Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

DIGITAL IMAGE PROCESSING USING MATLAB

Dr. MANJEET Associate Professor, Electrical Engineering Department, KVMT SIWANI, MDU University, ROHTAK

Abstract
Text is a prominent and direct source of information in video, while the recent surveys of text detection and recognition in imagery focus mainly on text extraction from scene images. Here, this paper presents a comprehensive survey of text detection, tracking, and recognition in video with three major contributions. First, a generic framework is proposed for video text extraction that uniformly describes detection, tracking, recognition, and their relations and interactions. Second, within this framework, a variety of methods, systems, and evaluation protocols of video text extraction are summarized, compared, and analyzed. Existing text tracking techniques, tracking-based detection and recognition techniques are specifically highlighted. Third, related applications, prominent challenges, and future directions for video text extraction (especially from scene videos and web videos) are also thoroughly discussed. To this aim, a supervised DNN is trained to project the input samples into a discriminative feature space, in which the blur type can be easily classified. Then, for each blur type, the proposed GRNN estimates the blur parameters with very high accuracy. Experiments demonstrate the effectiveness of the proposed method in several tasks with better or competitive results compared with the state of the art on two standard image data sets.

Keywords: graphics, Matlab, Image Processing, Image Resolution, processing, signals

Introduction

Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Images are also processed as three- dimensional signals where the third-dimension being time or the z-axis. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. This article is about general techniques that apply to all of them. The acquisition of images (producing the input image in the first place) is referred to as imaging. Closely related to image processing are computer graphics and computer vision. In computer graphics, images are manually *made* from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from *natural* scenes, as in most animated movies.

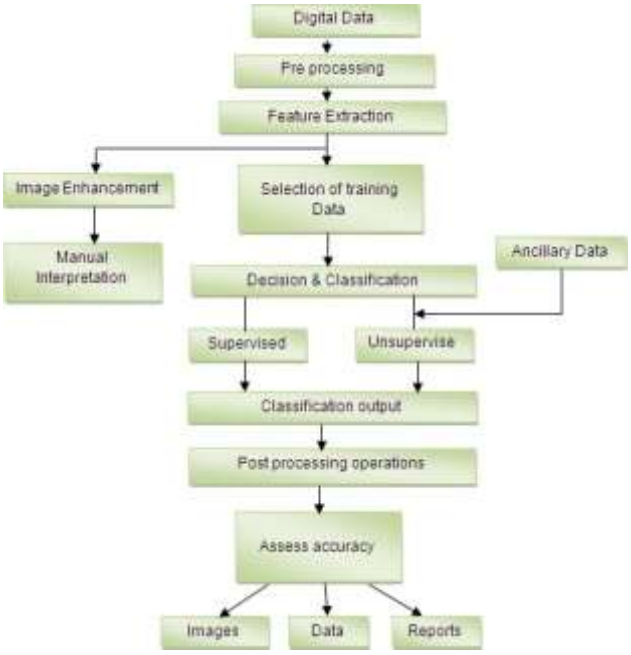
Computer vision, on the other hand, is often considered *high-level* image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans). In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance. It is among rapidly growing technologies today, with its applications in various aspects of a business. Image Processing forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps.

1. Importing the image with optical scanner or by digital photography.
2. Analyzing and manipulating the image which includes data compression and image enhancement and spotting patterns that are not to human eyes like satellite photographs.
3. Output is the last stage in which result can be altered image or report that is based on image analysis.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre-processing, enhancement and display, information extraction.

Fig 1: Flow chart for operations.



Color Trasnform

Multiple methods of representing color data exist. Whilst RGB is most widely used for capture and display, it is not always the best for image processing, since it is a perceptually non-uniformrepresentation. This means that if we change the RGB values by a fixed amount, the observeddifference depends on the original RGB values. One way of observing this is to mix the output ofstandardized colored lights to generate a color, then alter the brightness of the input until an observerjust notices a change in the light's color [5]. The original color and the color of the just noticeabledifference can be plotted. By making measurements systematically over the whole color space, we cangenerate a MacAdam diagram. The points represent the original color, the ellipses the just noticeable difference contours. It is also possible to categories color spaces as being device dependent or device independent. Devicedependent spaces are used in the broadcast and printing industry [6], largely for convenience. The most widely used spaces are YIQ, YCr C and HSV. Conversion between the spaces is by using simple functions. E.g.

