Restoration of Damaged Images

ECN-316 DIGITAL IMAGE PROCESSING

K. Gowtham: 22116045
karri_ssgr@ece.iitr.ac.in
ECE, Indian Institute of Technology, Roorkee, India

S V S Vivek: 22116089
sugamanchi_vsv@ece.iitr.ac.in
ECE, Indian Institute of Technology, Roorkee, India

K. Tharun: 22116042
kaipa_tkr@ece.iitr.ac.in
ECE, Indian Institute of Technology, Roorkee, India

Abstract—We need to preserve old historical records and old memories because as Time passes everything will start to degrade and images are no exception. Commonly images degrade by undergoing through stains, folds, scratches or fading. This project focuses on restoring damaged images by removing stains and folds and enhancing visual quality without using deep learning methods. Initial preprocessing includes noise removal using a median filter to minimize salt-and-pepper noise. Restoration techniques for stain and fold removal are applied next. As a final step, we enhance the image various process like histogram equalization for improving the contrast, Unsharp masking for improving the edges and highlighting details and brightness and gamma correction to manage the visibility of the image. This report contains details of how we made our application and how to use it to restore damaged images.

I. INTRODUCTION

Old images often undergo various forms of degradation like noise, stains, folds, scratches or fading. These issues impact visual quality and limit their usability in required tasks. The primary goal of this project is to restore and enhance such images using traditional image processing techniques we learnt in our course and tried make our own customs methods, avoiding computationally intensive deep learning methods.

To achieve our goal, we have made a python GUI based application using multiple methods from python libraries, TKinter for GUI, NumPy for array computations, matplotlib to plot histograms and OpenCV to directly apply some of our image processing techniques and to process and display our images. Our UI after uploading the image by clicking on the upload image button and selecting the required image looks like the following Figure. To open the application, we run the main.py file in our project folder.

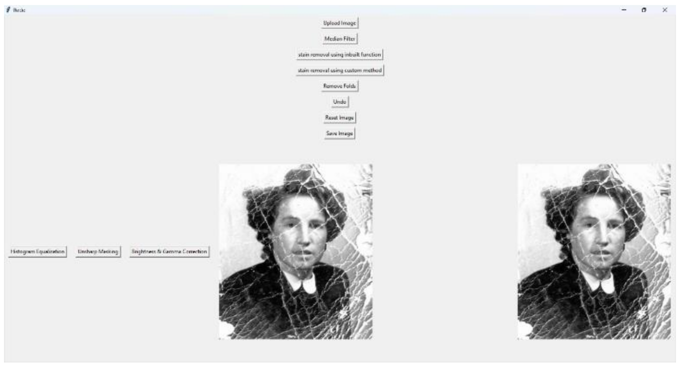


Fig. 1 UI of the main application after uploading image

The image on the relative left side is the original image initially uploaded and as we apply the operations by clicking on the buttons, the image on the right side (processed image) changes accordingly. We can reset the processed image to the original image by clicking on the reset image button and undo the operation done last by clicking on the undo button. We do the operations by clicking on the required buttons according to our needs. (median filter, stain removal using inbuilt function, stain removal using custom method, remove folds for noise removal and image restoration, histogram equalization, unsharp masking and brightness and gamma correction for image enhancement). Each operation will be explained in the methodology section of this report. Finally, after we get our desired result, we can save the image to our required location on our system by clicking on the save image button.

As we made the application only for grayscale images, we first converted the image into grayscale as soon as it is uploaded and apply any further operations on the grayscale image.

II. Methodology

Let us take 2 images, one degraded by stains and another with folds or scratches as shown in fig. 2 and fig. 3 respectively**.**

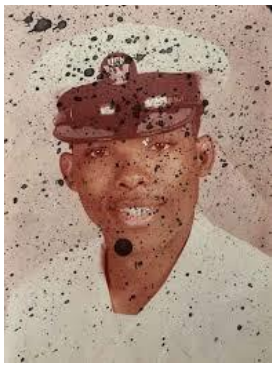


Fig. 2 Stained Image

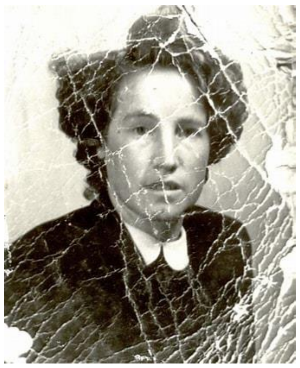


Fig. 3 Folded Image

Source: https://in.pinterest.com/rootschat/

Step – 1: Noise Removal

To eliminate salt-and-pepper noise introduced by age and storage conditions, we have used a median filter.

A median filter is applied as it is great in preserving edges while removing noise. In this filter, we take the kernel on each pixel and find the median intensity value after applying that kernel and assign the median intensity level to that pixel.

