Median in Soil_segmentation.py

1) NON VA BENE PER IMG PICCOLE

PERCHÉ? ⇒ NON È UN VERO E PROPRIO MEDIAN FILTER

2) VA BENE PER BLOCK SHAPES SOTTO MULTIPLI DELLE DIM. DELL'IMG DI INPUT

esempio

⇒ INPUT_SHAPE = (3648, 5472, 3) # input shape of the image

⇒ Block_SHAPE = (32, 32, 1)

⇒ VIEW = view-asblock(in_img, Block_SHAPE)

VIEW_SHAPE = (114, 171, 3, 32, 32, 1)

3648/32 5472/32 3/1

BLOCK_SHAPE

[x, y, z, i, j, k]

OGNI [x, y, z] —(punta)→ [i, j, k]

[3, 4, 1] → è un blocco di (32 × 32 × 1)

Per cui VIEW_SHAPE È COME SE FOSSE "COMPOSTO"

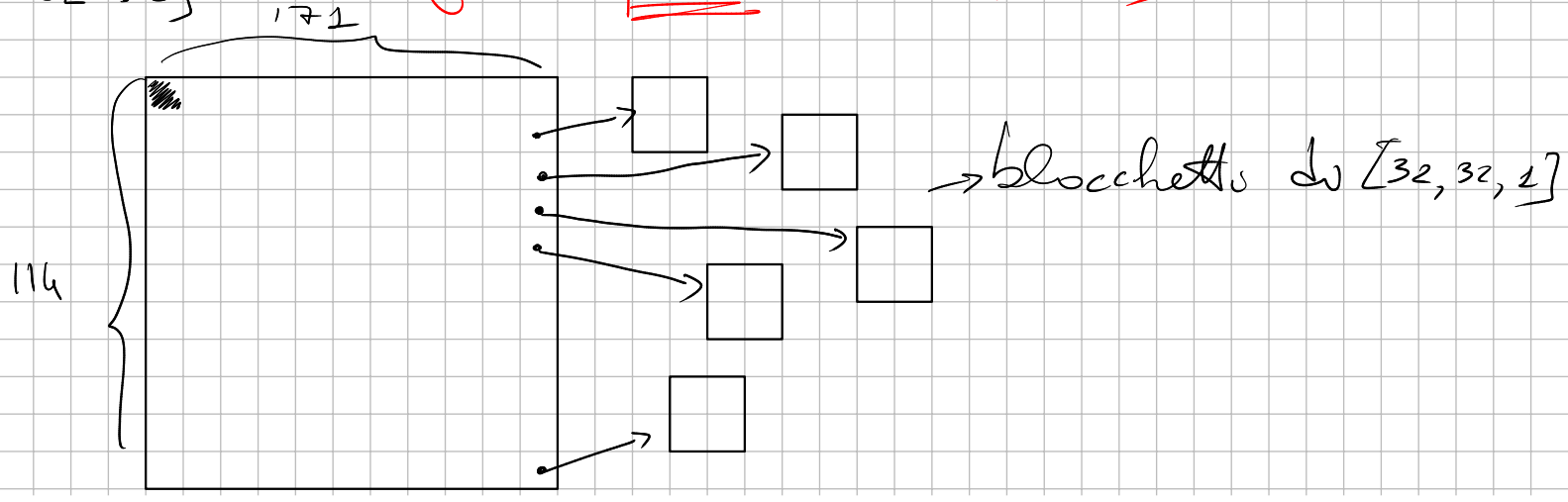
2 ARRAY

A1 [114, 171, 3] A2 [32, 32, 1]

primo secondo

A1[::z]

ogni el. punta a [32, 32, 1]



=> FLATTEN_VIEW = VIEW.RESHAPE(VIEW.SHAPE[0], VIEW.SHAPE[1], VIEW.SHAPE[2], -1)

FLATTEN_VIEW_SHAPE = (114, 171, 3, 1024)

