IMAGE PROCESSING ESSENTIALS

A Comprehensive Guide to Image Processing: Fundamentals

A step-by-step Image Processing flow with MATLAB or Python Implementations

Yağmur Çiğdem Aktaş · Follow
Published in Towards Data Science
15 min read · Aug 18, 2021

All of the operations performed on a digital image are subject to Image Processing. Although there are various purposes of changing an image by performing various operations on it, Computer Vision is one of the areas where Image Processing is used the most. In this series, I will mention different terms and operations and *Digital Image Processing — Third Edition* by Rafael Gonzalez and Richard E. Woods is the sourcebook that I used both in my education process and while preparing these articles.

Here is a list of the contents of each post that makes part of this series—click to start where ever you want to learn!

Image Processing Part1 (which starts just below 🏃)

- What is an image? & Image Acquisition
- Sampling & Quantization
- Image Enhancement with Gray Level Transformations
- Histogram Processing

- MATLAB implementation of these processes from scratch

Image Processing Part 2

2.1

- Non-Linear Spatial Filtering
- Min Max Median Filtering
- Padding
- Python Implementation of these processes from scratch and PIL library

2.2

- Linear Spatial Filtering
- Convolution and Correlation
- Smoothing Filters
- Sharpening Filters
- Edge Detection Filters
- Noise Removal Filters
- Python Implementation of these processes from scratch and OpenCV library

Image Processing Part 3

- Morphological Operators
- Erosion & Dilation
- Combining Morphological Operators for Noise Removal and Edge Detection
- OpenCV examples for applying these operations on images, Scipy examples for applying these operations on 2D matrices.

Image Processing Tool

A tool implemented using OpenCV 3.2.0 on QT Creator with C++ to apply almost all the Image Processing operations discussed in these posts.

You can access all the codes used here visiting my **github** link 💻

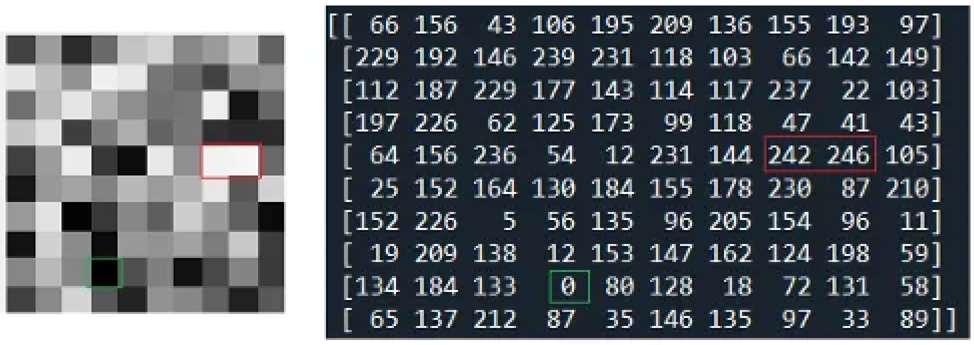
Let’s get started with the first part!

Image Processing Part 1

A scene, a view we see with our eyes, is actually a continuous signal obtained with electromagnetic energy spectra. The value of this signal perceived by the receptors in our eye is basically determined by two main factors: the amount of light that falls into the environment and the amount of light reflected back from the object into our eyes.

As we know, there are various light sources with different wavelengths on the earth, and we can only distinguish light with a certain wavelength with our eyes. Here, we can call all the scenes that we can perceive thanks to the light in this range, as “**pictures**”. We said that the information reaching our eyes by this light source is “**continuous**”. When we want to transfer this information to digital media and store it in this environment, it means that we are talking about the concept of digital picture, namely “**discrete**” signals. [1]

That is, the digital picture is a 2-dimensional matrix and each element of the matrix actually carries the value of the relevant part of that discrete signal.



```
[[ 66 156  43 106 195 209 136 155 193  97]
 [229 192 146 239 231 118 103  66 142 149]
 [112 187 229 177 143 114 117 237  22 103]
 [197 226  62 125 173  99 118  47  41  43]
 [ 64 156 236  54  12 231 144 242 246 105]
 [ 25 152 164 130 184 155 178 230  87 210]
 [152 226   5  56 135  96 205 154  96  11]
 [ 19 209 138  12 153 147 162 124 198  59]
 [134 184 133   0  80 128  18  72 131  58]
 [ 65 137 212  87  35 146 135  97  33  89]]
```

Figure 2 “Image by Author”

Accordingly, have you noticed that as the amount of light in the signal increases (so the color value in the picture moves from black to white), the values in the matrix also increase?

So how is the transition shown in Figure 1 actually done? How is the signal obtained in the form of a continuous signal converted into a discrete signal, that is, into a digital picture? For the answers to these questions, we will mention 2 basic concepts:

Sampling & Quantization

Sampling is the process that will determine the size of the picture to be obtained. It deals with the number of pixels of the digital picture, that is, the matrix size. Increasing the size of a digital image is named "*up sampling*" and decreasing is named "*down sampling*". You can often see the word *spatial* when researching the sampling process. Sampling is a spatial process. Because it deals with the size of the picture, completely independent of the content of the picture or the values it will have.

Quantization is the process that will determine the value of each pixel in the image. First of all, a basic gray-scale level, that is, the range of values that a pixel can have, is determined. Then, according to the value coming from the continuous signal, the value of the pixel in the digital picture is calculated. If the *gray level* of a picture is *n bits*, its *grayscale range* will have the value of 2^n. For example, the range of values that pixels can have in a gray level of 3 bits is [0–7], while the range of values that pixels can take in a gray level of 8 bits is [0–255]. The values that pixels have are also called the "*intensity*" of that pixel.

!!! We mentioned that when a scene is transferred to the digital picture environment, a continuous signal is transformed into a discrete signal. Another important point here is that the discrete signal does not carry all of the information in the continuous signal. For example, as in Figure 3, instead of receiving all the information in the continuous signal, we only received 16 pieces of information, so we obtained a 16-pixel picture as the output picture. The larger the size of the output picture, that is, the larger the sampling is done, the closer the output picture to the scene, with less information loss. Likewise, the larger the grayscale level, the closer the value of a pixel in the output image will be to the value in the actual scene.

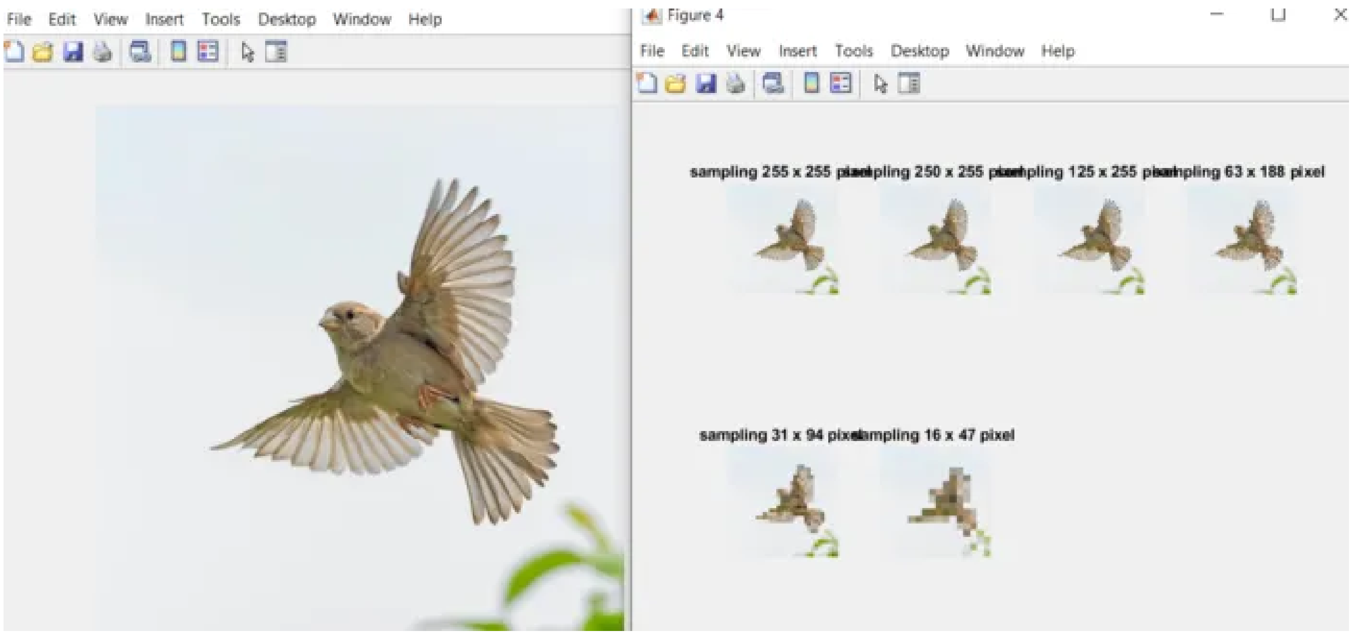


Figure 4 "Image by Author"

!!! The sampling or quantization properties of a digital image can be changed repeatedly, resulting in different outputs of the same image.