After applying noise removal for fig. 2 by median filter the output image we get is shown in fig. 4**.**

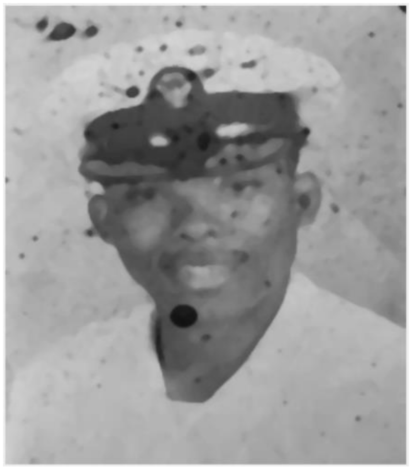


Fig. 4 Median filter output image

Step – 2: Stain and Folds Removal

A. Stain Removal

In the context of image restoration, stains refer to irregular discolorations or spots on an image caused by aging, exposure to environmental factors, or physical damage. These artifacts distort the original appearance of the image and are typically considered unwanted noise.

The image after removing noise is given and the user is prompted to paint on the image where we need apply the operation to remove stains. A mask will be generated using this by assigning 1 to all the pixels which were painted, and the remained pixels will be 0.

Now the pixels where 0 is assigned are filled with the average value of surrounding pixels by using the inbuilt function inpaint TELEA. The output after inpainting is shown in fig. 7**.**

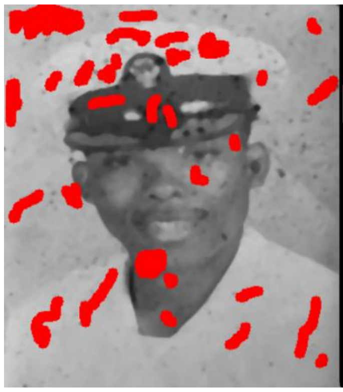


Fig. 5 Mask drawn by user

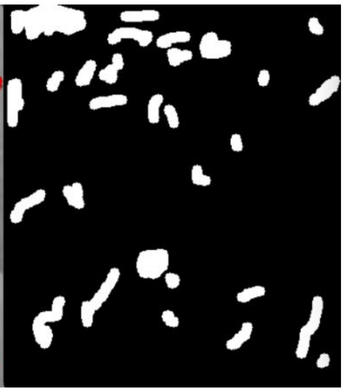


Fig. 6 Generated Binary Mask

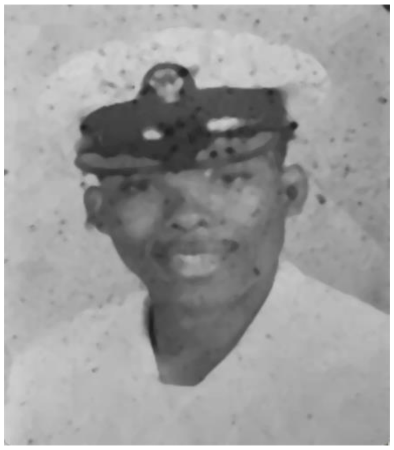


Fig. 7 Output Image after inbuilt inpaint

We made another two function to fill the masked regions where we used a morphological operation called dilation. We dilated the masked image and subtracted the masked image from the dilated image to get only the mask of the boundary pixels. Then we filled the area of the mask regions by measuring the distance from the boundary pixel to each pixel and then assigned the intensity of the nearest boundary pixel to that pixel. The other function(binary search fill) is to take the average of the radial boundary pixels and assign it to the middle pixel and average the middle and boundary to 3/4th and 1/4th pixel in a form of binary search to get a gradient filling.

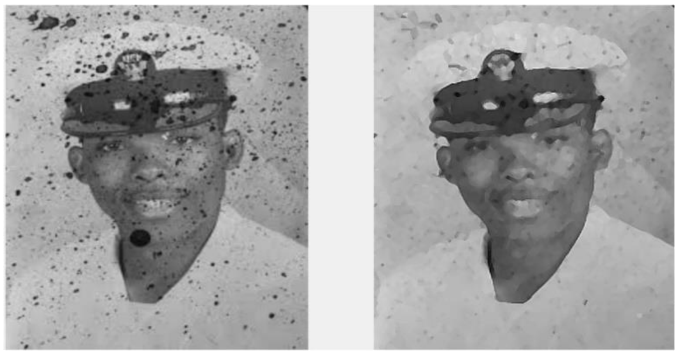


Fig. 8 Custom method Input and Output Images

B. Folds and Scratches Removal

Folds and scratches can be identified as long white lines on our image formed due to mishandling of the image or aging of the image.