>> YIQ = rgb2ntsc(RGB);

Device independent spaces are used because the device dependent spaces include subjective definitions. The CIE defined a standardized color space in 1931. It specifies three color sources,called X, Y and Z. All visible colors can be generated by a linear combination of these. The X, Y andZ values can be normalized, to sum to 1. The color's represented by the normalized x and y values canbe plotted – as in the MacAdam diagram. Conversion of data between color spaces is a

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

two stageprocess. A color transformation structure is first defined, e.g. to convert from RGB to XYZ:

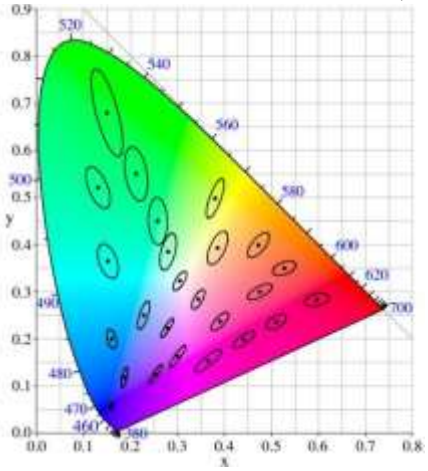


Fig 2: Color Transform

Functions

imread	Read image from graphics file
imwrite	Write image to graphics file
imfinfo	Information about graphics file
nitfinfo	Read metadata from National Imagery Transmission Format (NITF) file
nitfread	Read image from NITF file
dpxinfo	Read metadata from DPX file
dpxread	Read DPX image
analyze75info	Read metadata from header file of Analyze 7.5 data set
analyze75read	Read image data from image file of Analyze 7.5 data set
interfileinfo	Read metadata from Interfile file
interfileread	Read images in Interfile format
hdrread	Read high dynamic range (HDR) image
hdrwrite	Write Radiance high dynamic range (HDR) image file
makehdr	Create high dynamic range image
tonemap	Render high dynamic range image for viewing
hdrread	Read high dynamic range (HDR) image

Image Representation

There are five types of images in MATLAB.

1. **Grayscale.** A grayscale image *M* pixels tall and *N* pixels wide is represented as a matrix of double datatype of size *M*×*N*. Element values (e.g., MyImage(m,n)) denote the pixel grayscale intensities in [0,1] with 0=black and 1=white [7].
2. **Truecolor RGB.** A truecolor red-green-blue (RGB) image is represented as a three-dimensional *M*×*N*×3 double matrix. Each pixel has red, green, blue components along the third dimension with values in [0,1], for example, the color components of pixel (m,n) are MyImage(m,n,1) = red, MyImage(m,n,2) = green, MyImage(m,n,3) = blue.

3. **Indexed.** Indexed (paletted) images are represented with an index matrix of size *M*×*N* and a colormap matrix of size *K*×*3*. The colormap holds all colors used in the image and the index matrix represents the pixels by referring to colors in the colormap. For example, if the 22nd color is magenta MyColormap(22,:)
= [1,0,1], then MyImage(m,n) = 22 is a magenta-colored pixel.
4. **Binary.** A binary image is represented by an *M*×*N* logical matrix where pixel values are 1 (true) or 0 (false).
5. **uint8.** This type uses less memory and some operations compute faster than with double types. For simplicity, this tutorial does not discuss uint8 further.
Grayscale is usually the preferred format for image processing. In cases requiring color, an RGB color image can be decomposed and handled as three separate grayscale images. Indexed images must be converted to grayscale or RGB for most operations.

