

5DATA005C.1

Data Engineering

Name: Sethmika Dias Bellana

UoW ID: w1985751

IIT ID: 20221906

Tutorial Group- BDA Group

Contents

[1. Part 1 3](#_Toc155548836)

[2. Part 2 3](#_Toc155548837)

[2.1. Data Collection 3](#_Toc155548838)

[Self Reflection 5](#_Toc155548839)

# Part 1 - Database for CBIR

## Objective

The objective here is to create a database suitable for content-based image retrieval system. CBIR is a technique that used the visual features of images to find similar images in a large database. Here, through MATLAB a diverse collection of colour images has been collected, pre-processed, annotated with meta data, and extracted relevant image features such as such as colour, pixel intensities, texture, and shape. Then, structured the meta data and extracted features into a JSON file to serve as a foundation for a CBIR system.

## Image Collection

The image collection consists of 50 colour images also known as RGB images. In this case, the image collection is focused on a particular type of object, i.e. animals. The collection consists of 50 colour animal images gathered from ‘Pexels’ website (Pexels, 2024). The 50 images were downloaded while saving their image address for reference, this has been saved as an Xcel sheet in the name of ‘url.xlsx’. The images were downloaded into a folder named ‘original\_images’, then images were then stored in GITHUB repository called ‘[**w1985751\_DataEngineering\_Coursework2**](https://github.com/w1985751/w1985751_DataEngineering_Coursework2)’ (w1985751, 2023).

## Image pre-processing

Image pre-processing was carried out to enhance the quality of the image and standardize them. Image pre-processing was crucial because, while observing the images most of them had noise in them, to a certain extent. Moreover, all the images were in different sizes and some images were too large to.

First the images were read and given the variable ‘OriginalImage’. Then, as shown in *Figure 1*, with the loop each image was resized to 500 x 500 pixels using the imresize function. This was done due to two reasons. Most images had a much larger size and resizing to 500 x 500 pixels was downsizing the to a smaller resolution, this was done to reduce memory usage during the analysis otherwise the processing large image would take longer periods of time. The second reason was to give all the image a standard size to each image, to get a consistent feature extraction.

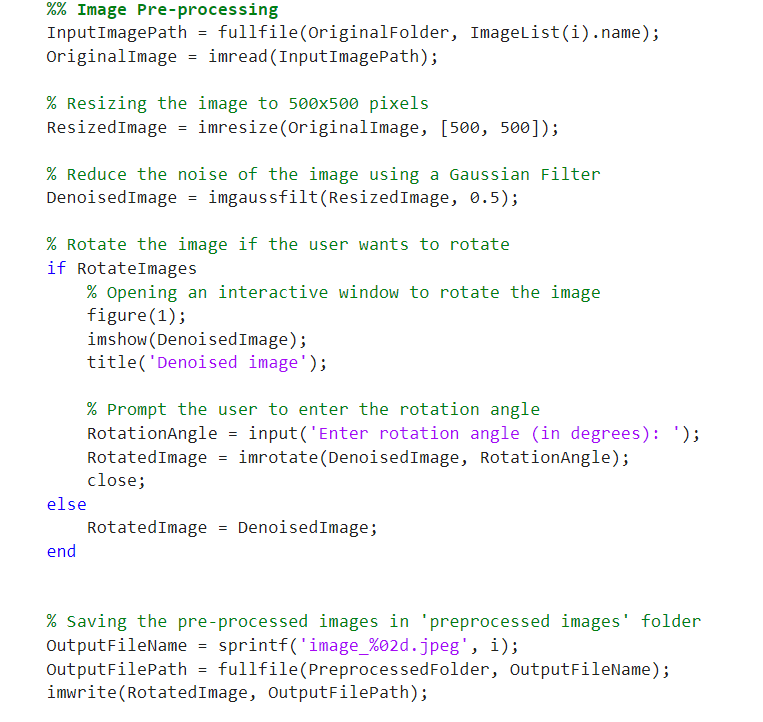


Figure 1 - Image Resizing

Next, when observing the images most of them consisted of noise, and noise is an unwanted variation in pixel values that degrades the quality of the image. Therefore, the images were denoised using the imgaussfilt function, as shown in *Figure 2.*

