**LAPORAN PRAKTIKUM PENGOLAHAN CITRA DIGITAL**

**4. IMAGE CROPPING, RESIZING, FLIPPING, AND**

**ROTATION**



**Disusun oleh :**

**Nama : Felix**

**NPM : 2327250059**

**Kelas : IF41**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER DAN REKAYASA**

**UNIVERSITAS MULTI DATA PALEMBANG**

**2025**

**TUTORIAL : IMAGE CROPPING, RESIZING, FLIPPING, AND**

**ROTATION**

**Goal**

The goal of this tutorial is to learn how to crop, resize, and rotate digital images.

**Objectives**

* Learn how to crop an image using the imcrop function.
* Learn how to resize an image using the imresize function.
* Learn how to flip an image upside down and left–right using flipud and fliplr.
* Learn how to rotate an image using the imrotate function.
* Explore interpolation methods for resizing and rotating images.

**Procedure**

In the first part of this tutorial, you will learn how to crop an image. Cropping in MATLAB can be done interactively—using the Crop Image option in the Image Tool (imtool) toolbar—or programmatically—using the imcrop function.

1. Open the cameraman image and use the *Crop Image* option in the Image Tool (imtool) toolbar to crop it such that only the portion of the image containing the tallest building in the background is selected to become the cropped image. Pay attention to (and write down) the coordinates of the top left and bottom right corners as you select the rectangular area to be cropped. You will need this information for the next step.
2. Double-click inside the selected area to complete the cropping operation.
3. Save the resulting image using the File > Save as... option in the imtool menu. Call it cropped\_building.png.



|  |  |
| --- | --- |
|  |  |

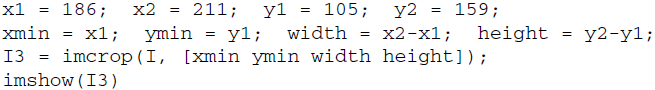
**Question 1** Which numbers did you record for the top left and bottom right coordinates and what do they mean? *Hint*: Pay attention to the convention used by the **Pixel info** status bar at the bottom of the imtool main window. The IPT occasionally uses a so-called *spatial coordinate system*, whereas y represents rows and x represents columns.

|  |
| --- |
| Gambar 1 |
| Gambar 2 |
| kiri atas di koordinat (1,4) dan kanan bawah (254,252), Dimana IPT menggunakan koordinat menggunakan (x,y), yang Dimana cara baca x seperti biasa pada koordinat kartesius, yang berbeda adalah cara membaca y yang Dimana 1 sampai n itu dari atas ke bawah maknaya pada gambar 2 mendapatkan koordinat (254,252). |

1. Open and display the cropped image.



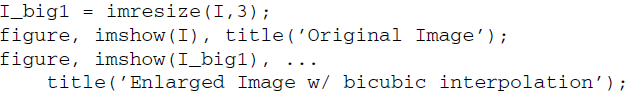
1. We shall now use the coordinates recorded earlier to perform a similar cropping from a script.
2. The imcrop function expects the crop rectangle—a four-element vector [xmin ymin width height]—to be passed as a parameter.
3. Perform the steps below replacing my values for x1, y1, x2, and y2 with the values you recorded earlier.



|  |  |
| --- | --- |
|  |  |

Resizing an image consists of enlarging or shrinking it, using nearest-neighbor, bilinear, or bicubic interpolation. Both resizing procedures can be executed using the imresize function. Let us first explore enlarging an image.

1. Enlarge the cameraman image by a scale factor of 3. By default, the function uses bicubic interpolation.



|  |  |
| --- | --- |
|  | |
|  |  |

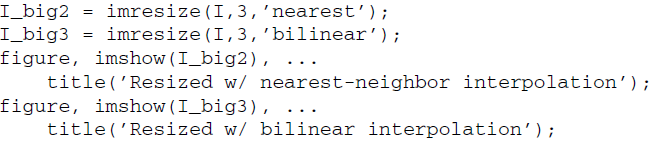
The IPT function imtool can be used to inspect the pixel values of an image. The imtool function provides added functionality to visual inspection of images, such as zooming and pixel inspection.

