Ilan Sapoznikov

Liad Arvatz

Image Processing: Assignment #1

Problem 1- Sampling, Quantization:

a) The considerations for determining the number of megapixels in a camera and the resolution of computer-generated images involve a range of factors, encompassing physical, hardware, and software aspects. Here are some key factors:

Image Quality: Higher megapixels can capture more detailed and sharper images, leading to better image quality, especially when capturing fine textures or small objects. In addition, resolution in computer generated images affects the clarity and quality of visibility of the content.

Storage Space and File Size: Although higher megapixels cause better quality images, it comes at the cost of larger file sizes, which obviously leads to occupying more storage space. In scenarios where storage is a concern, for example if the storage space of the user is limited, the resolution of computer-generated images may be optimized to balance quality with storage efficiency.

Sharing and Transfer Files: When sharing or transferring files, opting for lower-resolution images becomes advantageous for swift distribution across networks or online platforms. This becomes particularly crucial in applications such as social media or when sending email attachments, where the priority is speedy transmission rather than maintaining the highest image quality. Choosing lower resolution facilitates faster uploads and downloads, ensuring a seamless and efficient sharing experience.

Display Device: The resolution of computer-generated images is shaped by the capabilities of display devices. In essence, when dealing with high-resolution displays, there is a need for computer-generated images to match this heightened resolution to ensure optimal clarity and visual fidelity. The relationship between the resolution of the generated images and the display capabilities is critical in delivering a visually satisfying and detailed viewing experience.

Sensor Size: The size of the camera sensor is a critical factor. Larger sensors can accommodate more megapixels without sacrificing image quality. Although larger sensors also contribute to an increase in both the overall size and cost of the camera. Smaller sensors, with too many megapixels, may lead to issues such as noise and reduced low-light performance.

Hardware: Both cameras and computers face critical considerations related to hardware capabilities. In the case of cameras, the processor must efficiently manage data from higher megapixel sensors, impacting crucial aspects such as burst speed, autofocus capabilities, and overall performance. On the other hand, in computer systems, particularly in real-time applications like gaming, the performance of graphic hardware, including GPUs, plays a pivotal role. Higher resolutions in computer-generated images demand more robust graphics processing units to ensure smooth rendering and an enhanced visual experience. Both scenarios underscore the importance of robust hardware to meet the demands of processing high-resolution data effectively.

b) When processing images in a digital environment, quantization is the process of converting continuous tones into a restricted set of distinct levels. This adjustment is essential due to the inherent digital nature of computer systems. Various factors contribute to deciding the extent of quantization, particularly when dealing with older computer hardware or less sophisticated screens:

Bit Depth: Bit depth refers to the number of bits used for each pixel's colour in an image. Higher bit depths offer more colour nuances but demand increased storage and processing resources. Older hardware may have limitations in achieving higher bit depths, influencing the extent of quantization and colour accuracy in images.

Storage space and File Size: Quantization affects the file size of an image; the more an image is quantified, the smaller the file becomes. In the past, older computer systems often had limited memory capacities. Strong quantization, which reduces the amount of data needed to represent an image, was beneficial in conserving memory. On the other hand, the quantization of an image needs to be balanced with the acceptable level of loss in image quality.

Display Technology: In the early stages of display technology, there were constraints in accurately representing a broad spectrum of colours or shades. Strong quantization was a practical solution to adapt images to the display capabilities and prevent artifacts or colour distortions.

Processing Power: Older hardware and less advanced screens may have faced challenges in managing the computational demands associated with high-precision images. Strong quantization helps in reducing the computational burden during image processing and display.

Problem 2- Nyquist:

a) As we know, the period of the sine wave is 2π. In this case, the function is $\sin(\pi kx)$, so in order to complete a full period of the sin function the x should be equal to $\frac{2}{k}$. The frequency is $f = \frac{k}{2}$, and therefore the wavelength is $\lambda = \frac{1}{\frac{k}{2}} = \frac{2}{k}$.

b) We calculated in section a that the frequency of the picture (f_{image}) is $\frac{k}{2}$.
Also, if A is the width of each column so full circle of the function is 2A.
As we learned, if we want to fully restore the sin image, we have to sample in Nyquist frequency so we will calculate it for each value of A:

A=2:

$f_{sample} = \frac{1}{4}$

$f_{sample} \geq 2 * f_{image} \implies \frac{1}{4} \geq 2 * \frac{k}{2} \implies k \leq \frac{1}{4}$

