Acceleration of pills detection in blisters

HPPL course project

by Viacheslav Vasilev

Skoltech 2022

Problem description

One of the most common manufacturing defects in pharmaceutical companies is missing tablets in packages. Manual inspection of such problem may be very challenging task. To reduce possible defects, as well as save company resources, the solution to this problem can be automated using computer vision.

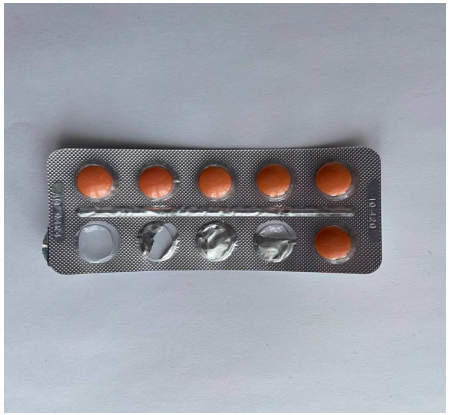
The research relevance of this task can be confirmed by the fact that one of the publications of CVPR 2020 was devoted to a similar task.

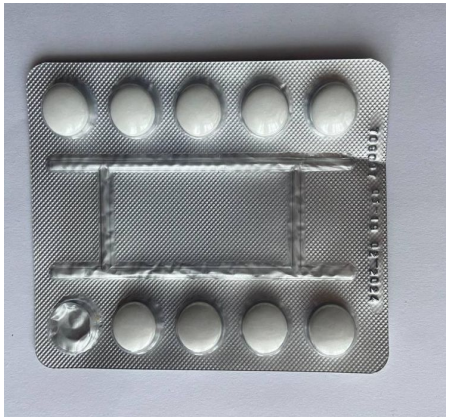
Problem statement

Input: photo of a blister

Output: number of missed pills in the blister

Data: 12 photos of blisters in different position and with different number of missed pills



