancement**  **using CLAHE** | | | |  |
|  |  | | | |  |
|  | |  |  | | |
|  | | **Suraksha Kotte 6996656** |  | | |
|  | | Submission date: 02/11/2023  —  ECTE 401  Multimedia signal processing  —  Image Processing Project |  | | |
|  | |  | |  | |

Table of Contents

[1. ABSTRACT 3](#_Toc149829809)

[2. INTRODUCTION 3](#_Toc149829810)

[3. METHODOLOGY 4](#_Toc149829811)

[3.1 System Block Diagram: 4](#_Toc149829812)

[3.2 Algorithms Used: 4](#_Toc149829813)

[4. RESULTS & DISCUSSION 5](#_Toc149829814)

[Test 1: 5](#_Toc149829815)

[Test 2: 5](#_Toc149829816)

[5. REFERENCES 6](#_Toc149829817)

[5.1 REPORT: 6](#_Toc149829818)

[5.2 ‌ CODE: 6](#_Toc149829819)

[6. APPENDIX 7](#_Toc149829820)

# ABSTRACT

The enhancement of images is essential in a variety of applications, including medical imaging and satellite analysis. This project investigates Contrast Limited Adaptive Histogram Equalization (CLAHE), a complex contrast enhancement approach, in MATLAB. CLAHE, in contrast to traditional Histogram Equalization techniques, provides adaptive contrast enhancement by breaking the picture into smaller overlapping parts. This method maintains fine details while maintaining overall picture coherence. Grayscale conversion precedes the application of CLAHE and contrast stretching methods during the execution of CLAHE. This project shows the usefulness of CLAHE in improving X-ray pictures using COVID-19 X-ray images as a dataset. The results demonstrate considerable increases in contrast and lighting, allowing for greater visibility of minute details.

# INTRODUCTION

The subject of image enhancement is critical in the field of digital image processing, since it is used in a variety of applications such as medical imaging and satellite analysis. This project utilizes MATLAB to build and analyze Contrast Limited Adaptive Histogram Equalization (CLAHE). CLAHE, in contrast to typical Histogram Equalization approaches, provides a specialized and adaptable approach to contrast enhancement, making it an important field of research.

The purpose of this project is to investigate the complexity of CLAHE, highlighting its efficiency in comparison to standard histogram equalization methods and other current algorithms.

While beneficial, traditional histogram equalization in MATLAB has limits when applied evenly to an entire picture. Its failure to adjust to changes in local contrast frequently results in the augmentation of unwanted noise. CLAHE, which is implemented in MATLAB, addresses this difficulty by splitting the picture into smaller, overlaying sections. CLAHE enables adaptive contrast enhancement customized to various sections of the picture by separately computing and dispersing the histogram of each region. This method retains delicate details as well as overall image coherence. [2]

Additionally, CLAHE introduces an important idea - contrast limitation. This function in MATLAB serves as a precaution, avoiding noise amplification in identical zones. CLAHE achieves a compromise between enhancing the picture's clarity and retaining its initial image information by restricting contrast enhancement. [3]

CLAHE stands out in the framework of MATLAB image processing because of its versatility. The adaptability of MATLAB enables accurate modification of CLAHE settings, making it an excellent solution for photos with fluctuating lighting circumstances, complicated surfaces, and contrasts. This project goes into the CLAHE implementation using MATLAB, with the goal of demonstrating not only its technology but also its operational usefulness in real-world circumstances. [6]

# METHODOLOGY

**Dataset used:** COVID-19 X-ray images. [1]

**Application:** MATLAB R2020b

**Main Algorithm used:** Contrast Limited Adaptive Histogram Equalization (CLAHE) for enhancing the X-ray images.

### System Block Diagram:

Enhanced X-ray image

Image Preprocessing & Enhancement Algorithm

Original X-ray image

**Original X-ray Image:** This is the input X-ray picture from the dataset.

**Image Preprocessing & Enhancement algorithm:** This method includes several preprocessing processes as well as enhancement algorithms. Grayscale conversion, CLAHE (Contrast Limited Adaptive Histogram Equalization), and contrast stretching are the approaches. The enhancement method increases picture quality after preprocessing procedures to make certain the image is in a proper format for enhancement.