A=0.25:

$f_{sample} = 2$

$f_{sample} \geq 2 * f_{image} \implies 2 \geq 2 * \frac{k}{2} \implies k \leq 2$

Problem 3- Histograms, Matching, Quantization:

Final results:

```
C:\Users\liada\AppData\Local\Programs\Python\Python311\python.exe C:\Users\liada\PycharmProjects\HW1_IP\TranscribeHistogram.py
Histogram a.jpg gave 2, 2, 4, 2, 2, 6, 3, 2, 1, 1
Histogram b.jpg gave 6, 2, 1, 1, 3, 3, 6, 2, 2, 3
Histogram c.jpg gave 0, 0, 0, 0, 0, 0, 1, 1, 1, 1
Histogram d.jpg gave 1, 0, 2, 3, 4, 3, 5, 5, 6, 2
Histogram e.jpg gave 2, 1, 1, 3, 2, 5, 1, 1, 2, 3
Histogram f.jpg gave 1, 0, 1, 1, 1, 4, 1, 1, 2, 1
Histogram g.jpg gave 1, 1, 1, 3, 1, 2, 9, 3, 3, 0

Process finished with exit code 0
```

Code explanation:

First of all we ensured that we read the images properly (the images of the grade histograms and the images of the 0-9 digits). We read them from a specified directory.

Now our target is to convert the images grayscale to simplify the work. For this purpose we use the quantization function that reduces the number of colors in a set of input images using K-means clustering. It converts the images into a specified number of gray levels, in our case to 3 levels (we explain why in our answer for section e). Later in the program we converted those images to black and white based on a threshold value. We convert each element that is equal or smaller than 220 gray level to 0 (black), and each element that is larger than 220 to 255 (white).

We chose the threshold value to be 220 after we figured out that it’s close (and little bit bigger for a margin of safety) to the gray colour of the lighter columns.

Then we looped through each image of grades histogram and did the following for each one of them:

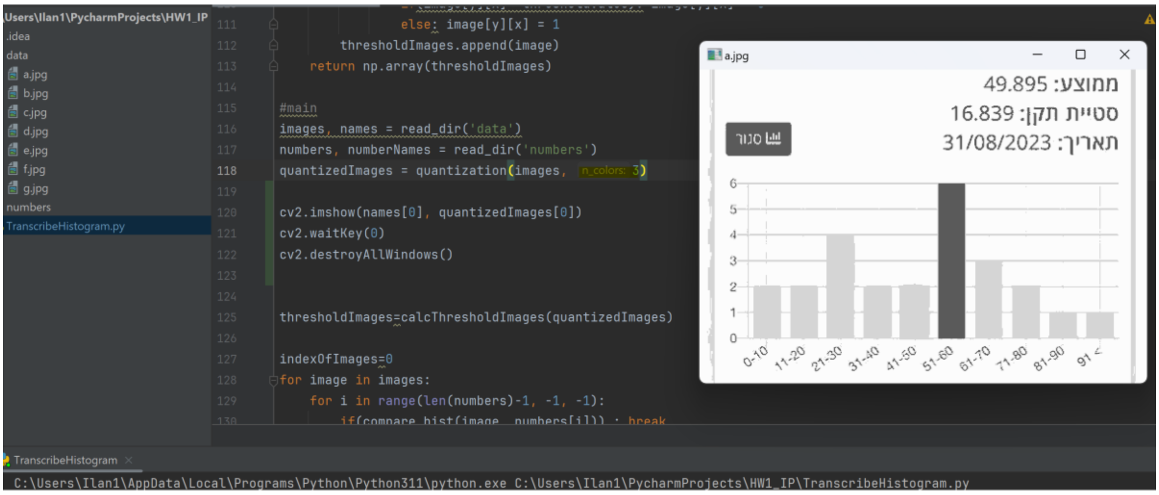
1. We loop through each image of a digit to get the max height bar in each one of them by using compare_hist with fixed specific area in the image.
2. compare_hist function divides src_image to slide windows and search in specific area in the window if EMD<260.
   Here we use the calcAccumulatedHist function that creates the pixel's histogram of the image, and then calculates the accumulated histogram and returns it.
3. After we found such window and digit we stop the inner loop and use the calsBinsHeight function. calsBinsHeight function estimates the

number of students in each bin of the histogram based on the height of the bars in the image (via get_bar_height).
We do it after using the threshold to convert the images to black and white.