To remove these types of damages we have used grayscale morphological image processing as filling these folds is like closing of image we learnt in the class.
In grayscale morphological image processing there are 2 basic function erosion and dilation.

Erosion: For a given pixel P0, the structuring element is centred on P0. The pixels masked by a coefficient of the structuring element equal to 1 are then referred as Pi.

$$P0 = \min(Pi)$$

Dilation: For a given pixel P0, the structuring element is centred on P0. The pixels masked by a coefficient of the structuring element equal to 1 are then referred as Pi.

$$P0 = \max(Pi)$$

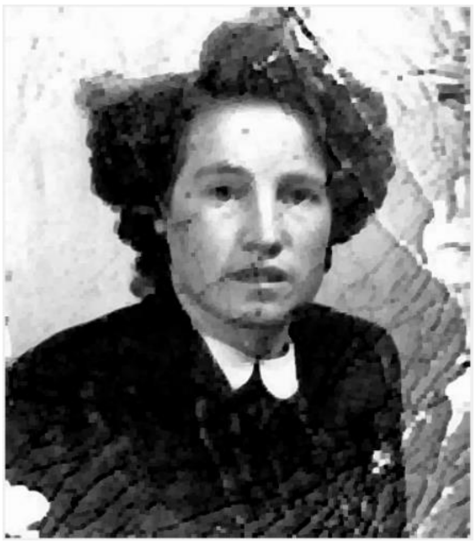


Fig. 9 Output image after fold removal

We took a 3x3 square structure element for the morphological processing for the best output, we found that applying the erosion function twice and dilution. The output would be blurred due these processes, to further improve the quality we have applied the unsharp masking filter already defined before to sharpen our processed image, the result of this process on fig 3 is shown in fig 9

Step – 3: Enhancement Techniques

Histogram Equalization:

Histogram Equalization is used to enhance the contrast in the image autonomously and adaptively. To do this technique we first converted the 2D image into a 1D array and looped over the array to measure the frequency of each intensity level.

Then we started to do cumulative addition of the frequencies of intensity levels and then normalized them to find the cdf of each intensity level, multiplied with (maximum intensity level - 1) and then rounded it off to the nearest integer then we map the initial frequencies to these new intensity levels. Then we converted back the 1D array to 2D image to get the enhanced output image.

The output after Histogram Equalization is shown in fig. 12 and the histograms of original and equilized images are shown in Fig. 10 and Fig. 11.

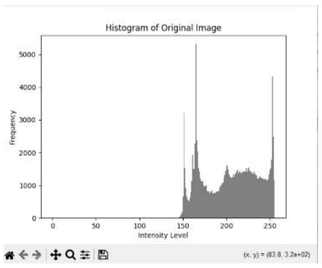


Fig10 Original Image

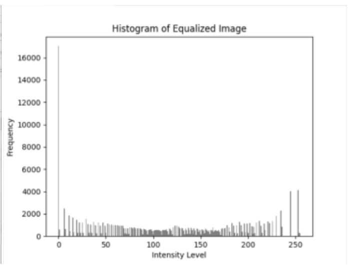


Fig11 Equalized image

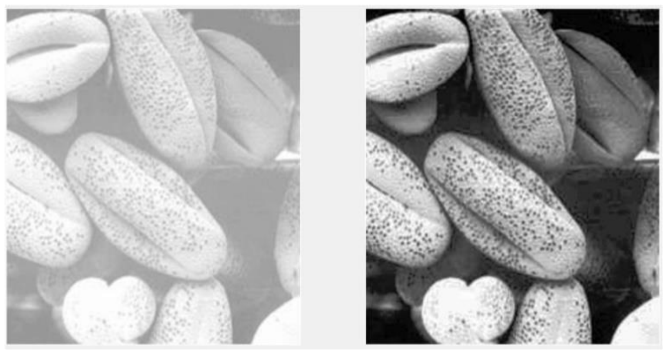
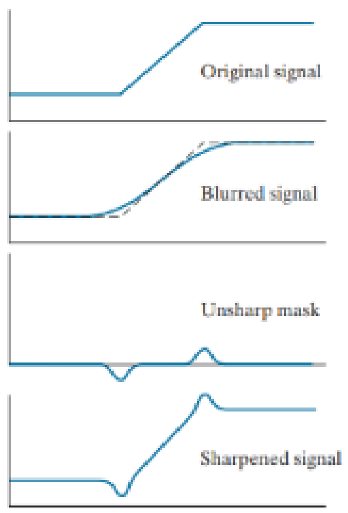


Fig 12 Original Image and Equalized Image

Unsharp Masking:

Most of the processing techniques mentioned



above blur the image or decrease the quality of edges in the image; to resolve this problem we added a sharpening function in our application. We are using an Unsharp Masking to do this.