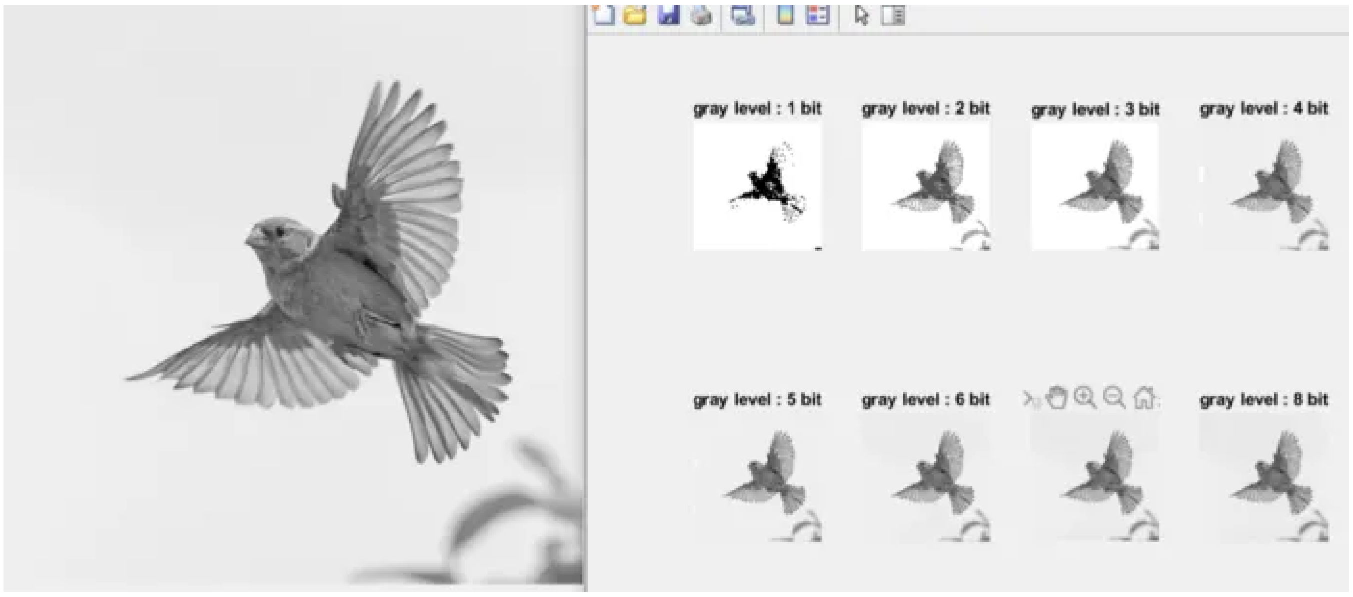


Figure 5 "Image by Author"

While the left image above shows the original version of a picture, we see different sampling examples of this picture in Figure 4. As the number of pixels used for sampling decreases so, the size of the output picture decreases, the quality of the picture we get decreases. In fact, the process is nothing more than getting a new picture by having less and less information from the same scene!

While the left picture above shows a black-white uint8 type picture (so the values of the picture matrix can be any value between 0–255), Figure 5 shows the outputs of the same picture at different quantization levels. When we examine Figure5, while the gray level is 1 bit, that is, the values in the picture matrix can only take one of the values of 0 or 1, the output shows that the contrast of the picture is at the highest level and the quality of the picture is at the lowest level. When the gray level reaches 8 bit, we see that the output image is of the same quality as the input image, since the quantization level is at the same level as the original image.

!!! In Figure 5, we see that beginning from the 5-bit gray level, the images are almost the same quality. But if you want to examine the picture matrix, you can see that matrix values are in the range of 0–32 at 5-bit gray level, matrix values in the range of 0–64 at 7-bit gray level, and in the range of 0–128 at 7-bit gray level.

If you want to do more exercises on this topic and pictures, you can access the relevant Matlab code by clicking: sampling & quantization.

Image Enhancement

The processes applied to obtain a different version of the image on an image that does not have any deformation in its essence are included in the operations in the Image Enhancement category. These can include changing the light intensity of a picture, changing its contrast, manipulating color settings, etc. Yes, these are exactly the operations behind that you can see your picture in different colors and different settings at the touch of a button in applications such as Instagram, etc.!

We will examine Image Enhancement in 2 different categories. If an image is processed pixel by pixel, that is, if the process is applied to each pixel individually, it’s examined in Gray Level Transformations and if the processes are applied to pixel groups larger than 1 pixel, it is examined in the Filtering category.

!!! The names Intensity Transformation, Mapping Transformations or Point Processing also mean the same as Gray Level Transformations. You may also come across these names on the Internet or in different books.

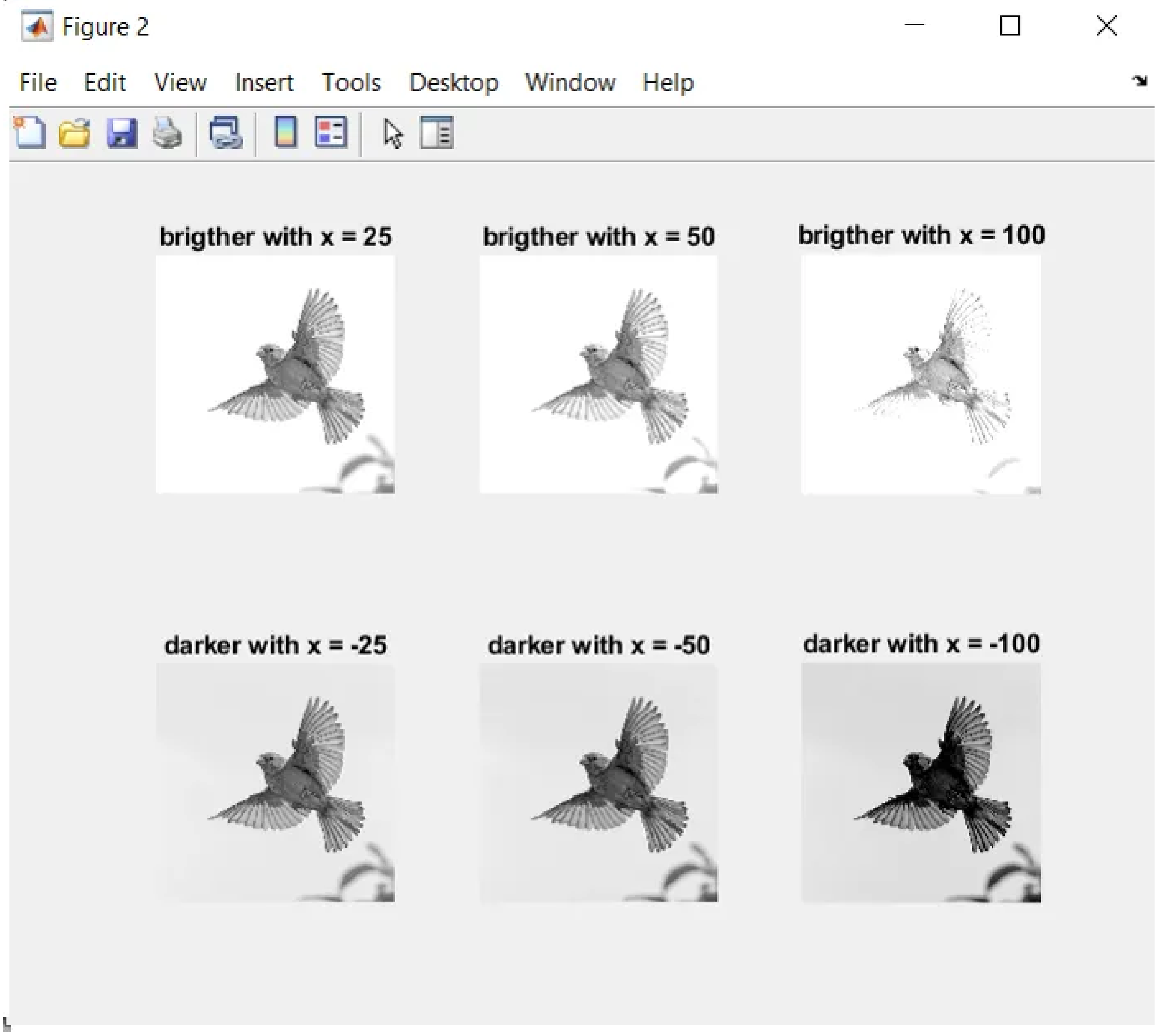
Gray Level Transformations

In this section, we will see the output picture by applying different operations to our basic picture seen below.