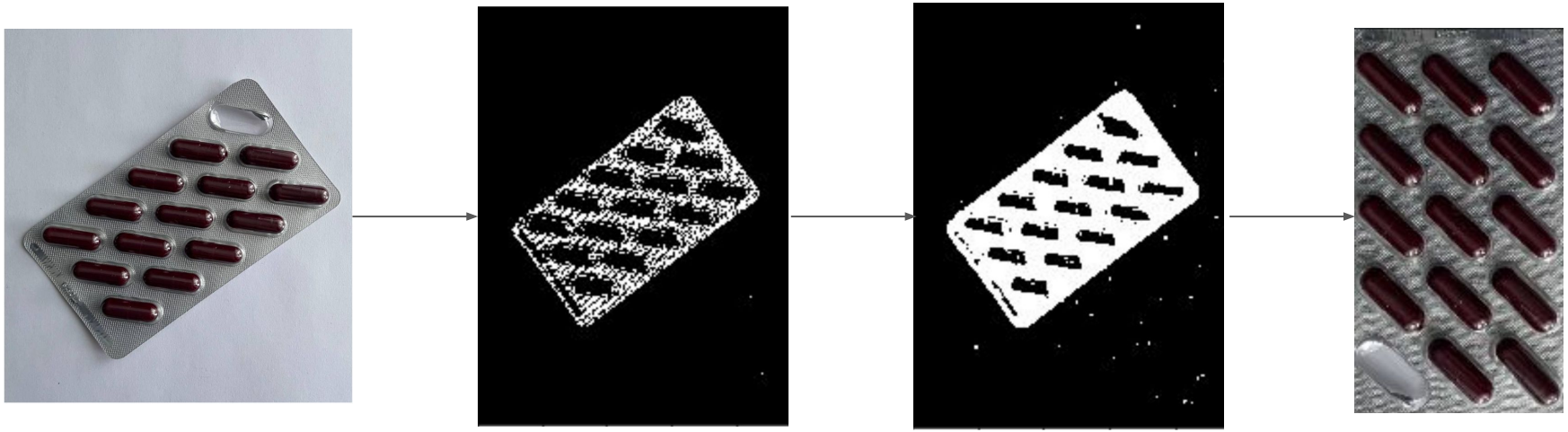


CV solution: Step 1 - detect blister

Sobel filter

Dilation

Contour with largest area + projective transform



CV solution: Step 2 - detect true pill template

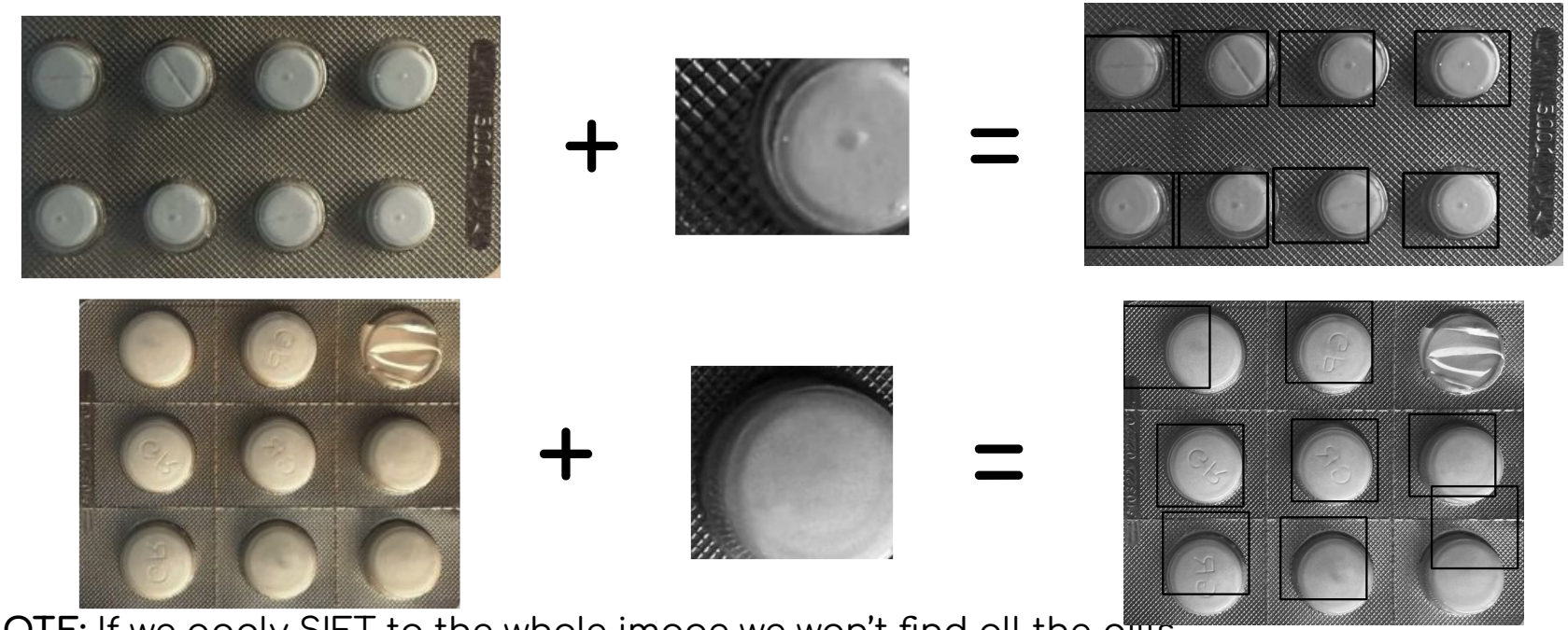
Sobel + Dilation + Inversion of black and white