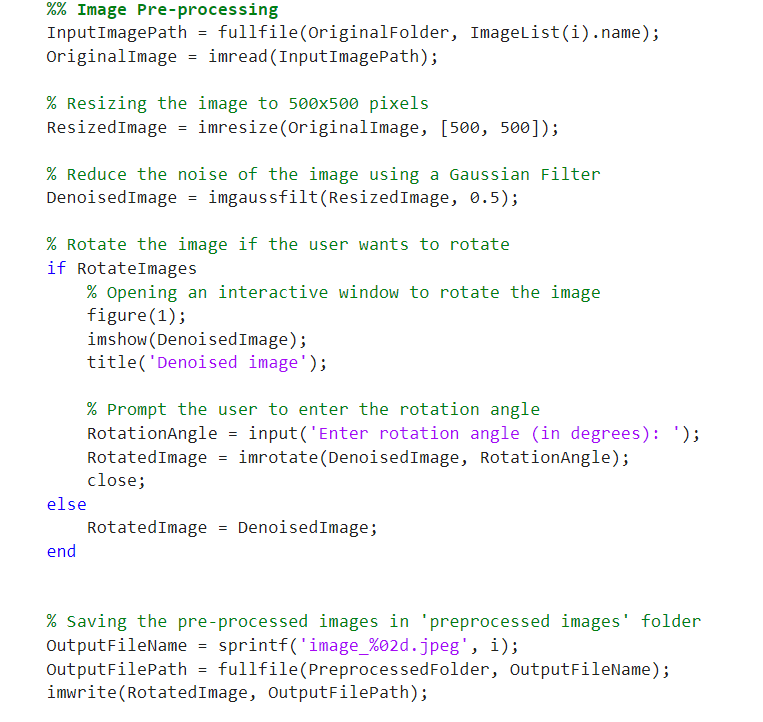


Figure 2- Image Denoising

Human itervension – Rotaation is HUMAN IN THE LOOP

A bird sitting on a branch

Description automatically generatedA blue and white bird on a branch

Description automatically generated

Give a display of both images, side by side code. Write it as a comment

# Part 2

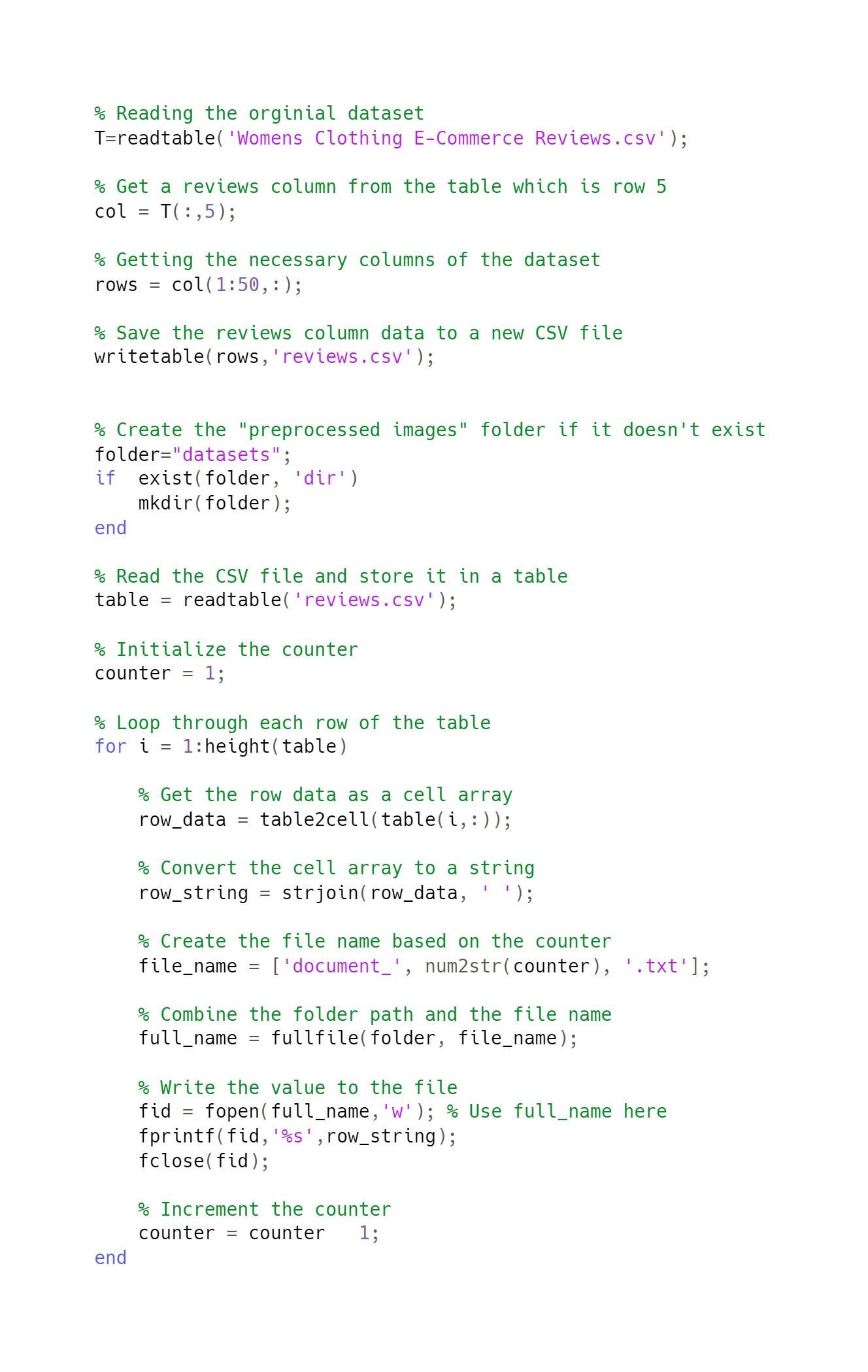
## Data Collection

From the Kaggle website the ‘Women's E-Commerce Clothing Reviews’ was downloaded. This dataset contained reviews given by customers for clothes. The attributes of the dataset were as follows:



Sentiment analysis was carried out for 50 reviews of the customers, therefore the first 50 rows of data from the ‘Review Text’ column was loaded into a new CSV file called ‘reviews.csv’. A close-up of a computer code

Description automatically generated

To conduct the sentiment analysis, the values were all loaded into text documents. For that these steps were carried out:  
  
First a new folder was created, called ‘datasets’ to save all the text documents.

For this process, the data in the newly created ‘reviews.csv’ was read to a table named ‘table’. Then using a loop each row’s data from the table was loaded as a cell and then converted to a string. Then the string value was written to a text file using the fopen, fprintf, fclose functions. The text documents were saves systemically using the format ‘document\_1.txt’, ‘document\_2.txt’ and so on. With this each cell’s data was put into a text document and they were saved in a folder called ‘datasets’.

# Self Reflection

* If I had more time to redo the coursework I would use a better way to filter the noise from the images than using imgaussfilt, as it blurs out an image to reduce the noise and this affects the sharpness and contrast of the image
* the global binarization method also had some drawbacks, such as introducing some artifacts or errors in the background, and losing some details or textures in the foreground. Had to conduct multiple experiments on how to apply the binrisation, made another loop to use the standard deviation of pixel intensities as a threshold to decide which binarization worked best. Although the “global” method gave the best result. If possible would explore more into this topic to binarize the image better.
* Could not save the tags and url documents as csv files, due to the comma’s in url, when saving as a CSV file it divided into many columns.