Homework 1: Digital Image Fundamentals

Giovanni Dal Lago, EmplID:031980994

4A)

```
function outputIm = subsample_image(inputIm,factor)
% after checking validiy of the inputs the function reduce the resolution
% of the imput image by a factor of "factor" eliminating omogeneusly
% elements of the inital matrix
outputIm = inputIm;
%cheking if the input is in fact a 2d matrix
if ndims(inputIm) ~= 2
error('Input must be a 2-D grayscale image.');
end
% cheking if the imput factor is in fact a valid integer
if ~isscalar(factor) || factor < 2 || factor ~= floor(factor)
error('factor must be an integer >= 2.');
end
% eliminating elements
outputIm = inputIm(1:factor:end, 1:factor:end);
% preserve class
outputIm = cast(outputIm, 'like', inputIm);
end
```

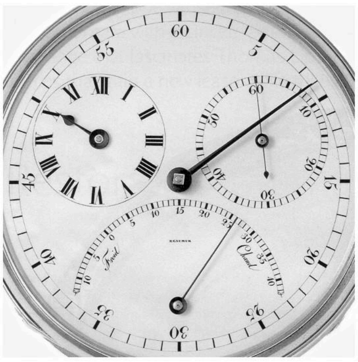
4B)

first I read the original image:
I=imread("chronometer.tif")

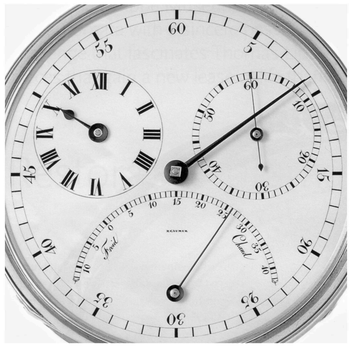
then i apply the function 5 times with factor= 2,4,8,16,32:
I_low = subsample_image(I, factor);

the results:

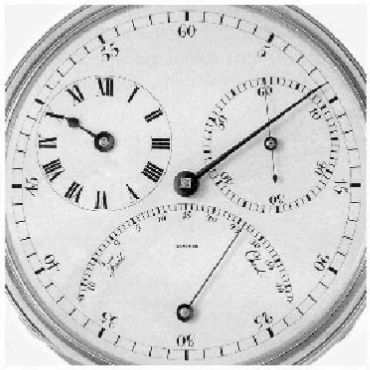
Factor = 2: The reduction is almost invisible. The overall structure is preserved and differences from the original are minimal.



Factor = 4: The quality remains good,
but fine details start to disappear.
The image is still clear and usable.

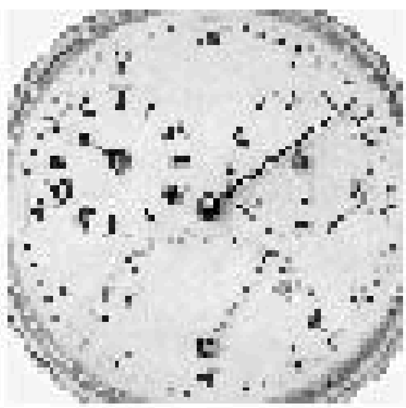


Factor = 8: The content is readable,
though sharpness is clearly reduced.
Edges lose definition and textures vanish.



Factor = 16: Numbers are barely legible and
most details are lost.
The image appears heavily pixelated.

Factor = 32: The image becomes too pixelated,
essentially unreadable and
it is hard to recognize what it represents.