Choose 3 components with largest area

Choose 1 component with smallest perimeter

Find bounding box and get template of true pill

CV solution: Step 3 - find other pills by SIFT

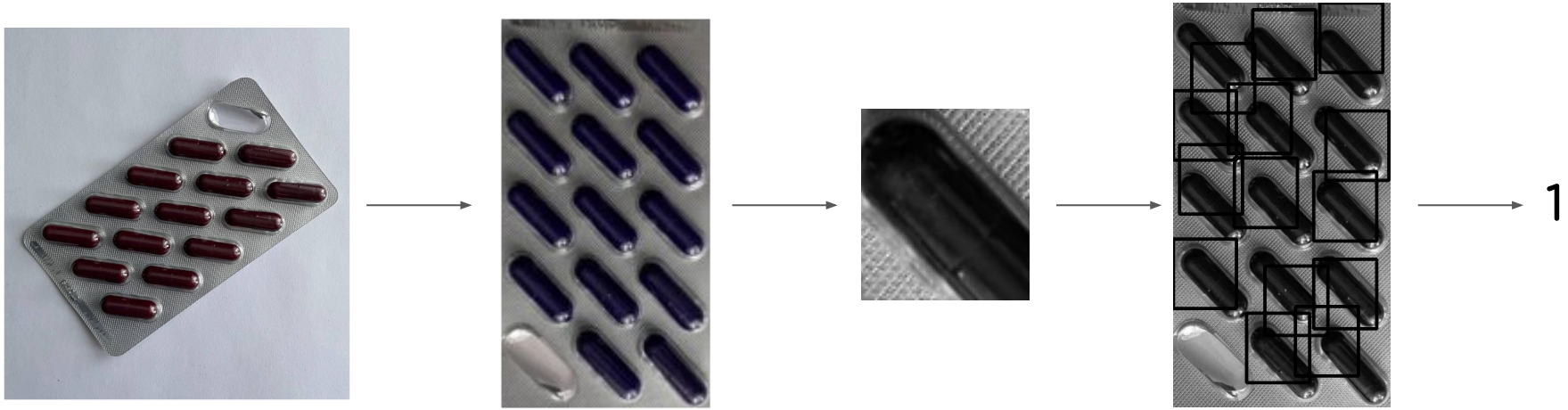


NOTE: If we apply SIFT to the whole image we won't find all the pills
SOLUTION: use **sliding window**, apply SIFT to small patches of image and then unite the results

Overall pipeline - Get them together

Step 1 (blister detection and projection) + Step 2 (true pill detection) +

+ Step 3 (SIFT) + Step 4 (count bboxes and get num of missed pills)



Overall pipeline - Problem

Step 1 (blister detection and projection) + Step 2 (true pill detection) +

+ Step 3 (SIFT) + Step 4 (count bboxes and get num of missed pills)

Overall pipeline - Problem

Step 1 (blister detection and projection) + Step 2 (true pill detection) +

+ Step 3 (SIFT) + Step 4 (count bboxes and get num of missed pills)

PROBLEM: using sliding windows with **SIFT works very slowly**

Example: 33.654 sec for one 1920x1140 image processing if **window.shape = template.shape = (171, 180)** and **stride = (171, 180) // 10**