**Enhanced X-ray Image:** This is the final picture once the enhancement technique has been applied.

### Algorithms Used:

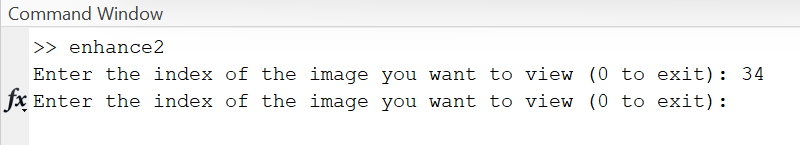
**Grayscale Conversion:** If the original X-ray picture is in color, it is converted to grayscale to ease processing afterwards. This step guarantees that the picture interpretation is consistent.

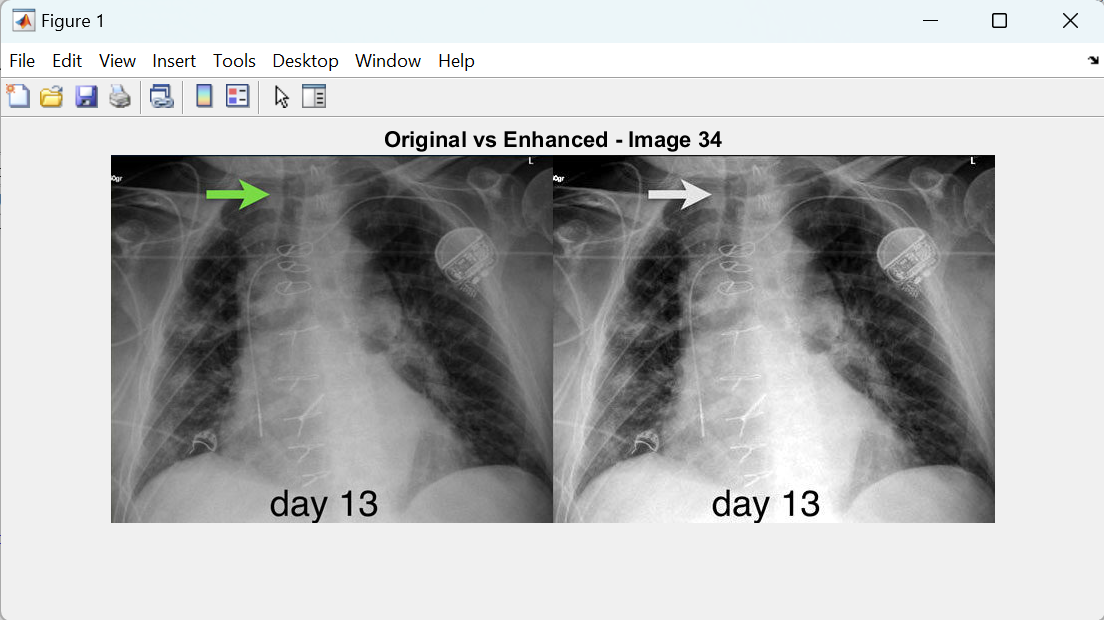
**CLAHE (Contrast Limited Adaptive Histogram Equalization):** It works on distinct sections of the image, changing the contrast to the unique properties of each. CLAHE reduces over-amplification of noise by reducing contrast enhancement, making it particularly useful for medical imaging. [4]

**Contrast Stretching:** Contrast stretching improves the image's contrast. It translates the values of pixels to a new range, essentially extending the intensity value range. This phase is used depending on the improvement requirements. [5]

