collage generator

December 2020

1 Introduction

1.1 Breif Description

This report demonstrates the process of collage generator.

1.2 Parameters Table

We list important parameters for generating a collage and further illustration will be provided in following sections.

Parameter Name	Default	Description
canvas_size	[3000, 3000]	The collage size
cluster_size	[2200, 2200]	The glomerulus-proximal tubules cluster size
example_image	/	The example image for background
gaussian_noise_constant	5	σ for the gaussian distribution
patience	100	the retry time for each insert if overlap
item_num	5	total number of glomerulus + artery + arterioles
ratio_dict	{"cluster":0.2, "artery": 0.5, "arteriole": 0.3}	The ratio of each case Denote the 3 parameters as: {"cluster": p_1, "artery": p_2,"arteriole": p_3}

Table 1: Important Parameters

2 Generative Process

We summarize the generative process in this section, and then demostrate related sub-steps in section 3.
Before generation, we need to **load all the images of components**. Meanwhile, we record the maximum size of those components on both x and y axis, denoted as max_component_size ($\in R^{1\times 2}$). Then the generation for each canvas is by following sequences:

- **Step 1: Preparations.** We first enlarge the canvas size by:

$$\text{canvas\_size\_enlarged} = \text{canvas\_size} + 2 \times \text{max\_component\_size}$$

  Denote the enlarged canvas as the main canvas.

- **Step 2: Generating and Inserting Glomerulus-proximal Tubules Cluster**
  We calculate the number of clusters by: n_cluster = $\lceil p_1 * \text{item\_num} \rceil$. Then for every cluster, we do:
  - Construct a temporary cluster canvas with size of cluster_size.
  - Randomly pick an item from the glomerulus class without replacement (once the image in the list is exhausted, the selection will be re-filled), augment the glomerulus item, and append the item to the cluster canvas.

- Randomly select pre-defined numbers (300 by default) of proximal tubules without replacement, apply augmentation and try to insert it into the canvas without overlaping with other items.
- Try to randomlly insert the cluster canvas to the main canvas.

- **Step 3: Generating and Inserting Artery and Arteriole**

  - we calculate the total number of arteries and arterioles, as well as the proportion by:

$$\begin{aligned} \text{n\_artery\_and\_arterioles} &= \text{n\_item} - \text{n\_cluster} \\ p_2' &= p_2/(p_2+p_3) \\ p_3' &= p_3/(p_2+p_3) \end{aligned}$$

  - Then we randomize the number of arteries and number arterioles by sampling from: (n_artery, n_artery) $\sim$ Multinomial(n_artery_and_arterioles, p_2', p_3').
  - For every item, we randomly pick it from corresponding class without replacement, apply augmentation and randomly insert it into the main canvas.

- **Step 4: Generating and Insering Distal Tubules**
  By default, we set the number of distal tubules in every canvas as 3000. For every item, we randomly pick it without replacement, apply augmentation and randomly insert it into the main canvas.

- **Step 5: Postprocessing** After inserting all components, we postprocess the main canvas by following steps:

  - Cut the enlarged main canvas by cropping the center part.
  - Generate the background by example_image
  - Append the background to the main canvas where there is empty.
  - Add random gaussian noise generated from $N(0, \sigma^2)$ on every pixel.

3 Sub-steps in Generative Process

To avoid redundency in the generative process description, we illustrate the details of related sub-steps in this section.

3.1 Augmentation

We applied augmentation for every items we added to the collage. For instances other than distal tubules, we applied flip, grid distortion, transpose, translate, scale and rotation with randomness. If the input items are distal tubules, the augmentation method is almost the same except that we provide a consistent direction augmentation. That is to say, when applying random rotation, we constrain the rotation degree between 0 to 15 (-90 degree to 90 degree for other items). Thus, the distal tubules' directions won't be diverged a lot from the original one.

3.2 Insertion

Every time we generated an augmented item, we need to insert it to the main canvas. Based on different situations and items, the insertion strategy are different.

3.2.1 Initial Insertion

The "Initial Insertion" is only applied to the first Glomerulus-proximal tubules cluster at Step 2, because there is no other items on the canvas. Hence this process just randomly pick a place on the canvas while keeping all of the cluster inside of the canvas, and then insert the cluster into the main canvas.

3.2.2 Secondary Insertion

The "Secondary Insertion" is only applied to the second and following Glomerus-proximal tubules cluster at Step 2 and the artery and arteriole at Step 3. In this process, we first randomly pick a place on the canvas while keeping all of the cluster inside of the canvas. Then we check whether there is an overlapping with existing items on the canvas, if so, we'll randomize the location and check again. If not, we'll insert the item to the canvas. For every item, we'll forgive appending that item to the canvas if we failed to find a legal position within "patience" number (100 by default) of times.

3.2.3 Try Insertion

The "Try Insertion" is applied to every distal tubule. In order to control the density of images by a numerical value called *step_length*, we apply an escape-overlapping algorithm. If the initial point is overlapping, try to "escape" the overlapping and append at the first position successfuly escape. If the initial point is not overlapping, try to find a overlapping point and append at the last non-overlapping point before this one. Similar to the "Secondary Insertion", for every item we'll forgive appending that item to the canvas if we failed to find a legal position within "patience" number (100 by default) of times. In detail, the algorithm is demostrated below.

Algorithm 1: vignette_insertion(vignette, patience, step_length)

```
Start with a random position on the canvas;
if Inserting the vignette here causes an overlap then
    for patience times do
        move the insertion point by step_length in both x and y, until we find a position
         that will no longer overlap any inserted vignettes;
    end
    Use this position as the add_point;
else
    for patience times do
        move the insertion point by step_length in both x and y, until we find a position
         that has an overlap with any inserted vignettes;
    end
    Use the last position before overlap as the add_point;
end
Insert Vignette at add_point;
```

3.3 Background Generation

Based on pre-processed background material, we randomly generating a new background by following steps:

- Scan the background image by a sliding window with certain stride. Meanwhile, every time the sliding window moves, we add random noises to those 4 corners of the window to allow us cut more stochastic background materials.
- For every cutted material, we ensure most of the places are nonempty by a threshold. Append all legal materials to a list.

- Construct an empty canvas. Scan the canvas by a sliding window with certain stride. Similarly, we'll add random noises to thoes 4 corners of window, and then we randomly pick a background material from the list, apply augmentations, and append to the canvas.

4 Example

We present the generative process by an example with default parameters.