1. Use the imtool function to inspect the resized image, I\_big1.



|  |  |
| --- | --- |
|  |  |

1. Scale the image again using nearest-neighbor and bilinear interpolations.



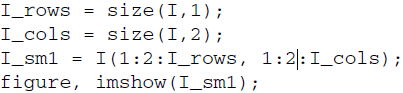
|  |  |
| --- | --- |
|  | |
|  |  |

**Question 2** Visually compare the three resized images. How do they differ?

|  |
| --- |
| Perbedaan dari nearest-neighbor dengan bilinear.  nearest-neighbor :   * Gambar menjadi lebih jelas karena menyalin warna di piksel sekitarnya * Perbedaan warna terlihat berbeda   Bilinear :  - Gambar yang dihasilkan lebih buram atau kabur  - transisi antar warna lebih enak di lihat di bandingkan nearest-neighbor |

One way to shrink an image is by simply deleting rows and columns of the image.

1. Close any open figures.
2. Reduce the size of the cameraman image by a factor of 0.5 in both dimensions.



|  |  |
| --- | --- |
|  |  |

**Question 3** How did we scale the image?

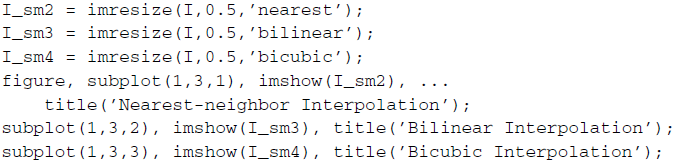
|  |  |
| --- | --- |
|  |  |
| Dengan cara menggunakan imresize(), memperbesar sebanyak 3x dan menggunakan metode bilinear, karena agar hasilnya lebih bagus. | |

**Question 4** What are the limitations of this technique?

|  |  |
| --- | --- |
|  |  |
| Keterbatasan Teknik ini adalah gambar yang diperbesar bisa menjadi kabur karena interpolasi. Bila gambar semakin besar maka akan semakin kabur gambarnya. | |

Although the technique above is computationally efficient, its limitations may require us to use another method. Just as we used the imresize function for enlarging, we can just as well use it for shrinking. When using the imresize function, a scale factor larger than 1 will produce an image larger than the original, and a scale factor smaller than 1 will result in an image smaller than the original.

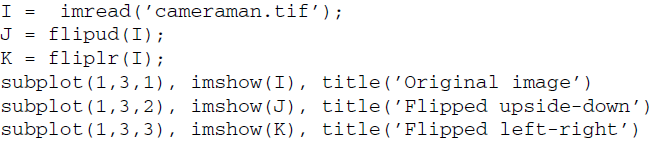
1. Shrink the image using the imresize function.



|  |  |
| --- | --- |
|  |  |

Note that in the case of shrinking using either bilinear or bicubic interpolation, the imresize function automatically applies a low-pass filter to the image (whose default size is 11×11), slightly blurring it before the image is interpolated. This helps to reduce the effects of aliasing during resampling. Flipping an image upside down or left–right can be easily accomplished using the flipud and fliplr functions.

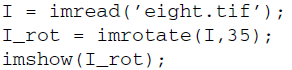
1. Close all open figures and clear all workspace variables.
2. Flip the cameraman image upside down.
3. Flip the cameraman image from left to right.



|  |  |
| --- | --- |
|  |  |

Rotating an image is achieved through the imrotate function.

1. Close all open figures and clear all workspace variables.
2. Rotate the eight image by an angle of 350.



|  |  |
| --- | --- |
|  |  |

**Question 5** Inspect the size (number of rows and columns) of I\_rot and compare it with the size of I. Why are they different?

|  |
| --- |
| Karena i\_rot diputar sebanyak 35 derajat maka pikselnya bergeser dan pasti gambar tersebut akan tidak pas maka matlab secara otomatis memperbesar size-nya agar semua gambar dapat dilihat secara lengkap. |

**Question 6** The previous step rotated the image counterclockwise. How would you rotate the image 35◦ clockwise?

|  |  |
| --- | --- |
|  |  |
| Dengan cara menambahkan minus(-) pada 35 nya maka akan rotet searah jarum jam. | |

We can also use different interpolation methods when rotating the image.

1. Rotate the same image using bilinear interpolation.



|  |  |
| --- | --- |
|  |  |

**Question 7** How did bilinear interpolation affect the output of the rotation? *Hint:* The difference is noticeable between the two images near the edges of the rotated image and around the coins.

|  |
| --- |
| Iinterpolasi bileniar memengaruhi hasil rotasi dengan membuat tepi gambar lebih halus dan membuat transisi warna menjadi lebih rapi |

1. Rotate the same image, but this time crop the output.



|  |  |
| --- | --- |
|  |  |

**Question 8** How did the crop setting change the size of our output?

|  |
| --- |
| Crop pada kode diatas berfungsi agar gambar yang dihasilkan tetap ukuran yang asli dan tetap dirotate. |