In Unsharp Masking we first smoothen the image using low pass filters like Gaussian filter and subtract it from the original image to get Unsharp Mask, containing high frequency edges, which can be again added to the image resulting in an edge sharpened image

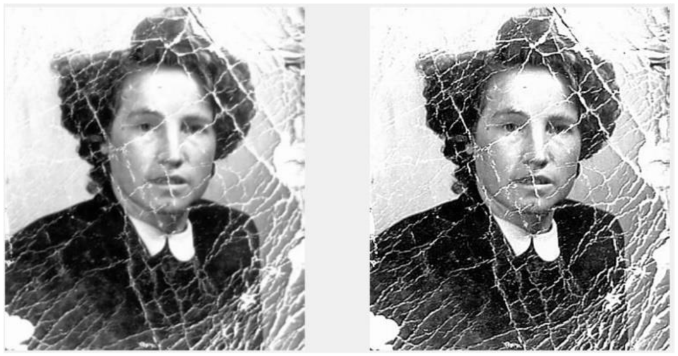


Fig.13 Input and Output Images of Unsharp Masking

Brightness and Gamma Correction:

It is a point operation on a pixel with operator $T = Cr^{\gamma} + K$. It is used to adjust the brightness of an image. It adds a constant value to each pixel intensity to increase brightness and applies a polynomial function to the pdf of pixel intensity to change the intensity level of image.

The output of after gamma correction of γ = 1.2 and C=1, K=30 is shown in Fig. 14.

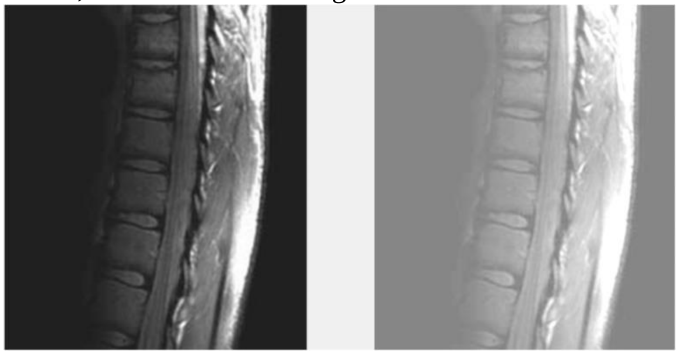


Fig. 14 Input and Output Images of Brightness and Gamma Correction

III. Results

After applying all enhancements, we will choose the best output images. The best output images for the given images are shown in fig. 15 and fig. 16.

Fig. 15 is obtained by applying unsharp masking 2 times, then median filter then stain removal using inbuilt function and then unsharp masking 2 times. Fig. 16 is obtained by applying unsharp masking 2 times at first and then fold removal and then finally median filter.

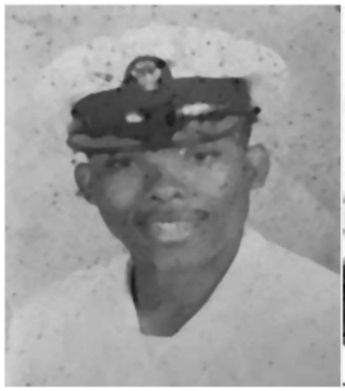


Fig. 15

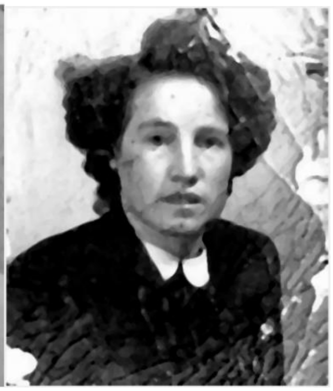


Fig. 16

IV. Conclusion

Finally, we applied different methods that were taught in class to restore the damaged images. The important methods used are Median filter, histogram equalization, unsharp masking, dilation, erosion, gamma correction. We were able to restore the distortions in the areas where correlation is high. In the areas where correlation is low, and information is lost, we cannot use these classical methods to restore and there is need for switching to deep learning techniques.

References

[1] Digital Image Processing FOURTH EDITION Rafael C. Gonzalez • Richard E. Woods.

[2] https://www.ni.com/docs/en-US/bundle/ni-vision-concepts-help/page/grayscale_morphology.html?srsltid=AfmBOooPGQrH1yjuIVS-w9uHZ6cSFjpU5SZXxxP0DerP9DZI_jAdDqTz.

[3] https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html

[4] https://docs.python.org/3/library/tk.html

Team Members

- Gowtham - GUI, Histogram equalization, Gamma correction, Binary Search Fill in stain removal
- Tharun Kumar - Unsharp Masking, Bit Masking, Stain removal
- S.V.S Vivek - Scratch removal, grad fill function in stain removal