“Find Viruses” Assignment Report

HORMI Sami 18079976D

What is it ?

Given an input image containing two classes of viruses, our task is to classify those viruses by means of colouring.

In that respect we define the classes as *Hole* virus for the first kind of viruses, and *Solid* virus for the second type of viruses as described in Fig (1) below.

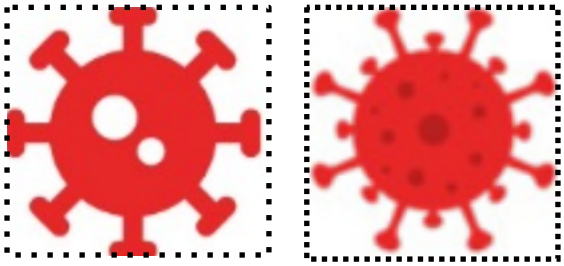

Fig 1. *Hole* Virus & *Solid* Virus

Our report will follow a progressive development of our code, from start to finish, which will serve as elements of response to the questions at hand

Questions

What is the most distinctive characteristic of these two kinds of viruses ?

Based on the structure of the two viruses, the most striking difference in appearance relates to the the **presence of holes** in the *Hole* virus class. In fact, all *Hole* viruses have two holes in them of different sizes, whereas the *Solid* viruses have **several dark circles** all throughout the body.

Additionally, several other disparities can be noted between the two classes at the level of **proteins.** In fact, the *Hole* viruses have only one kind of **T-shaped** protein surrounding the body. For each object, we count **eight** proteins.

By contrast, solid viruses have two types of protein, which we will call **Small** proteins and **Large** proteins. As their namesake suggests, these proteins are varying in size and we can count eight of each category per object, respectively.

How do you classify these viruses with the characteristics?

White Pixel Count

Based on the phenotypical differences we have just presented, the most straightforward way to classify the viruses is to check for each one whether the object body has holes or not.

In other words, given a black and white image check for the presence of white pixels inside each virus.

A. Image Preprocessing

We start of first by converting the original image to **grey scale** then ***threshold filtering*** it to get rid of the background.

At this point we have a binary image where black pixels correspond to virus pixels and white pixels to the rest.

We finally, inverse the image so that white pixels correspond to the virus and black pixels correspond to the rest.

B. Protein Filtering

At this point, our virus contains the entirety of the body — including the proteins — to make the task of classification easier, we probe the image with **Morphological Operations** in order to keep only the body of the image.

After several trial and errors, we settled on a **Manual Opening operation**(i.e erosion followed by dilation) with a Cross Structuring element of size (2,2) & (10,10) respectively.

This structure allows us to get rid of the proteins layer of the virus while maintaining the number of independents virus bodies and holes for viruses of the *Hole* class.

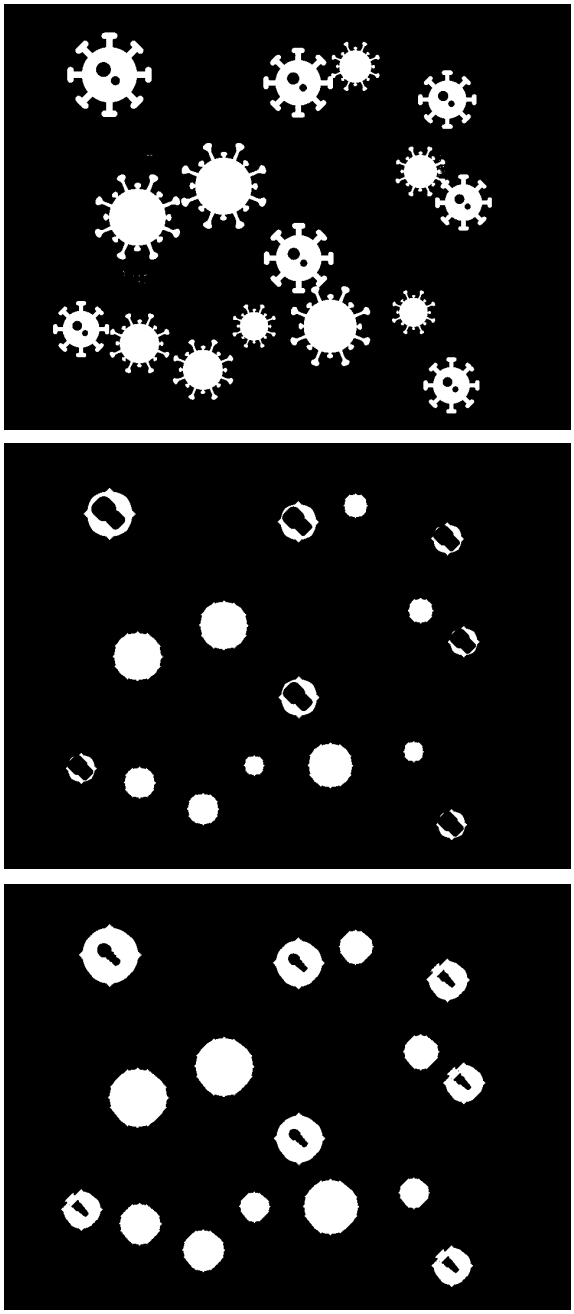

Fig 2. Protein Filtering
(from top to bottom);
Preprocessed, Erosion, Dilation

C. Pixels Counting

Finally, we take the white mask of each virus and compute **the difference between the mask and the virus pixels**. For viruses with no holes the white mask and the virus mask are equal hence the difference will be null (i.e no white pixels). For viruses with holes, the difference will contain the white pixels of the white mask. An example of this procedure can be seen in Figure 3

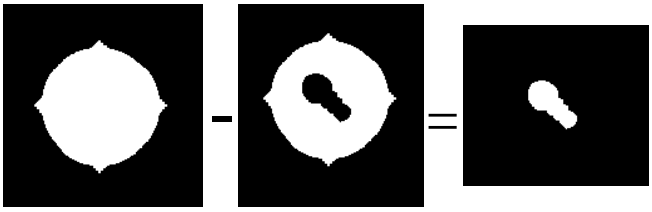

Fig. 3: Pixel Counting for Hole Virus

Have you used any strategies to promote the labelling accuracy?