“Image by Author”

1. **Linear Transformations**

When we consider a picture matrix that can take values between 0–255, we said that the pixels closer to 0, are closer to black and the ones closer to 255, are closer to white. Accordingly, it would be sensible to increase the pixel values when we want to lighten a picture and to decrease the pixel values when we want to darken it, right? Increasing or decreasing the value of each pixel by a constant number x is a linear process.

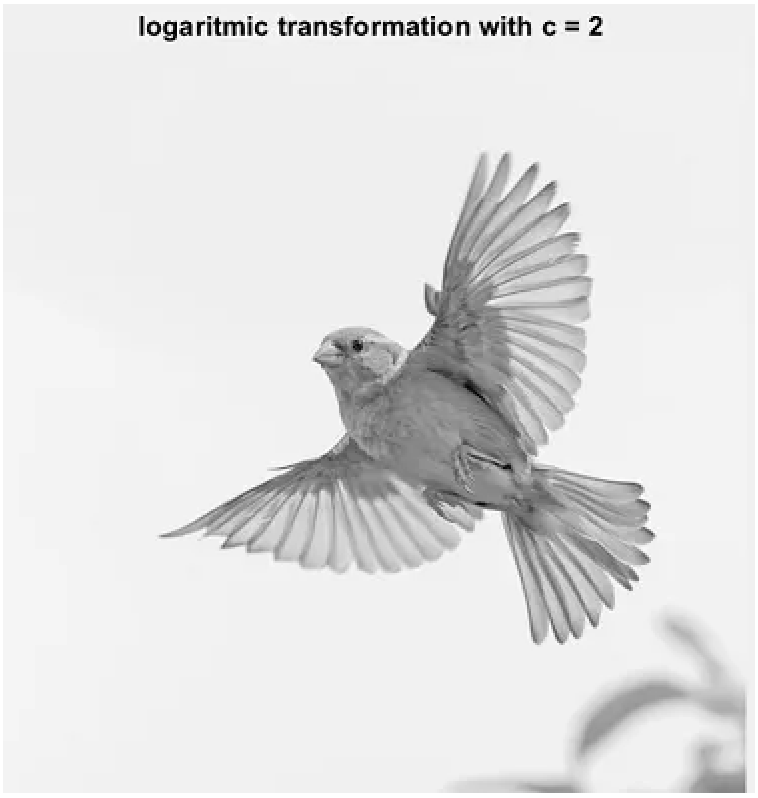


“Image by Author”

Another linear process is to take the negative of the picture. This process is to subtract each pixel from its complement (the largest value a pixel in the picture can get according to the grayscale range) For example, taking the negative of an 8-bit grayscale level image equates to subtracting each pixel from 255.

2. **Logaritmic Transformations**

Another process used in changing the light intensity is the logarithmic transformation. The pixels of the output picture are obtained by applying each pixel to the following process: s = clog(1+r). Here c is the coefficient that will determine the effect of the process, and r is the relevant pixel.



"Image by Author"

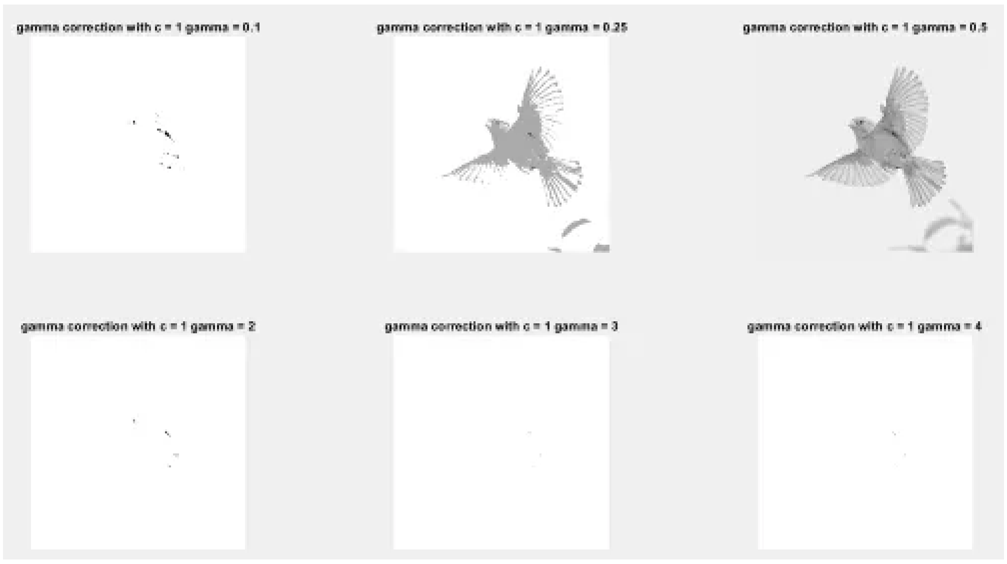
3. **Power Law Transformations — Gamma Correction**

Another process used to change the light of the picture is power-law transformation, also known as gamma correction. The pixels of the output picture are obtained by operating each pixel with the following process :

$$s = cr^{\gamma}$$

Here c and γ are two different coefficients, r is the respective pixel.

Gamma correction changes not only the amount of light but also the contrast settings of an image. Because the power of gamma coefficient is taken for each pixel and it decreases or increases the distance between pixels values depending on whether this number is in the range 0 -1 or greater than 1.


gamma correction with c = 1 gamma = 0.1
gamma correction with c = 1 gamma = 0.25
gamma correction with c = 1 gamma = 0.5
gamma correction with c = 1 gamma = 2
gamma correction with c = 1 gamma = 3
gamma correction with c = 1 gamma = 4

“Image by Author”

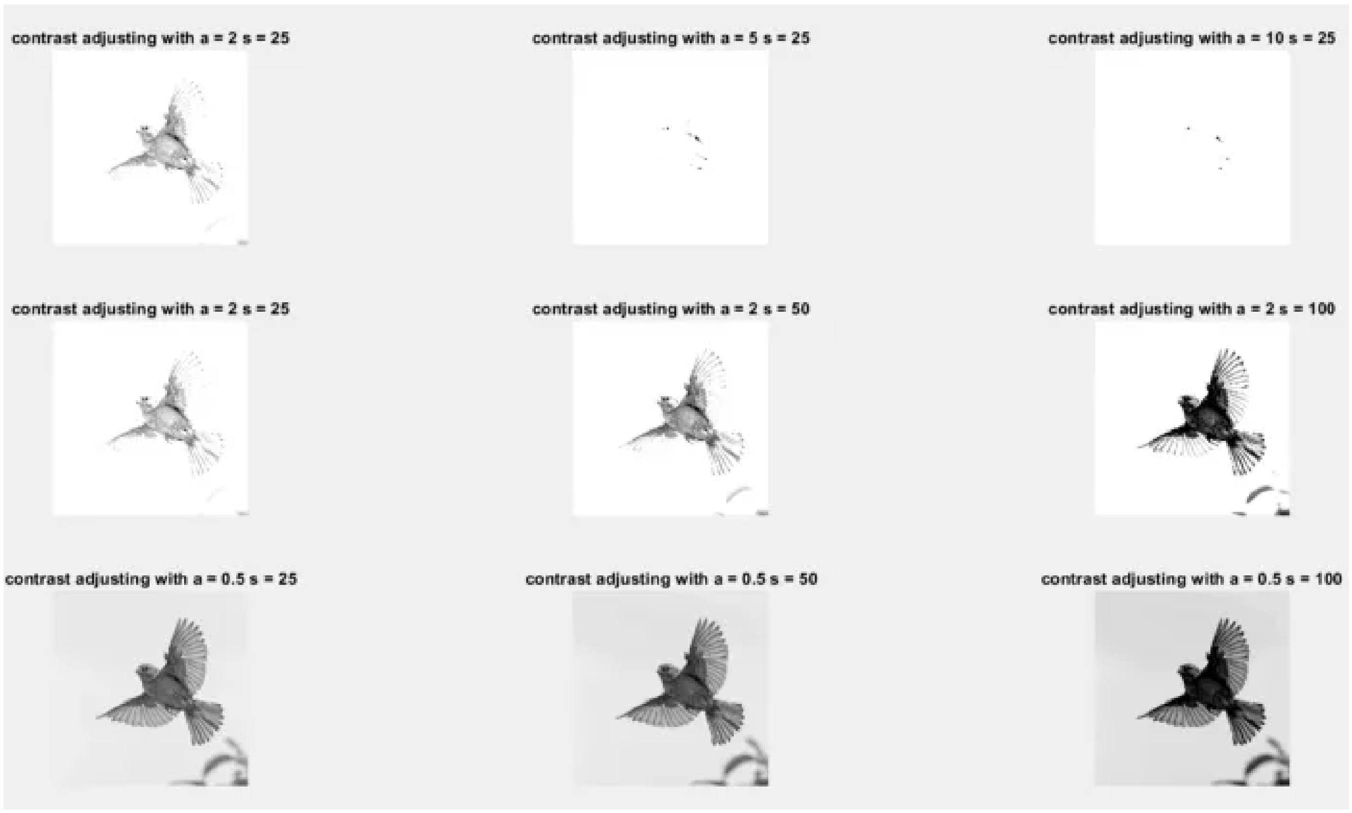
When we apply gamma correction to our same base picture, we see that we get useless pictures in gamma > 1 cases because the original contrast of the picture is already high, and we get pictures with lower contrast with a gamma correction in the case of
0 <gamma <1.

