Feature Extraction from Image

(GrayScale Intensity)

Yamini Kasediya (0801IT171092)

gs0801it171092@sgsitsindore.in

December 20, 2020

Contents

Chapter 1

Introduction

1.1 Feature Extraction From Image

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process them. So Feature extraction helps to get the best feature from those big data sets by select and combine variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with the accuracy and originality. When you have a large data set and need to reduce the number of resources without losing any important or relevant information. Feature extraction helps to reduce the amount of redundant data from the data set.

1.2 Grayscale Intensity

Grayscale is a range of monochromatic shades from black to white. Therefore, a grayscale image contains only shades of gray and no color. While digital images can be saved as grayscale informations. This is because each pixel has a luminance value, regardless of its color. Luminance can also be described as brightness or intensity , which can be measured on a scale from black (zero intensity) to white(full intensity). Most image file formats support a minimum of 8-bit grayscale, which provides 2^8 or 256 levels of

luminance per pixel. Some formats support 16-bit grayscale, which provide 2^16 or 65,536 levels of luminance.

Chapter 2

Mathematical Formulation

2.1 Formulation

How do you convert a color image to grayscale? If each color pixel is described by a triple (R, G, B) of intensities for red, green, and blue, how do you map that to a single number giving a grayscale value?
The GIMP image software has three algorithms.

1. The **lightness** method averages the most prominent and least prominent colors: (max(R, G, B) + min(R, G, B)) / 2.

2. The **average** method simply averages the values: (R + G + B) / 3.

3. The **luminosity** method is a more sophisticated version of the average method. It also averages the values, but it forms a weighted average to account for human perception. We're more sensitive to green than other colors, so green is weighted most heavily. The formula for luminosity is 0.21 R + 0.72 G + 0.07 B.

Chapter 3

Algorithm

[1],

3.1 Algorithm: Grayscale to Color Coversion and feature extraction

Input : image(.jpg)

Output: feature vector

1. Read image in the form of M X N X 3 matrix.
2. For each pixel image extract RGB value and calculate corresponding mean pixel value and store it into feature matrix.

for each pixel in image:

Y = (0.299 x R) + (0.587 x G) + (0.114 x B)
U = (B - Y) x 0.565;
V = (R - Y) x 0.713;
UV = U + V;
R1=R*0.299;
R2=R*0.587;
R3=R*0.114;
G1=G*0.299;
G2=G*0.587;
G3=G*0.114;
B1=B*0.299;
B2=B*0.587;
B3=B*0.114;
R4=(R1+R2+R3)/3;
G4=(G1+G2+G3)/3;
B4=(B1+B2+B3)/3;
I1=(R4+G4+B4+UV)/4;
feature matrix = I

3. Convert 2D feature matrix into feature vector.

Chapter 4

Documentation of API

4.1 Package organization

Package Name: GrayScaleIntensity

class GrayScaleIntensity

Parameters:
Image = it is a 3D matrix having dimension M X N X 3

Methods:
g_feature(image)

Method which returns the extracted feature vector from grayscale image. if image is not a grayscale image then it first convert the image into grayscale image and do the feature extraction and return feature vector.

