Introduction

Table of Contents

[Levels of Processing 3](#_Toc120739461)

[Applications 4](#_Toc120739462)

[Digital Image Representation 4](#_Toc120739463)

A **digital image** is a representation of a 2D function, , as a finite set of digital values called **picture elements** or pixels.

A black and white image consists of a set of pixels, with each pixel having either a 0 or 1 value. Thus, each possible value of is either a 0 or a 1. If we move up to a greyscale image, each value is represented using 8 bits, thus 256 possible shades of grey.

If we have bits available, we can divide the values into discrete levels. 0 represents black and represents white. The other levels in between are divided into different shades.

For **greyscale images**, we have **256 values per pixel**, from 0 to 255. Each value represents a different brightness value. If we consider **RGB images**, we have three such channels for every pixel. If we separate out a channel, we will basically end up back with a greyscale image since there would be no way to know which colour it represents. This means that if we want an RGB image with just the red channel, we need to use all three channels and set the green and blue channels to all 0s.

In the real world, images are visible when light falls on an object and is reflected. In this case, we receive a continuous signal of information. This is why nothing appears blurry in real life. In the digital world however, it is not possible to show an infinite amount information. Each pixel holds a limited amount of data.

If we increase the number of pixels without increasing the size of the image, each pixel needs to represent information from a smaller area. Essentially, instead of using 8 bits to capture the data in 1 square, we use 16 bits, divided into 2 pixels. This makes the picture clearer.

The opposite happens if we view the same image on a larger screen. Suppose we have an image with just 2 pixels, with the values 10 and 20. If we try to view this image on a 4-pixel screen it will mean that there is not enough information in the 2-pixel image to fill the screen. Since the image needs to be stretched, values are replicated twice, which becomes 10, 10, 20 and 20. This leads to the staircase effect.

We can deal with the staircase effect by using **interpolation**. Basically, this means that the values in between the stretched pixels are estimated. For the above example, it could be something like 10, 13, 17, 20. This reduces the amount of pixelation.

In digital image processing, we are mainly trying to solve two tasks:

* Improve pictorial information for human interpretation
* Process image data for storage, transmission and representation for autonomous machine perception

## Levels of Processing

There is no general agreement about exactly where ‘digital image processing’ starts or stops. It could or could not include general digital image processing, computer vision or artificial intelligence. However, it can be divided into three **processing levels**:

* **Low-Level Processing** – At this level, the input is an image, and the output is also an image. It includes things like sharpening the image, removing background noise, etc.
* **Mid-Level Processing** – At this level, the input is an image, but the output is attributes of the image. It includes things like segmentation and object recognition.
* **High-Level Processing** – Images are not directly involved at this level. Instead, the input is the attributes of images obtained from mid-level processing and the output is understanding of the contents. It includes things like scene understanding, autonomous navigation, etc.

## Applications

* Satellite Imagery
* Animation
* Medical Domain
* Law Enforcement
* Human-Computer Interaction

## Digital Image Representation

Digital images are typically represented as a 2D matrix of pixel values, denoted as . However, when processing the images, it may be computationally beneficial to **flatten** the representation either using row-major formatting, where the 2D matrix is flattened row by row, or column-major formatting, where the 2D matrix is flattened column by column.