FEATURE EXTRACTION FROM IMAGE

Pixel Average-Median-Mode

Kshitij Gour (0801CS161035)
Sanchit Bhat (0801CS161075)

Contents

1. Introduction

1.1. Feature Extraction from Image

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. Feature Extraction from image Therefore, processing the data becomes easier. The technique of extracting the features is useful there is a large data set and it is needed 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. Pixel Average-Median-Mode

A digital photo is not one non-dividable thing. If we zoom in far enough, we’ll see that our image is like a mosaic formed by small square tiles, which are called pixels. We see images as they are – in their visual form. Machines, on the other hand, store images in the form of numbers. Each number corresponds to a pixel value which denotes the intensity or brightness of the pixel. Smaller numbers (closer to zero) represent black, and larger numbers (closer to 255) denote white. A colour image is typically composed of multiple colours and almost all colours can be generated from three primary colours – red, green and blue.Hence, in the case of a colour image, there are three Matrices (or channels) – Red, Green, and Blue. Each matrix has values between 0-255 representing the intensity of the colour for that pixel. To calculate Mean, Median, Mode of these pixel values for a colourful image, we need to perform these functions on each channel separately to get these values for each channel.

2. Mathematical Formulation

2.1. Formulation

1. Mean means the average of the given values. To compute mean, sum all the values and divide the sum by the number of values.

   $$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

   In our case, values will be pixel values of each channel of the image.

2. Median means the value where the upper half of the data lies above it and lower half lies below it. In other words, it is the middle value of a data set.To calculate the median, arrange the data points in the increasing order,

   - If there is an odd number of data points:

     Median will be [*(n + 1) / 2*]th value.

   - If there is an even number of data points:

     Median will be average of [*n / 2*]th value and [*(n / 2) + 1*]th value.

3. Mode is the value that occurs most frequently in our data set. To calculate mode, find the frequency of each data point and find the data point with highest frequency.

3. Algorithm

3.1. Mode Algorithm

Algorithm:

Input: 3-dimensional image

Output: numpy array containing mode of each channel and entire image

1. Image is received as a three-dimensional matrix of shape M x N x 3.
2. Create a tuple *color* with three strings i.e., ‘b’, ‘g’ and ‘r’ and a variable *maximum* = 0.
3. Start a *for* loop which enumerates *color* to run a total of 3 times for each channel in an image:
    - 3.1. Calculate histogram for current channel.
    - 3.2. Find the pixel value where maximum frequency of histogram occurs and store it.
    - 3.3. Check if it is the max frequency greater than *maximum*, if true then update overall mode value and store max histogram value in *maximum.*
4. Store all calculated most frequent pixel values in a numpy array *mode*.
5. Return the array *mode.*

3.2. Mean Algorithm

Algorithm:

Input: 3-dimensional image

Output: numpy array containing mean of each channel and entire image

1. Image is received as a three-dimensional matrix of shape M x N x 3.
2. Separate blue, green and red channels using cv2.split() and store in variables *b*, *g*, and *r*.
3. Calculate mean of each channel by passing *b, g,* and *r* one by one to numpy.mean() and store the resulting values in different variables.
4. Calculate mean of entire image by passing *image* to numpy.mean().
5. Store all calculated average pixel values in a numpy array *mean*.
6. Return the array *mean.*

4. Documentation of API

4.1. Package Organization

from pixel_features import pixel_features

__init__() not defined in this class.

4.2. Methods

def mean_pixel_value (image)

Returns numpy array containing average pixel values of each channel and entire image

Parameters:

image: input three-dimensional image

def median_pixel_value (image)

Returns numpy array containing median pixel values of each channel and entire image

Parameters:

image: input three-dimensional image

def mode_pixel_value (image)

Returns numpy array containing most frequent pixel values of each channel and entire image

Parameters:

image: input three-dimensional image

5. Examples

5.1. Example 1

```
image = cv2.imread('puppy.jpg')

median = features.median_pixel_value(image)
```

Output:array([23., 32., 26., 27.])

5.2. Example 2

```
image = cv2.imread ('puppy.jpg')

mean = features.mean_pixel_value (image)
```

Output:array([43.78883905, 51.82413609, 57.27795148, 50.96364221])

5.3. Example 3

```
image = cv2.imread ('puppy.jpg')

mode = features.mode_pixel_value (image)
```

Output:array([21, 25, 21, 21], dtype=int64)

6. Learning Outcomes

- Capacity to integrate theoretical and practical knowledge to evaluate features of an image using different mathematical functions.
- Extracted and analysed some important features related to pixels of an image.
- Analysed the way in which machine stores and processes images in the form of arrays.
- Analysed and evaluated higher mathematical concepts with the ability to clearly implement and present the conclusions and the knowledge behind it.
- Capacity to design and perform research on different aspects of feature extraction from images.It can help with real life applications in form of computer vision projects.

Appendix A

References

1. https://www.analyticsvidhya.com/blog/2019/08/3-techniques-extract-features-from-image-data-machine-learning-python/?#

2. https://www.mygreatlearning.com/blog/feature-extraction-in-image-processing/