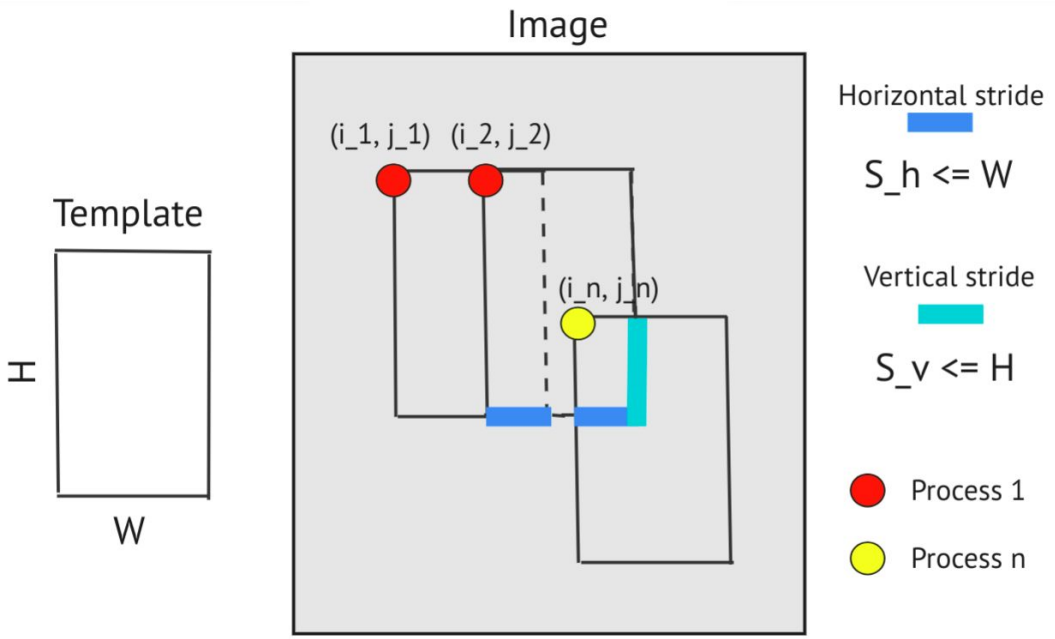
Solution - MPI parallelization

Idea: We can delegate different parts of the image to separate processes

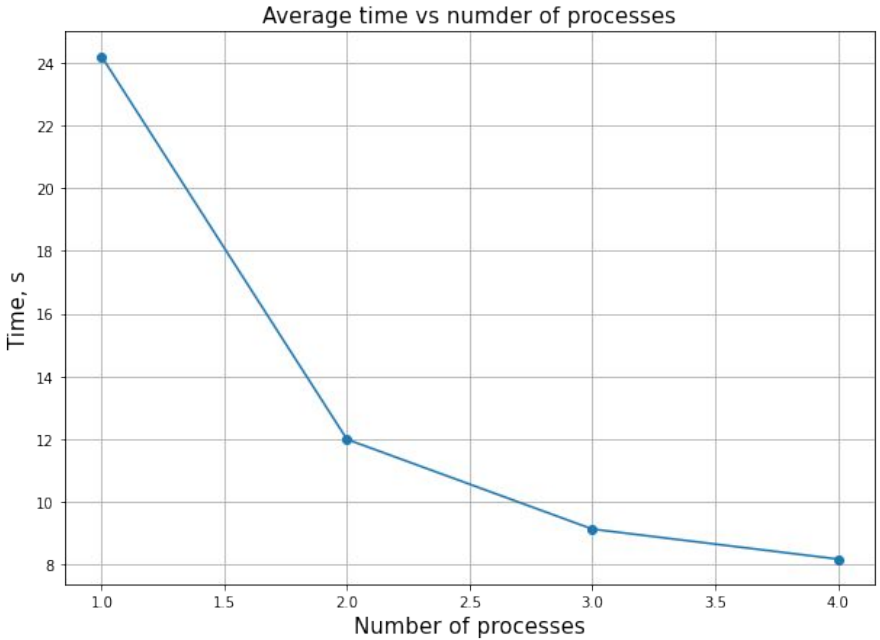
ij-pairs := {(0, 0), (0, S_h), (0, 2*S_h),

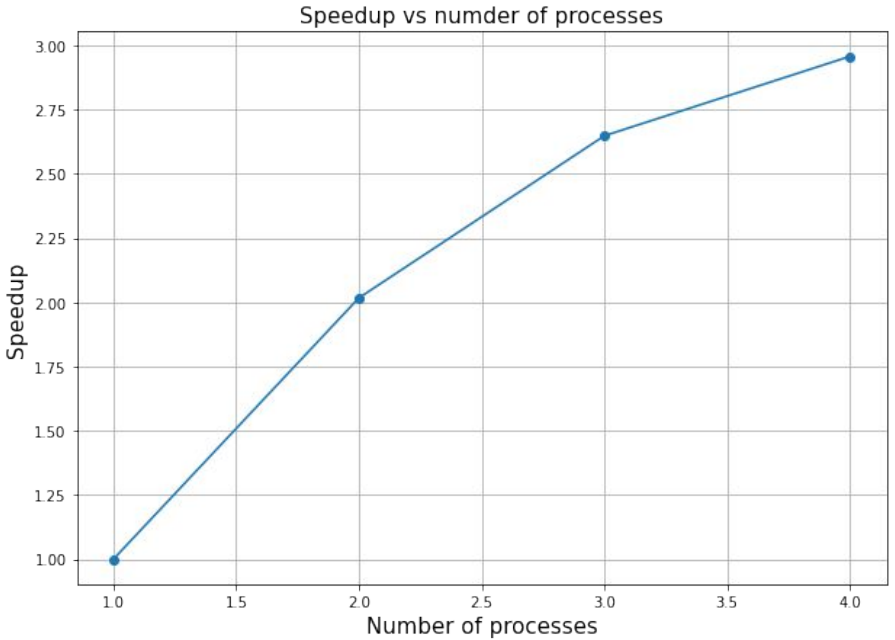
… , (i*S_v, j*S_h), …}

Then we can run SIFT algorithm for each part of ij-pairs set



Results





Processor: 1.8 GHz 4-core Intel Core i7

Thank you for your attention!

https://github.com/vivasilev/HPPL-project