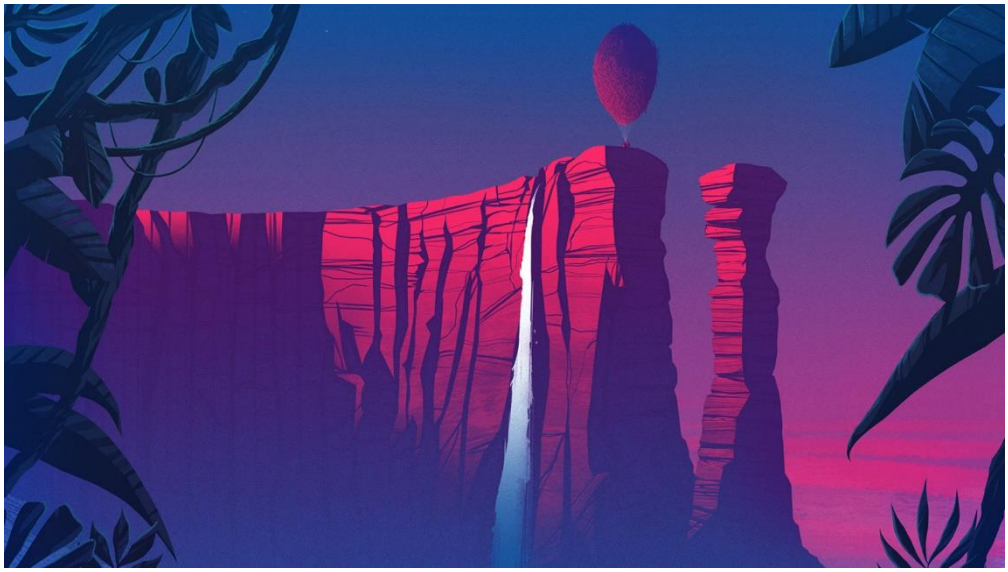
Chapter 5

Example

```
import pandas as pd
import numpy as np
from skimage.io import imread, imshow
from GrayScaleIntensity import GrayScaleIntensity
#reading color image using imread()
image = imread("purple.jpg")
#make Object by calling GrayScaleIntensity
gsi=GrayScaleIntensity()
#passing image to g_feature to get grayscale feature vector
f = gsi.g_feature(image)
print("feature vector of Grayscale image")
#Outputted feature vector of grayscale image
print(f)
```

```
feature vector of Grayscale image
[23.79280033 21.23384383 22.63675483 ... 48.77127933 48.16890767
 47.42259067]
```

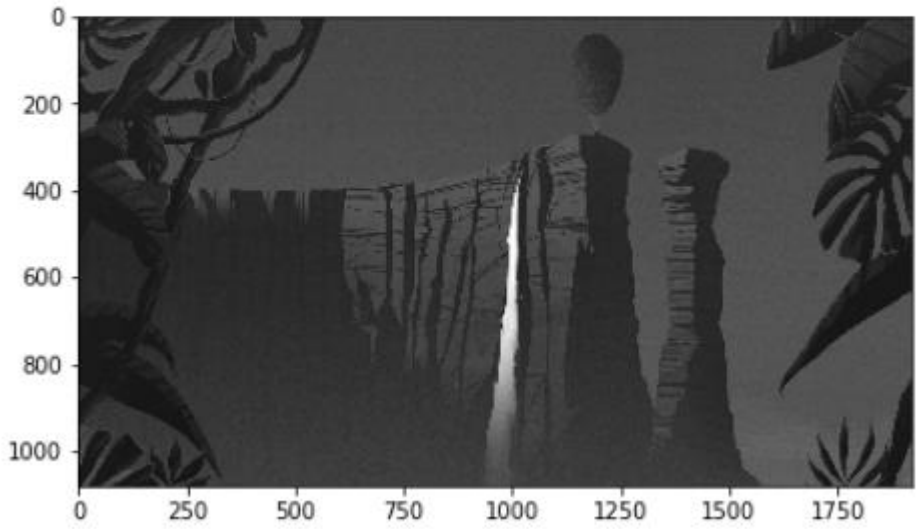
Input image:-



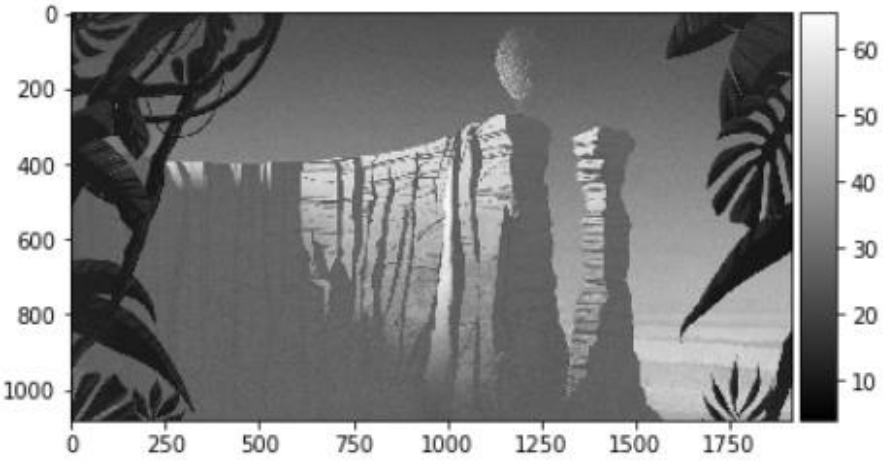
Converted grayscale image by built in method as_gray(True)

```
image = imread("purple.jpg",as_gray=True)
imshow(image)
```

```
<matplotlib.image.AxesImage at 0x279a6b26c40>
```



Grayscale image conversion by proposed algorithm



Chapter 6

Learning Outcome

- Gained deeper knowledge of how the images are stored in the devices.
- The conversion of color image to grayscale image using the proposed algorithm uses approximation of RGB values using luminance RGB components approximated RGB reduced by three, added with chrominance value and average of these four value results perceptually and structurally good quality of grayscale images.
- Feature extraction helps to reduce the amount of redundant data from the data set. In the end, the reduction of the data helps to build the model with less machine's efforts and also increase the speed of learning and generalization steps in the machine learning process.

References

1. https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/
2. https://www.analyticsvidhya.com/blog/2015/01/basics-image-processing-feature-extraction-python/
3. https://www.researchgate.net/publication/224130500_Color_Image_to_Grayscale_Image_Conversion
4. https://en.wikipedia.org/wiki/Grayscale