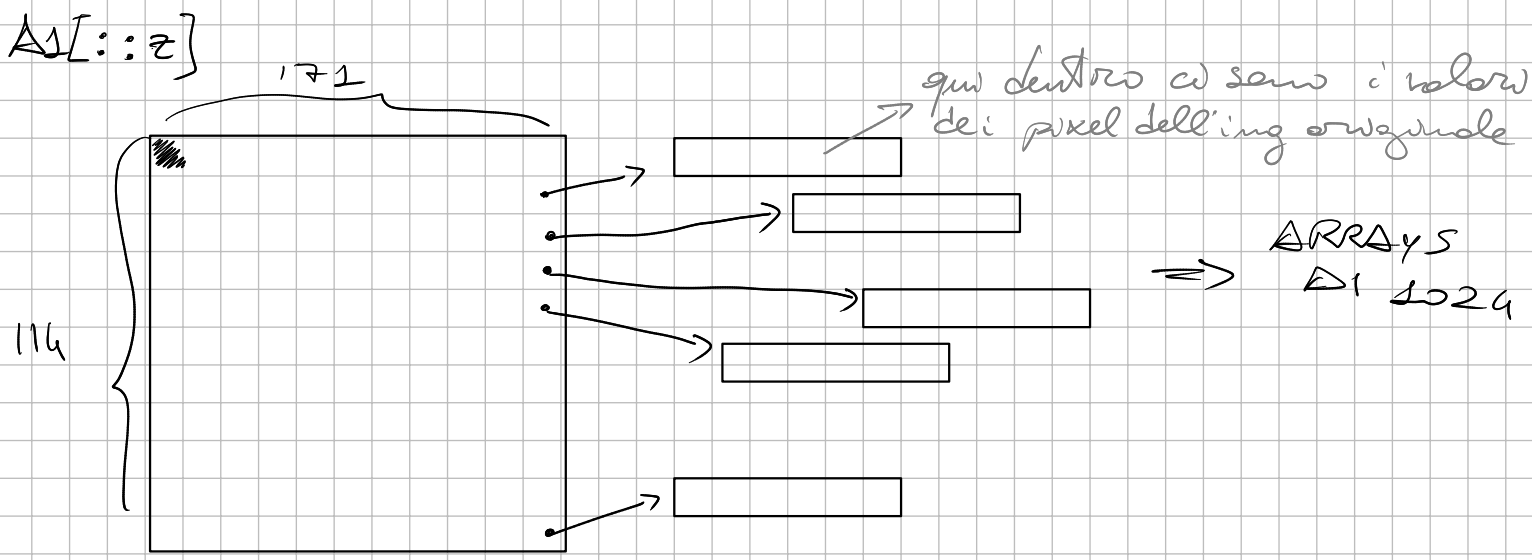
↓

A2 di VIEW È COLLASSATO IN BLOCCHI DI 1024

↓

I KERNELS

ORA A1 punta a A2 flattened (kernel)



=> MEDIAN_VIEW = NP.MEDIAN(FLATTEN_VIEW, AXIS=3)

MEDIAN_VIEW_SHAPE = (114, 171, 3)

↓

L'IMMAGINE MEDIATA con RISOLUZIONE ABBASSATA

↳ questo consente di avere un immagine con meno informazioni

↳ LA COMPUTAZIONE È PIÙ VELOCE

PERCHÉ?

↓

QUESTO MEDIAN PROCEDE IN MANIERA LEGGERMENTE DIVERSA DAL "CLASSICO"

- NP.MEDIAN(FLATTEN_VIEW, AXIS=3) → (114, 171, 3, 1024) = (A1, 1024)

Se consideriamo A1 = (114, 171, 3) # DIM. A1 MEDIAN_VIEW ricordando che A1[x, y, z] punta a 1024 pixels

VIENE CREATA MEDIAN VIEW COME SEGUE:

```
# ogni A1[x,y,z] = median(FLATTEN_VIEW[x,y,z,:])
for(x=0; x<114; x++)
    for(y=0; y<171; y++)
        for(z=0; z<3; z++)
            #salvo il kernel
            kernel = FLATTEN_VIEW[x,y,z]   # ricordiamo che punta a 1024 el.
            #ordino il kernel
            mergeSort(kernel, 0, 1023)
            #prendo il med px
            med_px = kernel[1023/2]
            #inserisco il pixel
            median_view[x,y,z] = med_px
return median_view
```

Quindi non sostituisce ogni pixel dell'immagine originale come l'algoritmo classico che rimuove/riduce l'effetto sale e pepe

NEXT STEPS

0) ANALIZZARE PARTE OTSU → quale img input? quale output?

1) CERCARE MEDIAN FILTER PER PRE-PROCESSING FOR SEGMENTATION

↳ VERIFICARE QUALE USARE SE → "CLASSICO"
→ VERSIONE PY

2) IMPLEMENTARE IN C

3) IMPLEMENTARE IN HW