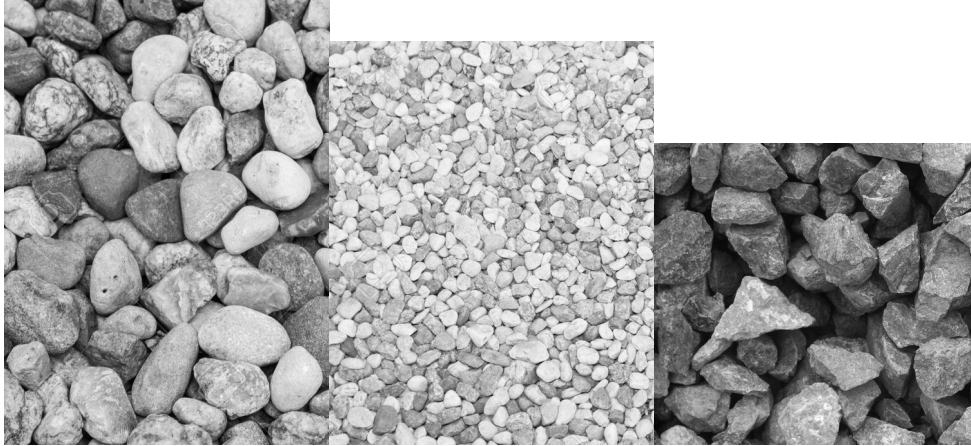


### Homework 3 (v1)

1) Implement the morphological hole filling technique shown in class (in C/C++ or Java), and apply it to the provided image: teker.png.

2) You'll find attached an aerial image of the city of Pompeii, Italy. It is corrupted with sinusoidal noise due to electrical interference. Examine its Fourier transform and implement (in C/C++ or Java) an appropriate band reject filter to denoise it.

3) You have graduated and you now work for a computer vision company (GörüBizimİşimiz A.Ş.); your latest client is a cement production company (BetonRıza A.Ş.). Mr. Rıza tells you that their company excavates gravel (çakıl) of 3 different types from the same site:



1

2

3

and they need an effective system to tell them apart automatically. In other words, the final software/system is supposed to take an image of the gravel (with a camera), and guess the gravel type as accurately as possible.

Mr. Rıza provides you with 1 sample image (labeled-1.png, labeled-2.png, labeled-3.png) from each gravel type. And also gives you 1 unlabeled sample from each (unlabeled-A.png, unlabeled-B.png, unlabeled-C.png). Your task is to develop a program that, when it is given as input an unlabeled gravel image, it will print on screen its type (type: 1, 2 or 3).

Upon hearing this, you immediately remember granulometries from that silly image processing course, and how they could be useful for distinguishing one image from the other. But, there are so many questions, should you use openings or closings? Or both? What shape/size of structuring elements should you use? Should you use openings/closings or openings/closings by reconstruction?

As if that wasn't enough, say you calculate the granulometry of an unlabeled image, how are you supposed to compare it against the granulometry of a labeled image? Should you rely on the Euclidean distance between granulometries? Or their Manhattan distance? Or Chebyshev distance?

Design and implement a solution for this problem **based on granulometries** and explore the aforementioned questions; DO NOT use weka, or any other machine learning library. You are free to use any technique you want, as long as you code it yourself.

Good luck.

Q1: 20 points

Q2: 40 points

Q3: 40 points