Owing to recent technological advancement, computers and other devices running several image editing applications can be further exploited for digital image processing operations. This paper evaluates various image processing techniques using matrix laboratory (MATLAB-based analytics). Compared to the conventional techniques, MATLAB gives several advantages for image processing. MATLAB-based technique provides easy debugging with extensive data analysis and visualization, easy implementation and algorithmic-testing without recompilation. Besides, MATLAB's computational codes can be enhanced and exploited to process and create simulations of both still and video images. Moreover, MATLAB codes are much concise compared to C++, thus making it easier for perusing and troubleshooting. MATLAB can handle errors prior to execution by proposing various ways to make the codes faster. The proposed technique enables advanced image processing operations such as image cropping/resizing, image denoising, blur removal, and image sharpening. The study aims at providing readers with the most recent MATLAB-based image processing application-tools. We also provide an empirical-based method using two-dimensional discrete cosine transform (2D-DCT) derived from its coefficients. Using the most recent algorithms running on MATLAB toolbox, we performed simulations to evaluate the performance of our proposed technique. The results largely present MATLAB as a veritable approach for image processing operations.

System design methodology The most common method of image acquisition in image processing task is conducted using a digital photography, usually a digital camera or a preloaded image from a memory source. Image processing technique is used to operate some processes on a picture which includes an image enhancement or the removal of some functional data from the image. Image processing is a type of signal processing, where some operations are performed on the input image data to enhance the quality and extract useful information. Major image processing techniques include image enhancement, image restoration and compression. Digital image processing technique is very effective and notably extensive with wide area of application. Image enhancement involves a process that enhances the quality of the original image including sharpening some of the features of the image to obtain a more useful graphic display suitable for analyzing the image data. The process consists of the following: gray-level adjustment, filtering, edge cropping and mage processing system using MATLAB-based analytics (Gerald K. Ijemaru) 2571 sharpening, interpolation and magnification, and adjustment of color. On the other hand, image restoration is the process that is meant for removal of image noise and estimation of the clean creative image. Blurred or misfocused images and noise are forms of images that can undergo image restoration to obtain the desired output image. All operations are performed on RGB image. In the course of this operation,

the RGB image is converted to gray-scale image, black & white image and binary image by some processes such as edge detection, noise removal, and addition. A three-dimensional (3D) matrix MxNx3 is used to represent RGB image format where the image resolution is a matrix of MxN showing the color picture of the image represented by each dimension of the RGB.

System functions and commands

In theory, the image is a two-dimensional (2D) linear function. Incidentally, sampling, and quantization of the image becomes critical, if the image must be digitally processed on a computer hardware. A 2D image that is uniformly sampled falls under the MxN samples, where it is an integer array. Therefore, a matrix description of the digital image is considered the most percipient and sample approach. The advantage of a MATLAB-based approach is its capability to process matrix operations. Hence, digital images can be processed using MATLAB-based analytics with convenience. The system supports various types of operations on an image and consists of some powerful functions including conversion functions to realize colour conversion and editing functions to perform geometric operations on an image. All these functions can be realized by simply writing some MATLAB code in M file based on MATLAB programming language.

Table 1. **Some MATLAB programming codes and their functions**

MATLAB Command Codes	Function	MATLAB Command Codes	Function
Inread	Reads image from system graphics file to MATLAB environment	imshow	For displaying the image
Imhist	Displays the histogram of the image	imrotate	For image rotation
rgb2gray	Converts RGB image or color map to grayscale	im2bw	Converts image to binary image, based on threshold
Imadjust	For adjusting contrast of an image	imnoise	Adds noise to an image
Imfinfo	Displays information about graphics	imfilter	To reduce image noise
Imadd	For adding images	imcrop	For cropping image
imcomplement	To get the complement of an image	imdistline	To get the measurement of distance between an image
Flipdim	For flipping image	imwarp	For applying geometric transformation
Flipdim	For flipping image	imwarp	For applying geometric transformation

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 1, February : 2023

Figure. Effect of the grayscale image processing

System implementation

The image processing software was created using the graphical user interface (GUI) present in MATLAB. Many programming command codes were used to create different push buttons to perform specific image processing operations and those push buttons were sub-sectioned into panels. The functions ofeach panel and push button are discussed as shown in:

1. **Axes:** the GUI is used to create two axes and they are labelled as axes1 and axes2. The axes1 is meant toinput image that will undergo processing by the user and the axes2 is for the output image.
2. **Input image:** the push button reads image from the system and displays it in the axes1 in the imageprocessing environment.
3. **Reset:** it is used to restore pre-processed image on axes2 to the original image on axes1
4. **Exit:** the exit button is used to close the image processing software window
5. **Rotation and flipping panel**: This panel consists of pushbuttons for image rotation and flipping purposes in anti-clock wise, clockwise and 180 rotation button and horizontal, vertical and flip (horizontal + vertical) button.
6. **Color conversion panel**: this colour conversion panel shows separate views the red, green and blue, and gray shade of the original image.
7. **Image arithmetic panel**: This panel consists of sub pushbuttons which are addition, subtraction, complement of image, linear combination, and absolute difference. Image can only be converted into matrixes in order to perform some mathematical manipulation on them. In this project, the addition pushbutton was programmed to add the original image with a constant, the subtraction pushbutton was meant to subtract from the original image, while constant can multiply on the image by the multiplication pushbutton. Also, the complement button was coded to give the complement of the original image; the linear combination button was programmed to add the image with a factor of the original image and absolute difference button would reveal the difference between the original image and the gray scale of the original image.
8. **Deblurring panel**: this panel contains the three different types of deblurring method pushbuttons which are blind deconvolution, Lucy-Richardson method, and Regularized method pushbuttons.
9. **Noise addition panel**: It contains three types of noise pushbuttons namely speckle noise, salt and pepper noise, and Gaussian noise.
10. **Filtering panel:** this panel contains three types of filtering method pushbuttons namely average filter, guide filer and mean filter. And each filtering pushbutton was coded to remove noise.
11. **Egde detection and contrast adjustment panel**: Edge detection technique is applicable only to binary images, but the original image has to be converted to a binary image before it can undergo edge detection techniques. This panel consists of four different types of pushbuttons namely

Sobel, Roberts, Log, and Prewitt. Also, RGB and Gray shade image adjustment pushbuttons are available in this panel for contrasting of the original image.

12. **Histogram:** this button helps to show histogram of the original image. It shows the frequency of pixels intensity values. The x- & y- axes show the intensities of the gray level & their corresponding frequenciesrespectively.
13. **Crop image**: this is used to select any particular portion of the original image that the user wants to crop.
14. **Blur edge**: this pushbutton is for blurring the edges of the original image.
15. **Sharpness:** it make the original image to have a clearer view (sharpness).
16. **Measure distance:** this button was coded to perform measurement in pixels on the original image but theoperation takes place on 2 axes.
17. **Geometric transformation**: it helps to transform the original image.

Conclusion

The present study discussed a novel image processing technique using MATLAB-based analytics. It also presented an empirical-based model that uses image processing applications with features derived from DCT coefficients. The system was evaluated in MATLAB. We developed algorithms for color detection in images and provided updated MATLAB programming codes for state-of-the-art image processing operations. With different algorithms running on MATLAB toolbox, we provided extensive simulations of several images to evaluate the performance of the proposed approach. The results present the proposed technique as a veritable state-of-the-art approach for digital image processing application. This study aims at providing readers with different image processing applications running on MATLAB platform and to provide researchers with wider knowledge of MATLAB-based image processing techniques that can be exploited for several application specifics. With MATLAB, the various steps employed for image processing can be properly documented and replicated. And MATLAB-based image processing algorithms are more advanced compared to other state-of-the-art image processing applications.

References

1. Rafael C. Gonzalez, Richard E. Woods. Digital image processing. 3rd ed. United States of America: Pearson; 2008. 943 p. Digital Image Processing with MATLAB http://dx.doi.org/10.5772/63028 41
2. John C. Russ. The image processing handbook. 5th ed. United States of America: CRC Press; 2007. 795 p.
3. Henri Maitre. Image processing. 1st ed. Great Britain: Wiley-ISTE; 2008. 359 p.
4. Maria Petrou; Costas Petrou. Image processing the fundamentals. 2nd ed. Singapore: Wiley; 2010. 781 p.
5. John D. Cook. Three algorithms for converting color to grayscale [Internet]. 24.08.2009. http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/. Accessed 02.01.2016
6. Shindong Kang. http://www.slideshare.net/uspace/ujavaorg-deep-learning-with-convolutional-neural-network [Internet]. 26.05.2015 . http://www.slideshare.net/ uspace/ujavaorg-deep-learning-with-convolutional-neural-network. Accessed 29.12.2015
7. Sung Kim. Applications of Convolution in Image Processing with MATLAB [Internet]. 20.08.2013. http://www.math.wash