# RESULTS & DISCUSSION

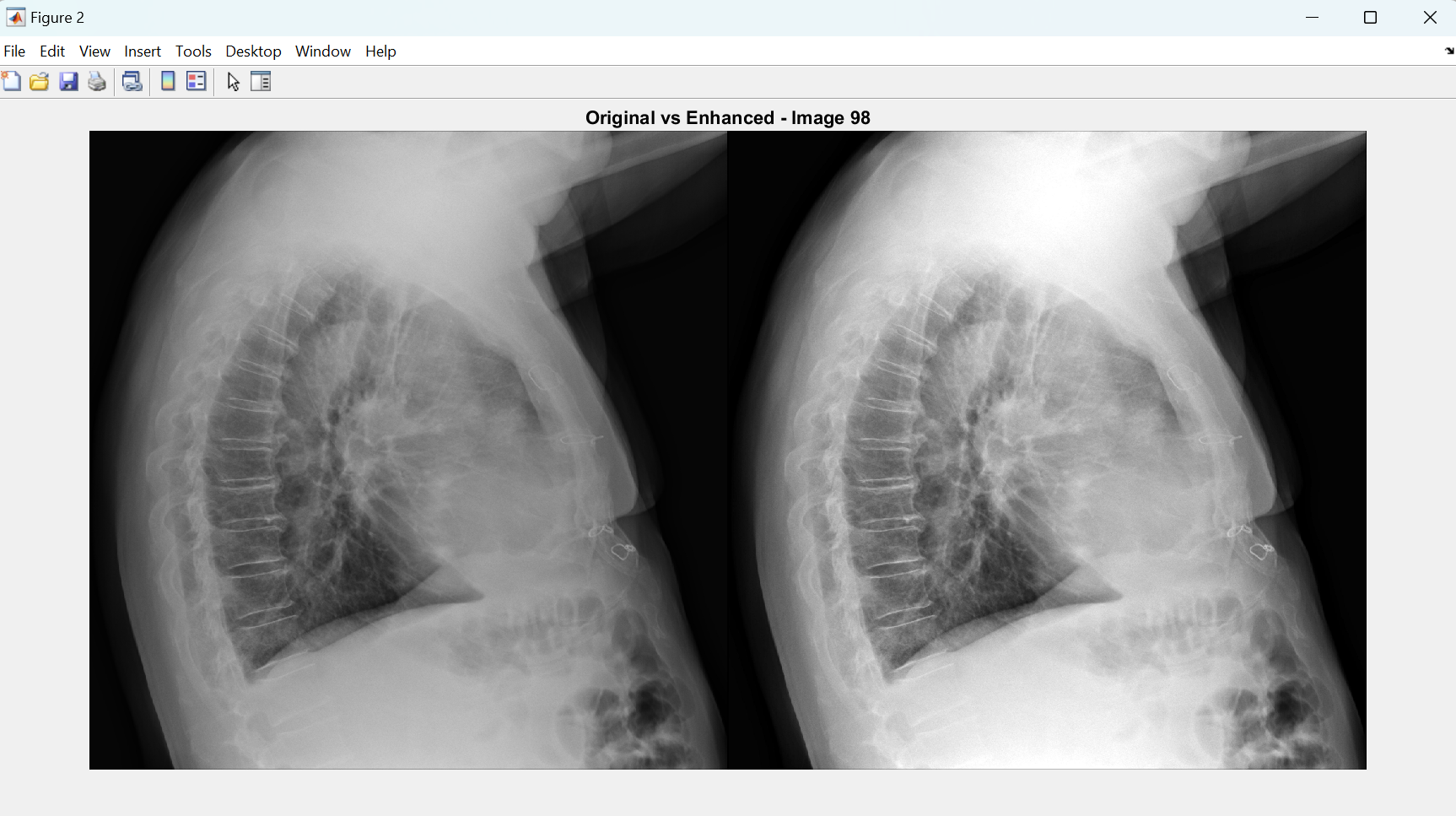
### Test 1:





### Test 2:





Even though the X-ray input images were already grayscale, grayscale conversion was performed for consistency. The Contrast Limited Adaptive Histogram Equalization (CLAHE) technique boosted contrast and illumination dramatically, highlighting smaller details. This improvement aided medical evaluations by allowing for more accurate detection of anomalies. The contrast stretching improved image clarity even further. The method enhanced X-ray picture quality, allowing for more efficient detection.

# REFERENCES

### REPORT:

[1]

bachir, “COVID-19 chest xray,” *Kaggle.com*, 2020. https://www.kaggle.com/datasets/bachrr/covid-chest-xray?select=images (accessed Oct. 30, 2023).

‌ [2]

O. G. S, S. V. V, A. Zh. Zh, B. S, O. A. B, and N. O, “Application of the Clahe Method Contrast Enhancement of X-Ray Images,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022, doi: https://doi.org/10.14569/ijacsa.2022.0130549.

[3]

“Exact histogram equalization and specification - File Exchange - MATLAB CentralFile Exchange - MATLAB Central,” *Mathworks.com*, Dec. 11, 2021. https://ch.mathworks.com/matlabcentral/fileexchange/26309-exact-histogram-equalization-and-specification.

‌[4]

Parthiban Marimuthu, “Image Contrast Enhancement Using CLAHE,” *Analytics Vidhya*, Aug. 17, 2022. https://www.analyticsvidhya.com/blog/2022/08/image-contrast-enhancement-using-clahe/#:~:text=CLAHE%20%E2%80%93%20Contrast%20Limited%20Adaptive%20Histogram,that%20prevents%20contrast%20over%2Damplification.

‌[5]

“Contrast Stretching in Image Processing using Matlab,” *Nuruzzaman Faruqui*, Feb. 10, 2019. https://www.nzfaruqui.com/contrast-stretching-in-image-processing-using-matlab/#:~:text=Contrast%20stretching%20is%20also%20known,is%20about%20to%20be%20normalized.

[6]

“Contrast Enhancement of MRI Images using AHE and CLAHE Techniques,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 2, pp. 2442–2445, Dec. 2019, doi: https://doi.org/10.35940/ijitee.b7017.129219.

### ‌ CODE:

“Apply Contrast-Limited Adaptive Histogram Equalization (CLAHE),” *Mathworks.com*, 2023. https://ch.mathworks.com/help/images/ref/adapthisteq.html (accessed Oct. 30, 2023).

“Find Limits to Stretch Contrast in Grayscale Image,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/images/ref/stretchlim.html (accessed Oct. 30, 2023).

“ExampleTopics,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/images/contrast-adjustment.html#buh9ylp-59 (accessed Oct. 30, 2023).

“Image Enhancement by Histogram Equalization,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/hdlcoder/ug/image-enhancement-by-histogram-equalization.html (accessed Oct. 30, 2023).

“Decode JSON Text,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/matlab/ref/jsondecode.html (accessed Oct. 30, 2023).

“VisibleBreadcrumbs,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/matlab/ref/dir.html?searchHighlight=dir%20&s\_tid=srchtitle\_support\_results\_1\_dir%20 (accessed Oct. 30, 2023).

“Create a Full File Path on Windows,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/matlab/ref/fullfile.html?searchHighlight=fullfile&s\_tid=srchtitle\_support\_results\_1\_fullfile (accessed Oct. 30, 2023).

“Search File for Text,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/matlab/ref/fileread.html?searchHighlight=fileread&s\_tid=srchtitle\_support\_results\_1\_fileread (accessed Oct. 30, 2023).