4. **Contrast Transformations**

The contrast of a picture is the amount of intensity — value difference between its pixels. If the pixels have values close to each other, the contrast of the picture is low, if they have values farther apart from each other, the contrast is high. The basic process applied to increase or decrease the contrast has the following formula.

$$a(r - s) + s$$

In this formula, r: the relevant pixel s: a coefficient a:> 1 is a coefficient that converts the applied process to contrast augmentation, and if 0 <= a <1, the applied process to contrast reduction.

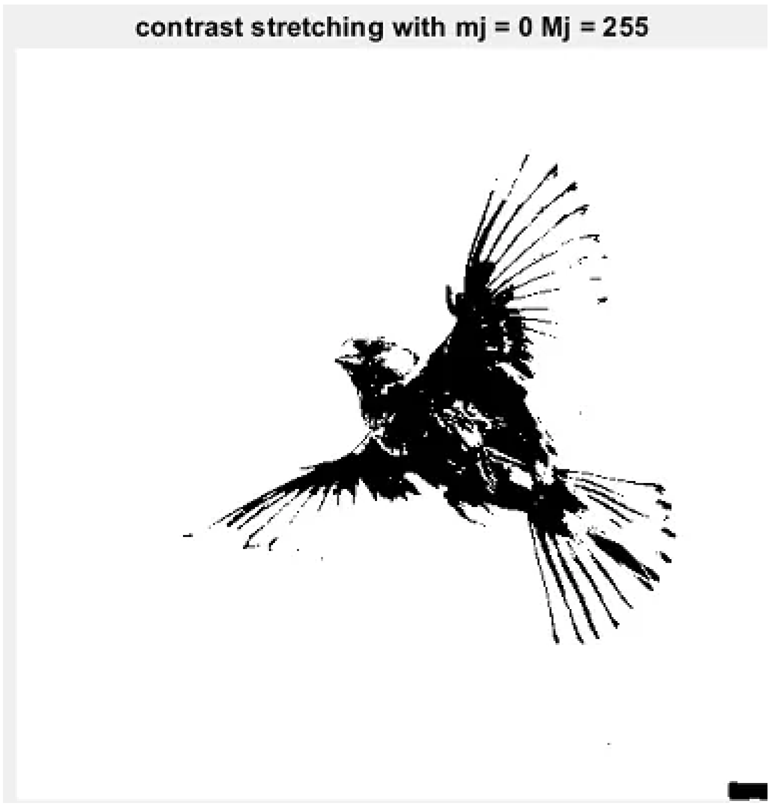

contrast adjusting with a = 2 s = 25
contrast adjusting with a = 5 s = 25
contrast adjusting with a = 10 s = 25
contrast adjusting with a = 2 s = 25
contrast adjusting with a = 2 s = 50
contrast adjusting with a = 2 s = 100
contrast adjusting with a = 0.5 s = 25
contrast adjusting with a = 0.5 s = 50
contrast adjusting with a = 0.5 s = 100

"Image by Author"

One of the operations used to change the contrast settings of the picture is **contrast stretching**. It changes the distribution range and ratio of the pixel values in the picture so that the histogram of the picture (which will be discussed in detail in the next subsection) has a more proportional spread. In summary, it can be expressed as balancing rather than increasing or decreasing contrast. This process is applied with the formula below.

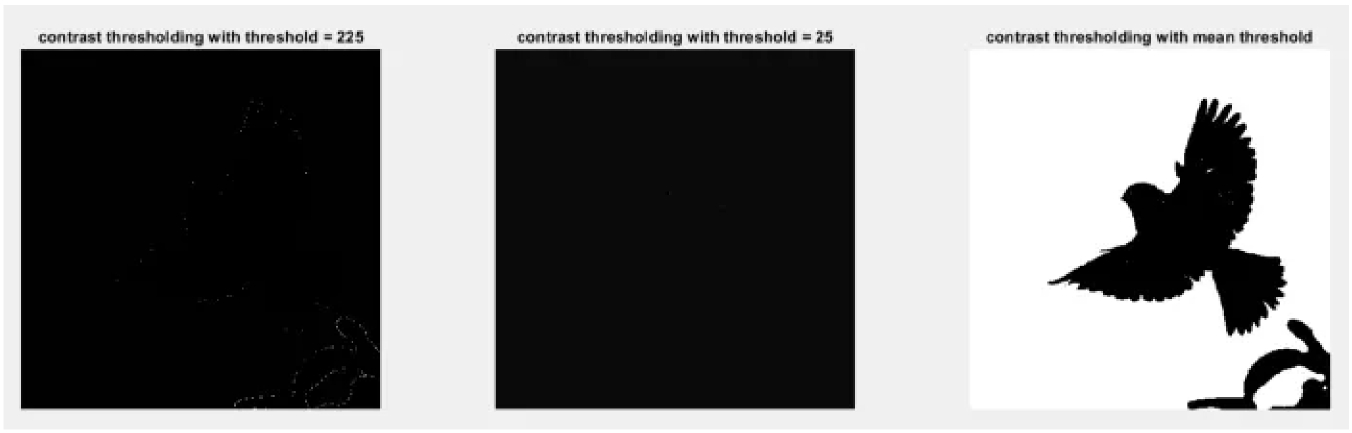
$$\mathbf{J}(r,c) = (M_{\mathbf{J}} - m_{\mathbf{J}}) \frac{\mathbf{I}(r,c) - m_{\mathbf{I}}}{M_{\mathbf{I}} - m_{\mathbf{I}}} + m_{\mathbf{J}}$$

mı : minimum pixel value in the picture
Mı: maximum pixel value in the picture
mj: the minimum value a pixel can have in the output picture
Mj: the maximum value a pixel can have in the picture
I (r, c): any pixel value in the picture



“Image by Author”

Another operation is **contrast thresholding.** In this process, a fixed value is selected as the threshold and by assigning 1 to values higher than this threshold and 0 to the lower values, high contrast binary output pictures are obtained. There are 2 commonly used methods to determine the threshold value. The first is simply to choose a fixed number (global thresholding) and the second is to use the average of the pixel values as the threshold (mean global thresholding).


contrast thresholding with threshold = 225
contrast thresholding with threshold = 25
contrast thresholding with mean threshold

“Image by Author”

You can access the relevant Matlab code from the gray_level_transformations link to apply these processes yourself and examine the outputs obtained with different values.

5. Histogram Processing

The histogram of a picture is a discrete function with the x-axis grayscale level and the y-axis the total number of pixels with the corresponding grayscale bit in the image. While histograms give us an idea about various issues related to the image we have, they form the basis of many operations performed in the spatial domain.