If we pay close attention to the image at hand, we can see that several viruses are "touching" or within one another's regions.

In that respect our labelling task concerns colouring each virus with one colour. Hence, the problem then becomes: **how to accurately colour viruses with overlapping areas?**

To answer this question, we have used several strategies to account for the peculiarities of each type of virus, in order to obtain the most optimised labelling(i.e. colouring).

I. General Approach

In order to label viruses, we have created a *class dictionary* during the classification part.

Since the viruses are sorted by their bounding box's y-value, we may use the index of each virus to track them.

Hence, our dictionary takes for each virus, **its index as the key** and the **initial letter of the colour it corresponds to as the value** — based on the white pixel rule we defined earlier. The finalised dictionary can be found in Table 1 below.

Key	Value
0	B
1	G
2	B
3	B
4	G
5	G
6	B
7	G
8	B
9	G
10	G
11	B
12	G
13	G
14	G
15	B

Table 1. Classification dictionary based on y-intercept value (key) for colour label (value)

Given the dictionary, we then proceeded to create masks for each virus and using the intersection between the mask and the virus to classify it, as in Tutorial II.

II. Dynamic Mask Size

It is clear from the input image that the viruses have *varying sizes*, hence their masks must differ in area.

Naturally, we may increment the iterations number for each mask as in the tutorial. Nonetheless, this proved to be inconclusive as the mask fluctuated in size.

Instead to get more control over the size of the masks, we kept the dilation operations at one iteration and created a *dynamic_radius* variable, that assigns a size to the structuring element based on the area of the virus.

Hence for each region — corresponding to a virus— we defined the size of the structuring element — used for dilation in the mask creation procedure — as **the sum of three functions** tweaked to obtain the size for the image at hand

III. Dynamic Mask Shape

As we mentioned earlier, the mask creation part proved to be more demanding than in the tutorial based on the fact that we have **more than one class of touching objects**.

If we pay close attention, we can observe the *Solid* viruses have **V-shaped** proteins, therefore the most adequate structuring element (SE) proved to be the **Diamond shape**.

By contrast, *Hole* viruses have a more symmetrical distribution of proteins, creating more of an **Arc-shaped** antenna. In that respect, we used **Disk** SE to account for their shape.

In accommodating for each class of viruses, we managed to create **hybrid masks**, that cover all the regions of viruses while minimising the intersection between touching viruses, such that colour overlap is pruned during labelling.

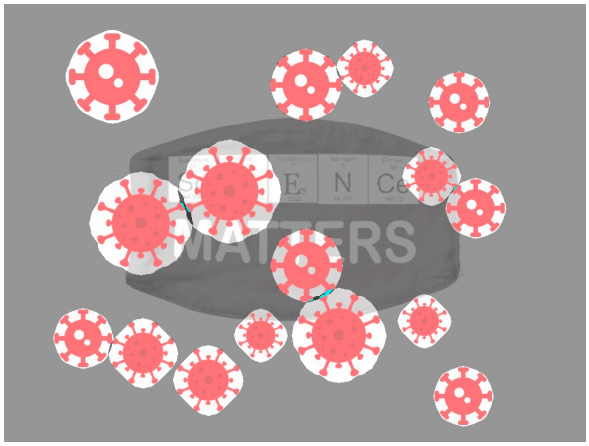

Fig 3. Dynamic Masks (Shape & Size)

III. Edge Case

Thought small, the pair of touching virus at the centre of the image, still seemed overlapping — as can be seen from Figure 3 — causing inevitable colouring of one another. The reason behind this lies in the fact that one protein of the *Solid* virus is actually **inside the area** of the *Hole* virus.

To address this, we have first modified the shape of the mask of the solid virus to **Star**, which added a layer of curve that went below the proteins of the *Hole* virus mask, hence reducing the overlap and showing decent results.

in order to perfect that, we eventually made use of a more advanced segmentation technique, namely **watershed algorithm**.

Without getting into the details, this technique essentially relies on the principle of topography, whereby images are interpreted as intensity maps (i.e. 0 pixel =is a valley, 255 pixel is a peak) which are then based on changes in elevation of 0 pixels(i.e. valley)

Conclusion

Finally, we coloured the pixels' original image based on the value of the classification dictionary obtaining the final result as in Figure 4.

In all, the task of colouring was the most challenging, due to the **structure of the viruses** and their **geospatial proximity**. We managed to achieve better performance through **dynamic masking** to accommodate for each virus, which we pushed even further through segmentation for even more visually similar results to the the final output for problematic edge cases.

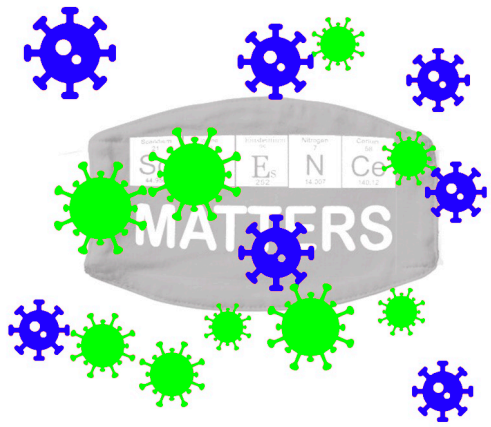

Fig 4. Final Output
(With edge case watershed)