“Determine If Arrays Are Cell Arrays,” *Mathworks.com*, 2023. https://ch.mathworks.com/help/matlab/ref/iscell.html?searchHighlight=iscell&s\_tid=srchtitle\_support\_results\_1\_iscell (accessed Oct. 30, 2023)

# APPENDIX

% Path to the chest X-ray image folders

imageFolderPath = "C:\Users\surak\Downloads\xray\images";

annotationsPath = "C:\Users\surak\Downloads\xray\annotations\imageannotation\_ai\_lung\_bounding\_boxes.json";

% Path for the enhanced images folder

enhancedImageFolderPath = 'C:\Users\surak\Downloads\xray\enhanced images';

% Load JSON annotations

annotateData = jsondecode(fileread(annotationsPath));

% To check if annotateData is a cell array

if iscell(annotateData)

annotateData = annotateData{1}; % extract the first element

end

List of all image files from imagePath

imageFiles = dir(fullfile(imageFolderPath, '\*.jpg'));

% Parameters

clipLimit = 0.8; % Clip limit for CLAHE (from 0 to 1)

stretchlimLimits = [0.01 0.99]; % Lower and upper percentile values for stretch limits

contrastStretching = true; % Assign contrast stretching as true

% Loop through each image and apply enhancement based on annotations

for i = 1:length(imageFiles)

% Read the image

imagePath = fullfile(imageFolderPath, imageFiles(i).name);

ogImg = imread(imagePath);

% Convert the image to grayscale if it's in color

if size(ogImg, 3) == 3 % for color, it is 3 color channels

grayImg = rgb2gray(ogImg);

else

grayImg = ogImg; % image is grayscale

end

% Extract bounding box coordinates from annotations

try

annotations = annotateData.annotations{i}; % Access annotations for the current image index

catch

% If an error occurs, skip this image

continue; % continue to the next one

end

% Apply CLAHE to the regions inside the bounding boxes

enhancedImg = grayImg;

% Loop through each bounding box in annotations and apply enhancement

for j = 1:length(annotations)

try

x = annotations{j}.bbox(1); % x-coordinate of the bounding box from annotations

y = annotations{j}.bbox(2); % y-coordinate of the bounding box from annotations

width = annotations{j}.bbox(3); % Width of the bounding box from annotations

height = annotations{j}.bbox(4); % Height of the bounding box from annotations

% Apply CLAHE to the region inside the bounding box

r = enhancedImg(y: y + height - 1, x: x + width - 1);

r = adapthisteq(r, 'ClipLimit', clipLimit, 'Distrubtion', 'rayleigh');

enhancedImg(y: y + height - 1, x: x + width - 1) = r; % replaces with enhanced image

catch

% If an error occurs, skip and continue to the next one

continue;

end

end

% if enabled

if contrastStretching

% stretch limits (percentiles)

stretchLimits = stretchlim(enhancedImg, stretchlimLimits);

% contrast stretching

enhancedImg = imadjust(enhancedImg, stretchLimits, []);

end

% % Display and save the enhanced image

% figure;

% imshowpair(grayImg, enhancedImg, 'montage'); % side by side

% title(['Original vs Enhanced - Image ' num2str(i)]); % displaying the title with image number

%

% Save the output images to the "enhanced images" folder

enhancedImgPath = fullfile(enhancedImageFolderPath, ['enhanced\_' imageFiles(i).name]);

imwrite(enhancedImg, enhancedImgPath);

end

while(1)

i = input('Enter the index of the image you want to view (0 to exit): ');

if i == 0

break; % exit the loop if user enters 0

end

if i >= 1 && i <= length(imageFiles)

% Read the original image

imagePath = fullfile(imageFolderPath, imageFiles(i).name);

ogImg = imread(imagePath);

% Read the enhanced image

enhancedImagePath = fullfile(enhancedImageFolderPath, ['enhanced\_' imageFiles(i).name]);

enhancedImg = imread(enhancedImagePath);

% Display the original and enhanced images using imshowpair

figure;

imshowpair(ogImg, enhancedImg, 'montage');

title(['Original vs Enhanced - Image ' num2str(i)]); % displaying the title with image number

else

disp('WRONG NUMBER!! Please enter a valid index :)');

end

end