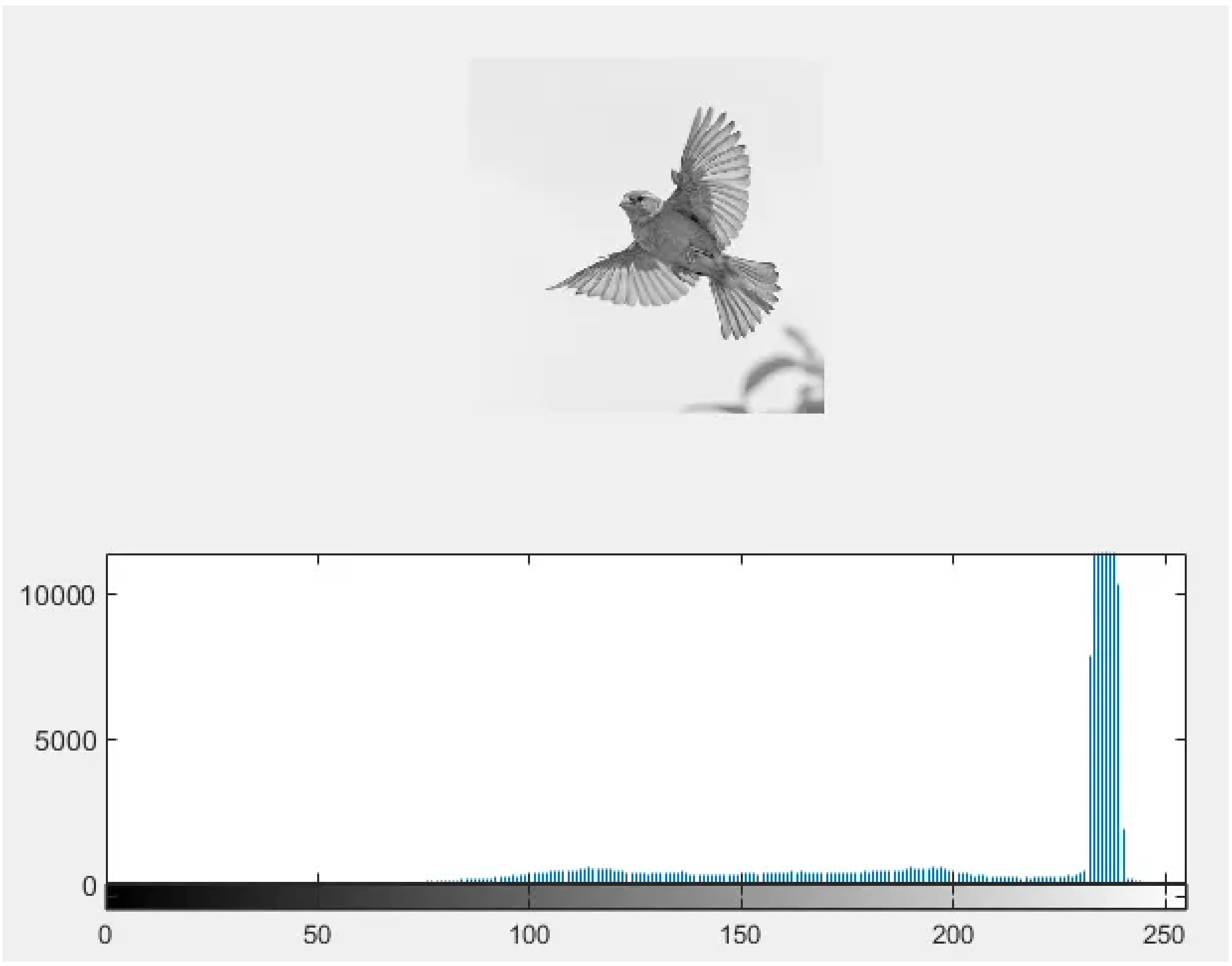


Figure 9 "Image by Author"

In Figure 9 we see the histogram of a black and white (1-channel) picture. The grayscale level of this picture is 8 bits, that is, its pixels can take values between 0–255. So the x-axis is in the range of 0–255, while the y-axis consists of numbers describing how many pixels are from each value in this range. For example, in this picture, we see that there are no pixels with a value of 0, whereas there are more than 1500 pixels with a value of 14.

!!! After obtaining these figures in MATLAB, you can navigate on the coordinate axis and examine the values of the (x, y) pair at the point you want by clicking on it.

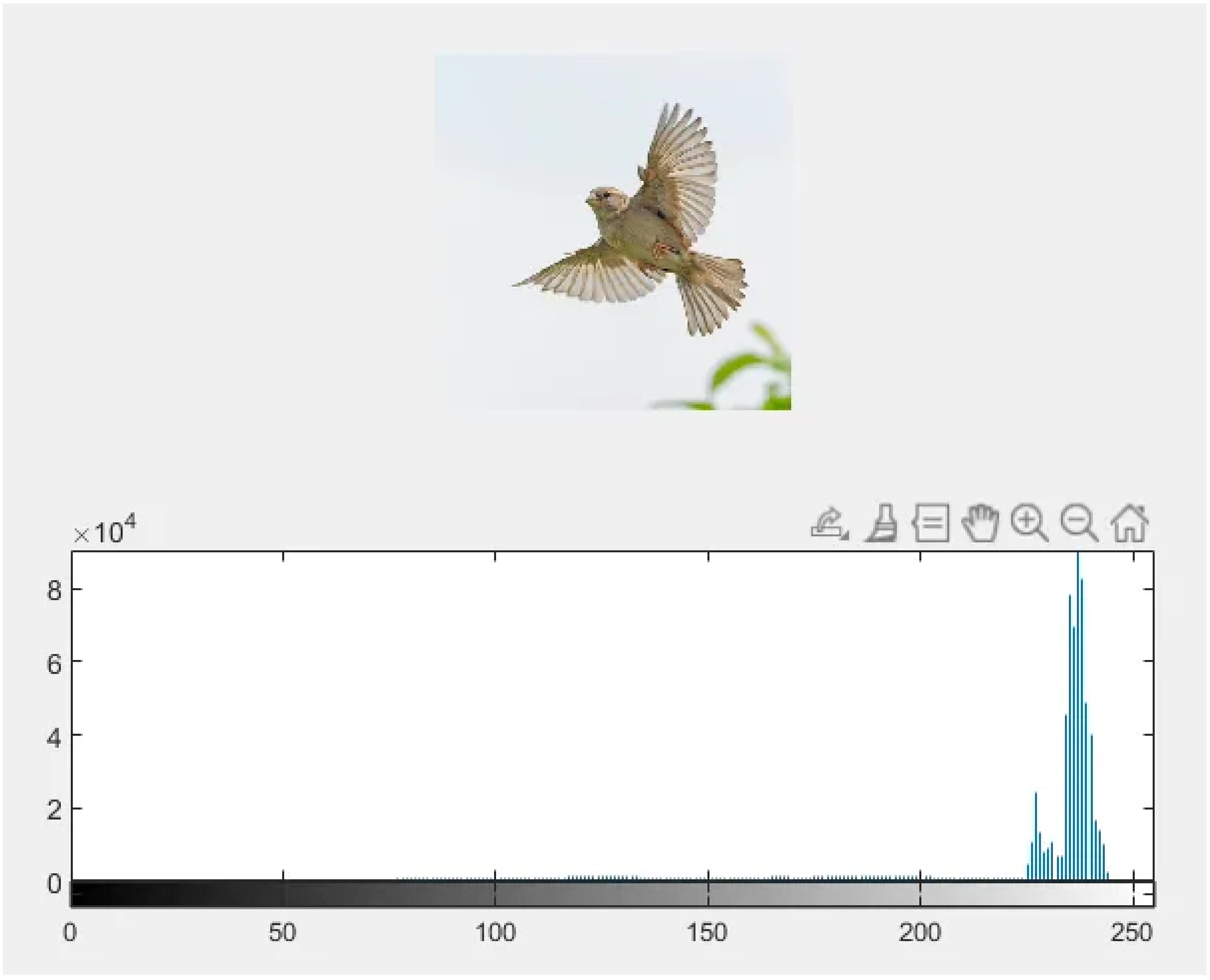
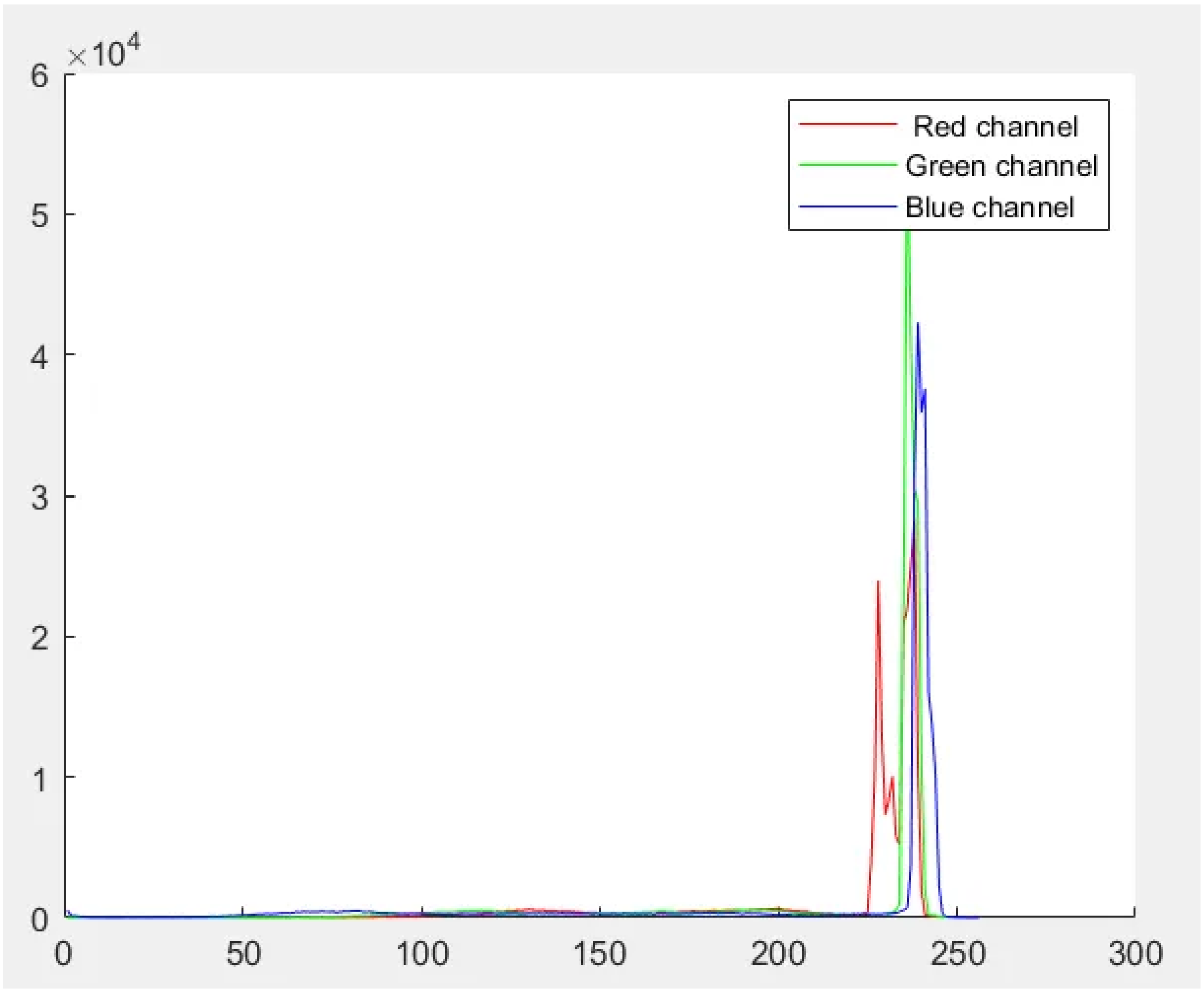


Figure 10 “Image by Author”

In Figure10, we see the histogram of an RGB (3-channel) picture. The grayscale level is again 8 bits. The difference from the black and white picture above is that the numbers that make up the y-axis are separately calculated and collected for the pixels in 3 different channels. For example, the value of x = 69 y = 10780 tells us that this picture has a total of 10780 pixels with a value of 69 in 3 different channels.

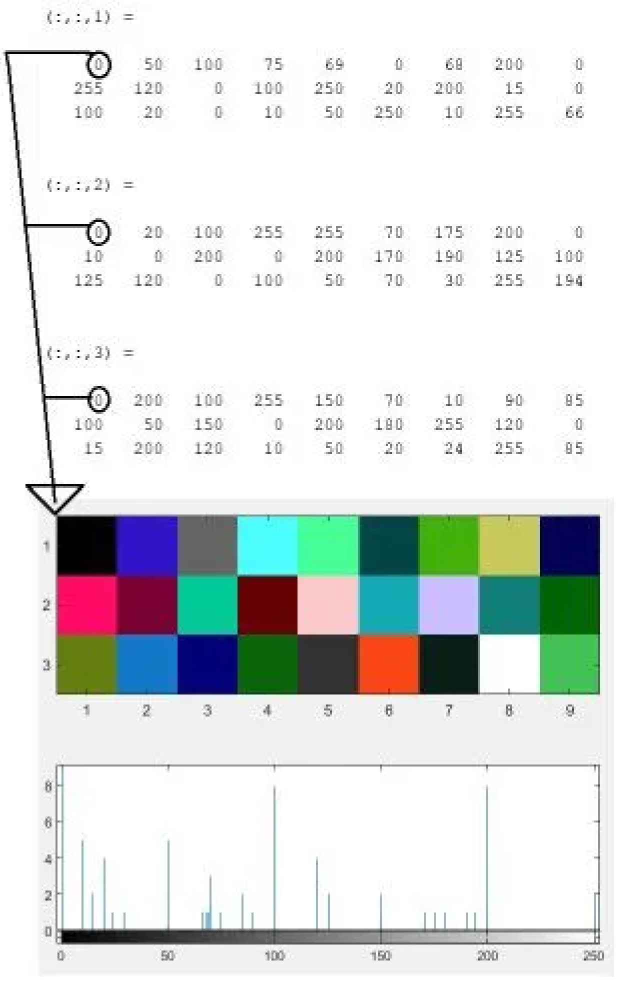


“Image by Author”

It is also possible to view the histograms of each channel separately in RGB pictures. So in the above image, we see the total number of each channel within itself, not the total number of pixels of 3 channels on the y-axis.

RGB Images

Color pictures, which we are used to seeing more than black and white pictures in daily life, are actually images where 3 different matrices come together to create 1 digital picture. All of the different colors perceived by our eyes are actually a mixture of 3 basic colors: red, green, and blue. RGB pictures are the pictures that have a separate matrix for these 3 main colors, and the pixel value in the x position is obtained from the mixture of the values in these 3 matrices. We said in black and white pictures, a pixel getting closer to 0, approaches black, and a pixel getting closer to 255, approaches to white. For 3-channel images, in the red channel, a pixel getting closer to 0, approaches black, and a pixel getting closer to 255, approaches to red; in the green channel, a pixel getting closer to 0, approaches black, and a pixel getting closer to 255, approaches to green; in the blue channel, a pixel getting closer to 0, approaches black, and a pixel getting closer to 255, approaches to blue. In the mix of these 3 channels, whichever matrix has the highest value for the relevant pixel, that color will be dominant in that pixel. We see that the image pixel is close to white where 3 colors have the same high-value pixels, and where the 3 colors have the same low-value pixels, the image pixel is closer to black.



```
(:,:,1) =

     0    50   100    75    69     0    68   200     0
   255   120     0   100   250    20   200    15     0
   100    20     0    10    50   250    10   255    66

(:,:,2) =

     0    20   100   255   255    70   175   200     0
    10     0   200     0   200   170   190   125   100
   125   120     0   100    50    70    30   255   194

(:,:,3) =

     0   200   100   255   150    70    10    90    85
   100    50   150     0   200   180   255   120     0
    15   200   120    10    50    20    24   255    85
```

“Image by Author”

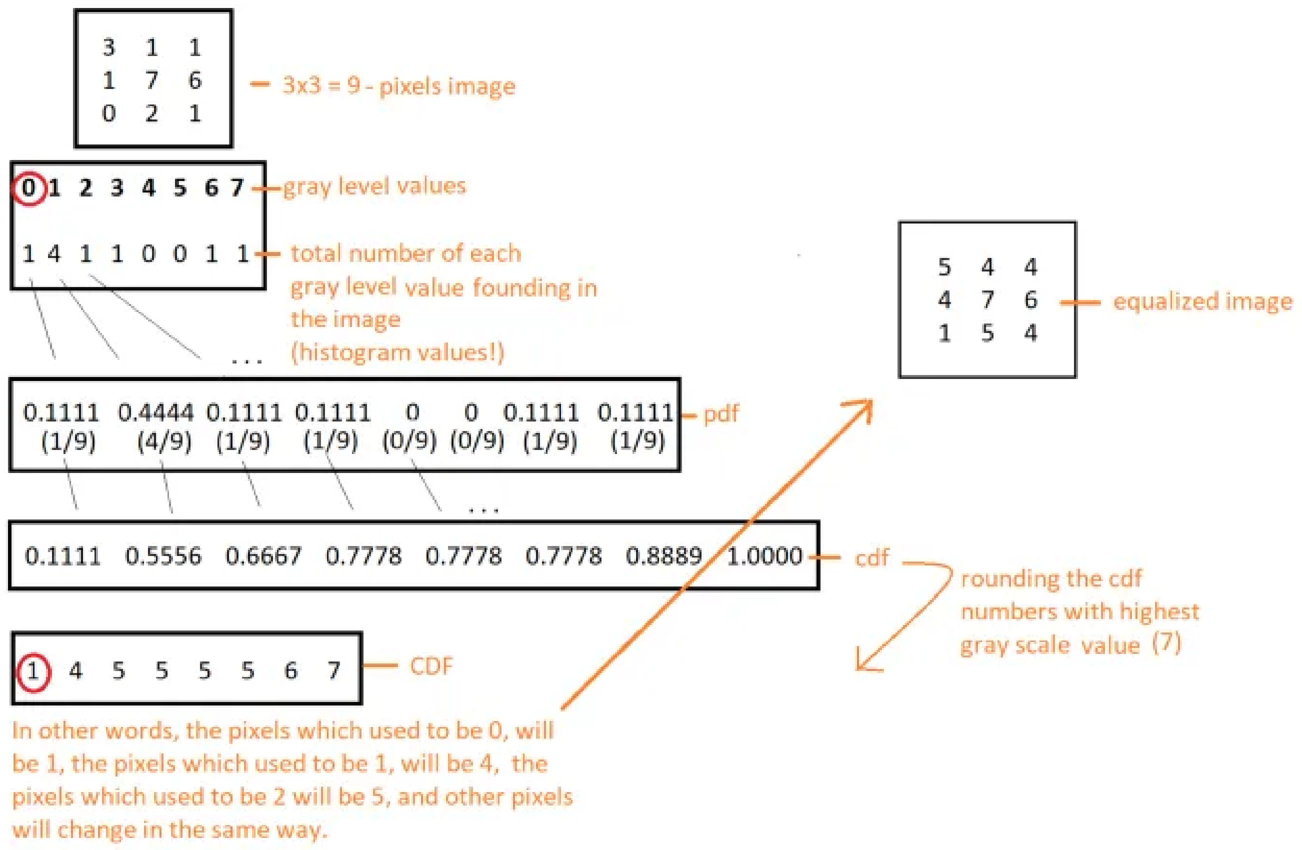
On the above picture, we see 3 different matrices of 3 x 9 size, forming the red, green and blue channels of a picture sequentially and the resulting 3x9 pixel picture. You can examine the colors created by the combination of different pixel values in 3 channels, and you can make your own experiments from the code link to be shared at the end of this section.

Histogram Equalization

It is the process of balancing the distribution in a picture’s histogram. Each pixel in the picture is passed through the steps described below, resulting in a more balanced histogram — then a more balanced picture. If the contrast distribution of the picture is already sufficiently balanced, not much change can be observed in the picture.

1. For each gray level value, the total number of pixels in the picture is calculated. (This is actually the histogram of the picture, right ?!)
2. For each gray level value, the probability of occurring in a random pixel is calculated. For this calculation, the total number of each gray level value in the previous step is divided by the total number of pixels in the picture. This calculation is called the **“probability density function”** calculation and is shown in **pdf**.
3. **“Cumulative density function”** calculation is made over probability density function values. The cumulative density value is calculated by adding the probability value of each grayscale value with the previous value, denoted by **cdf.**
4. The cumulative value of each gray level value is multiplied by the highest grayscale value and rounded to the nearest integer value to calculate the **“cumulative distribution function”** which is denoted by **CDF.**

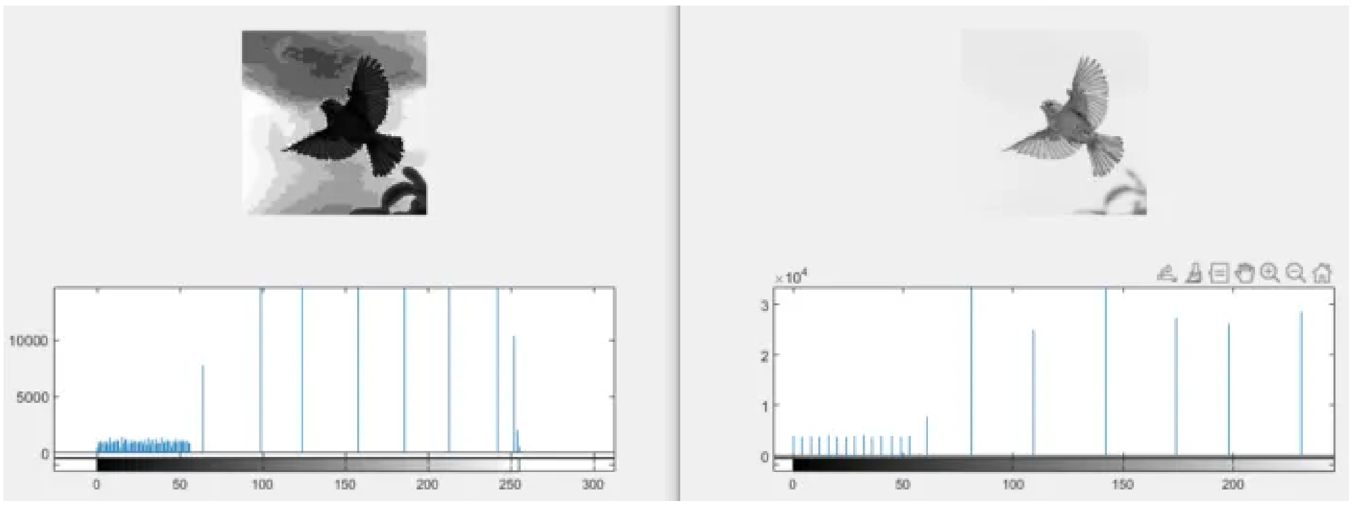
At this point, the values obtained against the grayscale values are exchanged with source grayscale values and pixel values are obtained with the balanced histogram of the new picture. Below we see a simple 3x3 picture matrix where all these steps are done one by one and how the balanced picture is obtained.



3 1 1
1 7 6
0 2 1
3x3 = 9 - pixels image
0 1 2 3 4 5 6 7
gray level values
1 4 1 1 0 0 1 1
total number of each gray level value founding in the image (histogram values!)
0.1111 0.4444 0.1111 0.1111 0 0 0.1111 0.1111
(1/9) (4/9) (1/9) (1/9) (0/9) (0/9) (1/9) (1/9)
pdf
0.1111 0.5556 0.6667 0.7778 0.7778 0.7778 0.8889 1.0000
cdf
rounding the cdf numbers with highest gray scale value (7)
1 4 5 5 5 5 6 7
CDF
In other words, the pixels which used to be 0, will be 1, the pixels which used to be 1, will be 4, the pixels which used to be 2 will be 5, and other pixels will change in the same way.
5 4 4
4 7 6
1 5 4
equalized image

“Image by Author”

Figure 11 shows our initial image and its histogram, followed by the output (equalized) image histogram and image. You can access the relevant Matlab code from the histogram_processing link to experiment with your own sample matrices or pictures and to examine the functions of these steps. In the code in this link, there are functions that contain all these steps one by one I have prepared. I suggest you examine these functions. Apart from this, using MATLAB’s functions, equalization is done on the same picture and the results are displayed as picture + histogram on the screen. You can examine both ways of use.

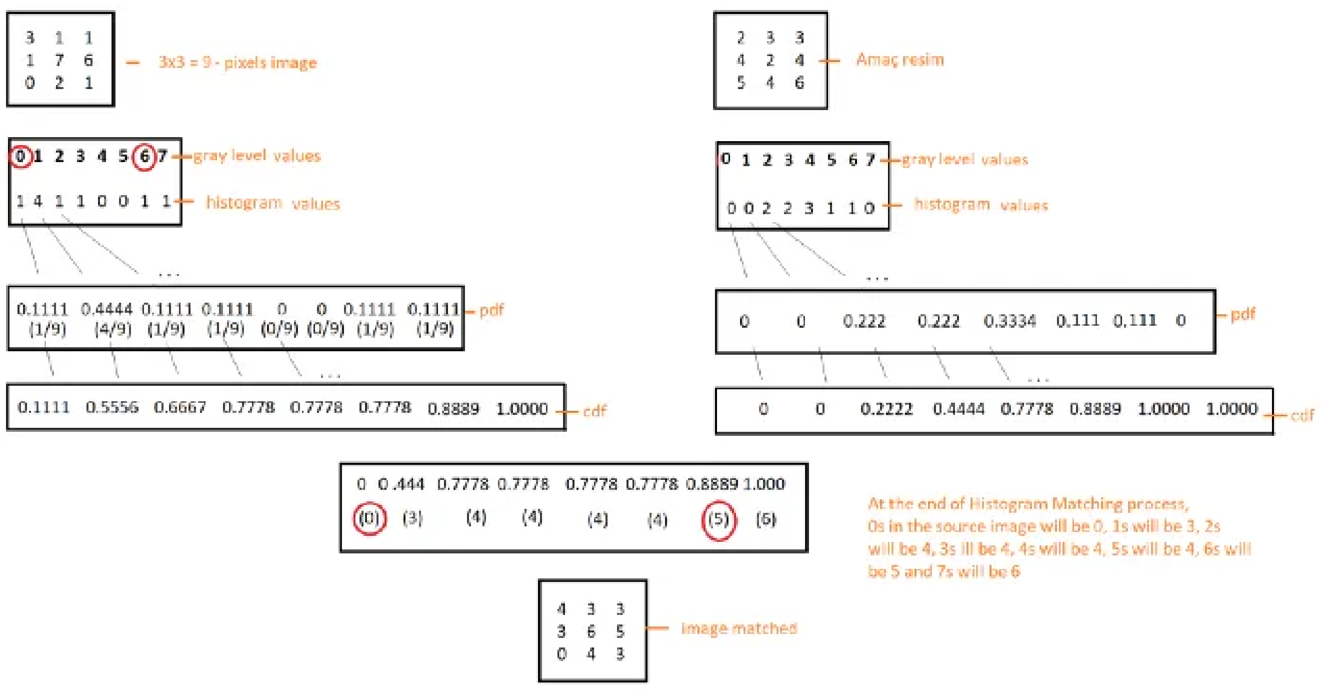


“Image by Author”

Histogram Matching

If the histogram of a picture is to be likened to the histogram of a target picture instead of just balancing, this is histogram matching. The main difference of Histogram Matching from Histogram Equalization is that after the pdf and cdf calculations are done for both pictures, instead of multiplying the source picture's cdf values with the highest grayscale value, it is matched with the closest value in the target picture's cdf. Then it is looked from which gray level value this value comes and that value is exchanged with the gray level value in the source image.

Below we see a simple 3x3 picture matrix where all these steps are done one by one and the matched picture is obtained.



“Image by Author”

Figure12 shows the matched picture and histogram obtained as a result of our histogram matching application where bird.jpg is the source and pepper.jpg is the target picture. (Each histogram belongs to the image shown above)

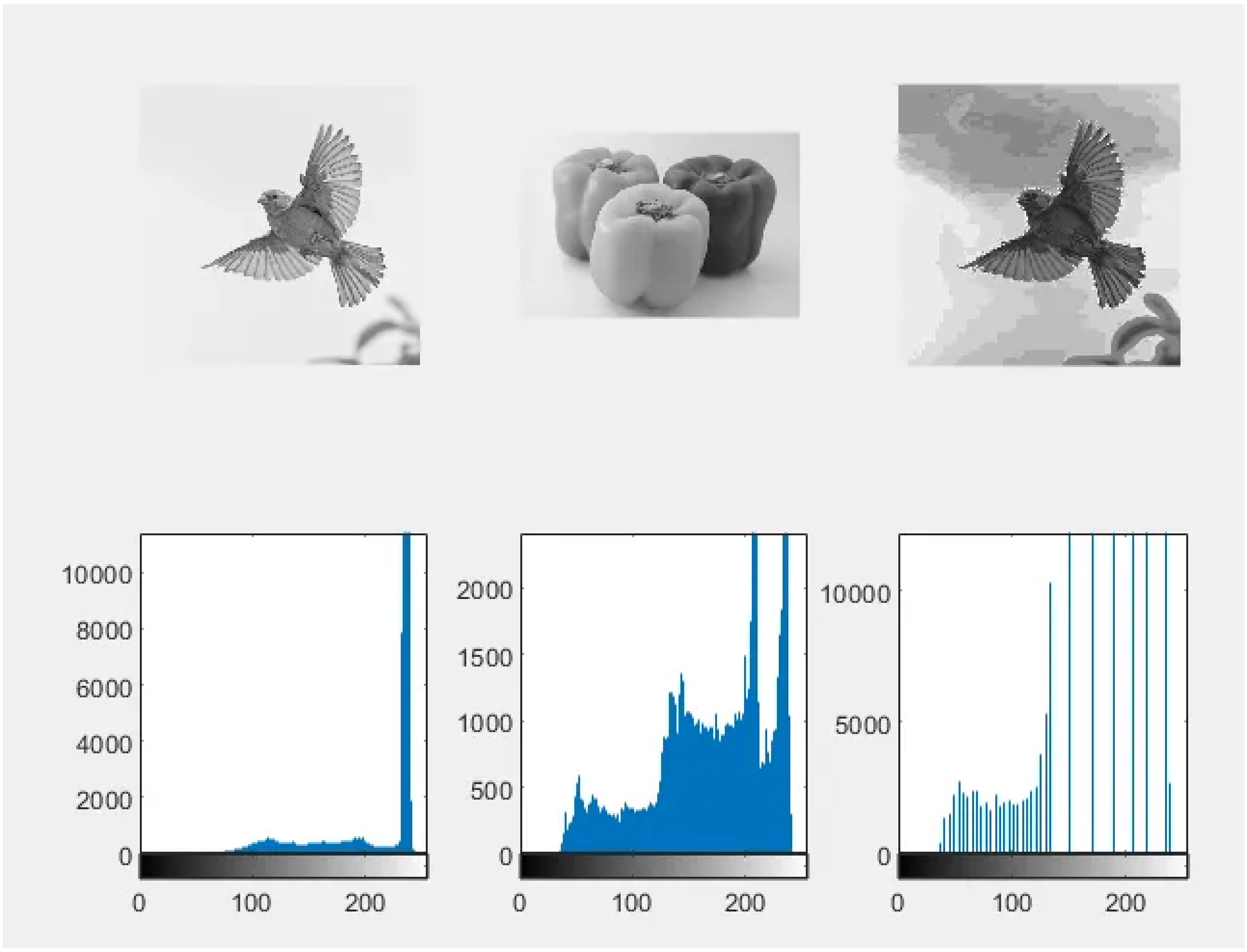


Figure 12 “Image by Author”

You can access the relevant Matlab code from the histogram_processing link to experiment with your own sample matrices and pictures and to examine the functions of these steps. In the code in this link, there are functions that contain all these steps one by one I have prepared. I suggest you examine these functions. But apart from that, using Matlab’s functions, matching on the same picture is made and the results are displayed as picture + histogram on the screen. You can examine both ways of use.

Congratulations! You have completed Image Processing Part 1. See you in the next sections! For your questions and suggestions, you can reach me from aktas.yagmur@gmail.com and you can also download the source codes from this **github link**.

In this post, the source images used for observing different operation results are taken from unsplash.com

Image Processing Image Analysis Computer Vision Python

Image Processing Projects

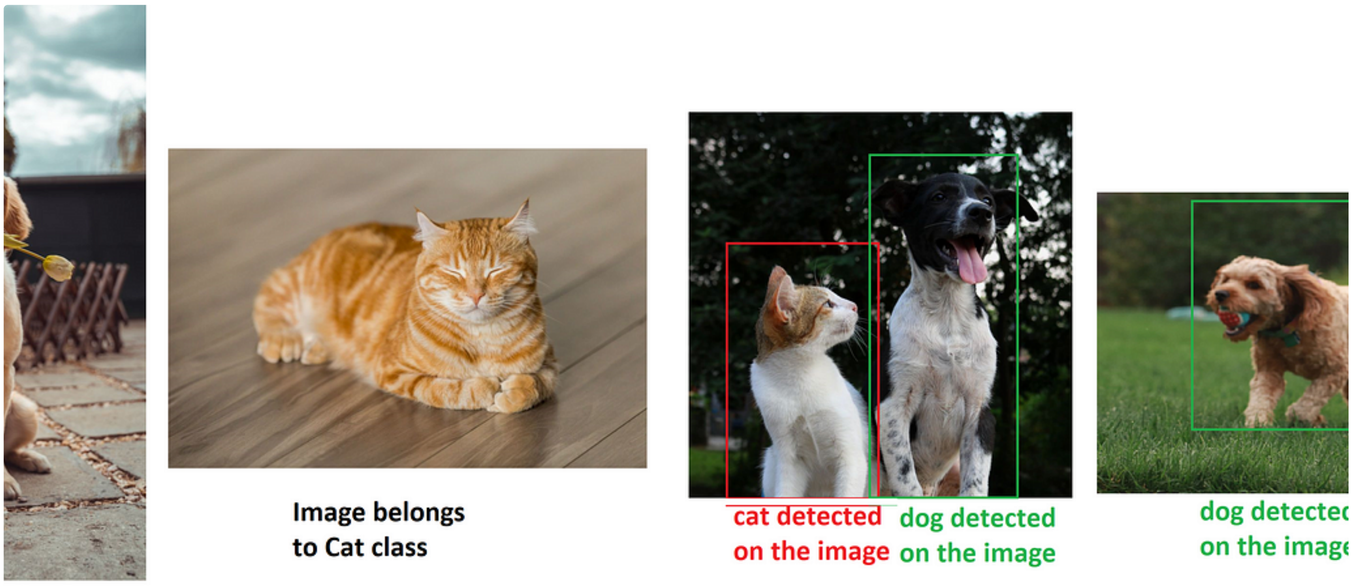
Follow

Written by Yağmur Çiğdem Aktaş

296 Followers · Writer for Towards Data Science

www.linkedin.com/in/yağmur-cigdem-aktas https://github.com/YCAyca You can reach all the codes used in my posts from this github link!

More from Yağmur Çiğdem Aktaş and Towards Data Science



Yağmur Çiğdem Aktaş in Towards Data Science

Object Detection with Convolutional Neural Networks