Problem 5

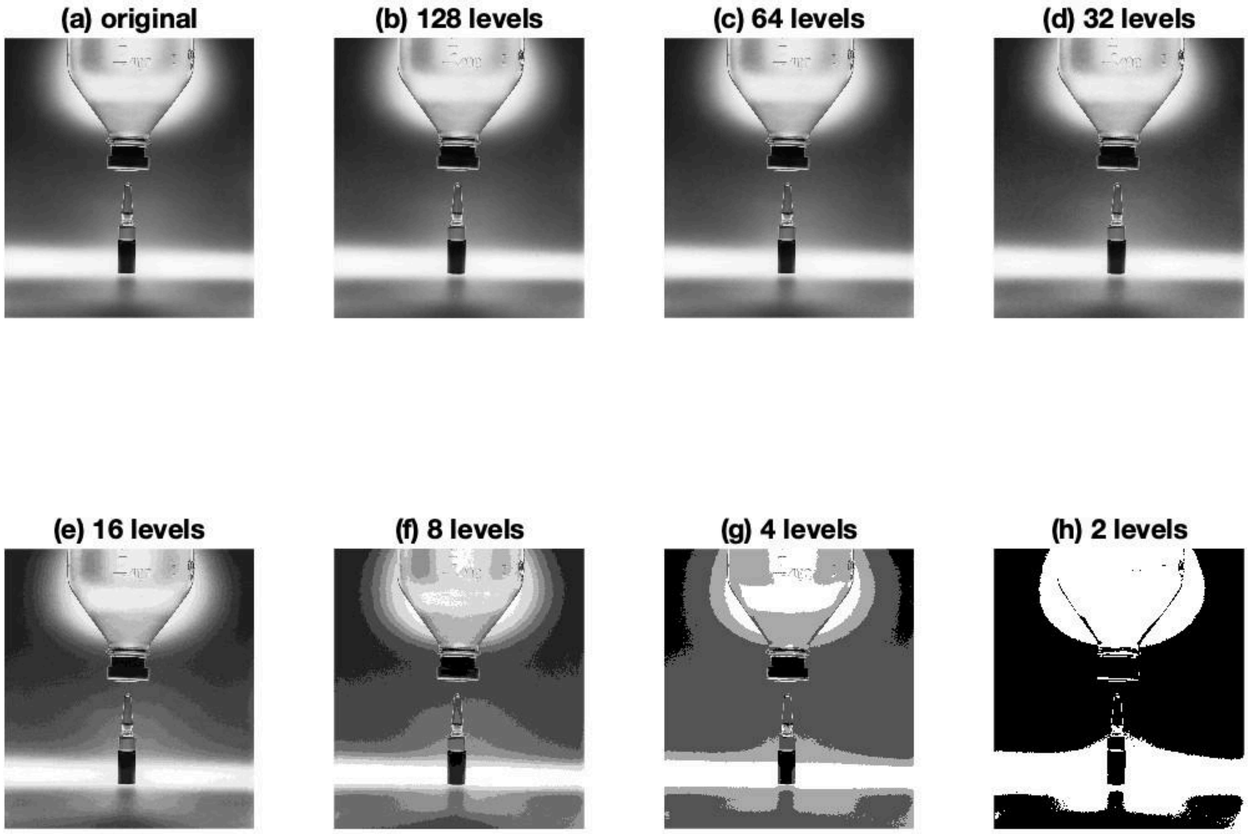
the goal is to reproduce Fig. 2.24 panels (b)–(h) by uniform intensity quantization
Panels correspond to:
(b) 128, (c) 64, (d) 32, (e) 16, (f) 8, (g) 4, (h) 2 levels.

What is produced:
Many gray levels → the image looks like the original.
Fewer gray levels → banding/false contouring in smooth regions, textures vanish.
2 levels → binary posterized result; only high-contrast edges remain.

the code:

```
%1) Load image
% Read the provided TIFF.
I = imread('drip-bottle.tif');

% Convert to double precision in [0,1]. This normalizes pixel math and display.
I = im2double(I);                          % now min(I)≈0, max(I)≈1

% 2) Defining  target levels
% L is the number of uniformly spaced intensity levels after quantization.
% These L values reproduce panels (b)–(h) from the textbook.
levels = [128 64 32 16 8 4 2];
panel_tags = {'(b)','(c)','(d)','(e)','(f)','(g)','(h)'};  % labels for titles

% 3) Uniform scalar quantization
% For each L:
%   - Scale [0,1] → [0, L-1] so each integer represents a quantization bin.
%   - ROUND implements a nearest-level quantizer:
%        q = round(I*(L-1));
%     This minimizes bias compared to FLOOR (which always biases downward).
%   - Rescale to [0,1] so imshow/imwrite interpret intensities correctly.
Qs = cell(numel(levels),1);
for k = 1:numel(levels)
   L = levels(k);
   q = round(I*(L-1));                  % integer codes 0..L-1 (nearest level)
   Qs{k} = q/(L-1);                     % map codes back to [0,1] representatives
end

%4) Visualize grid
```

```
% using a tight 2x4 layout: original in the first tile, then (b)–(h).
figure('Name','Intensity Quantization','Color','w');
tiledlayout(2,4,'Padding','compact','TileSpacing','compact');
nexttile; imshow(I,[]); title('(a) original');

% Quantized panels with explicit labels and level counts
for k = 1:numel(levels)
   nexttile; imshow(Qs{k},[]);          % [] to autoscale
   title(sprintf('%s %d levels', panel_tags{k}, levels(k)));
end
```



Visual effect by level:

-128/64: visually identical.
-32: very very fine banding appear in smooth regions.
-16: false contouring clearly visible in smooth areas.
- 8: strong banding, loss of fine textures.
-4: heavy posterization, only coarse tones remain.
-2: bilevel image; everything is either full black or full white.