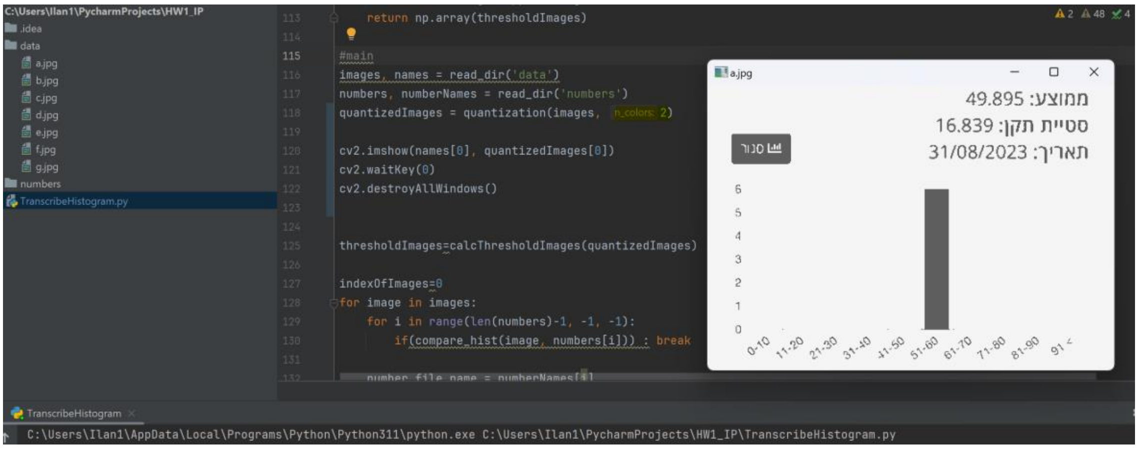
4. We print the results for the current image.

e) We want to find the threshold value of the image so we have to keep the 3 critical levels of gray that enable us to see the bins of the histogram: the gray of most of the bins, the darker gray of one of the bins and the gray-white of the background. So the minimum number of gray levels we have to keep on is 3.

This is an example of one of the images after quantization to 3 gray levels:

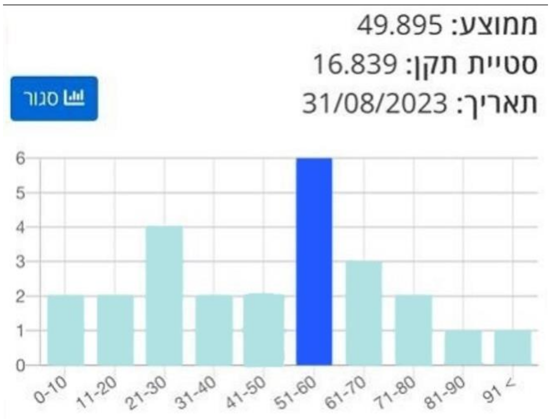


Just to make sure, when we tried switching the number of gray levels from 3 to 2, we saw that we lost critical data from the image:

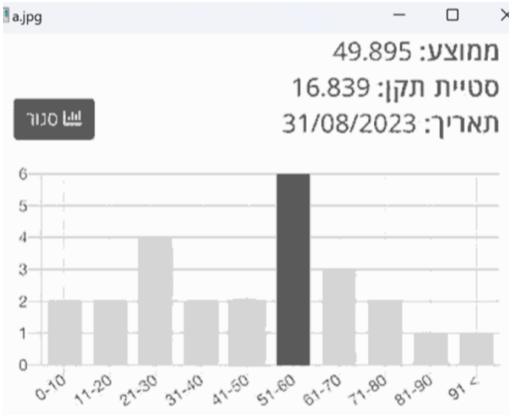


Here an example of part of the process for image a:

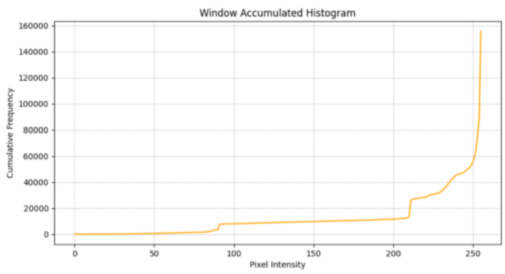
1) Getting the original image:



2) Image after quantization:



3) Accumulated Histogram of one of the numbers that we used for comparing to the numbers in the image, by using sliding window:



4) Calculating number of students per bin: