Artifact Evaluation: Euro-PAR 2024 Paper-ID 136
PriCE: Privacy-Preserving and Cost-Effective Scheduling for Parallelizing the Large Medical Image Processing Workflow over Hybrid Clouds

Yuandou Wang, Neel Kanwal

17 May, 2024

The entire artifact has been stored in the following Google Drive link: https://drive.google.com/drive/folders/1U7p8NOx5Z50mYx0wTt588BHdPad4-fft?usp=sharing. This folder contains:

- Euro_PAR2024_Artifact_paperID_136_PriCE.zip[1]
- Euro_PAR2024_Artifact_paperID_136_PriCE.md5[2]

— Zipped Artifact file name: Euro_PAR2024_Artifact_paperID_136_PriCE.zip
— MD5: [391914d9020aceaacba74ef10b5e9ba9]

It is worthy noting that the results shown here are in whole or part based upon data generated by the TCGA Research Network: 'https://www.cancer.gov/tcga'[3]. The original Whole Slide Images (WSIs) are protected from the TCGA copyrights, more details please check the website[4]. Therefore, the authors are not be able to share the original WSI but instead share patches or give links to the original TCGA repository with the names of the files. In our case, we used 'TCGA-E9-A1N3-01Z-00-DX1' to present the research.

Question 1: Where to download a large medical image data?

The original WSI is available on the TCGA repository[5] Download the Whole Slide Image (WSI) named 'TCGA-E9-A1N3-01Z-00-DX1' from the TCGA research network and place it at the unzipped folder "/PriCE/dataset/1WSI/data/"

Question 2: How to setup experimental environment

Use the terminal for the following steps:

Part (a): Create the environment from the environment.yml file

```
conda env create -f environment.yml
```

Part (b): Activate the new environment

```
conda activate test-env
```

Part (c): Verify that the new environment was installed correctly

```
conda env list
```

[1] https://drive.google.com/file/d/1RpSf52Jxk5L4YYfB3cLemp1nLJHmPJG9/view?usp=sharing

[2] https://drive.google.com/file/d/1OEvDEpsCz0AxqWpdqw1s6Tl8U_2ETDVF/view?usp=drive_link

[3] https://www.cancer.gov/ccg/research/genome-sequencing/tcga/using-tcga-data/citing

[4] Using TCGA data. https://www.cancer.gov/ccg/research/genome-sequencing/tcga/using-tcga-data

[5] TCGA SLIDE IMAGE VIEWER for the WSI named TCGA-E9-A1N3-01Z-00-DX1. https://portal.gdc.cancer.gov/image-viewer/MultipleImageViewerPage?caseId=03c143e0-d8a1-4d60-a4a3-df0501fc6b6e

Question 3: Folder explanations?

There are the following four main folders in the `.ZIP` file.

- dataset: storing the datasets, e.g., the cropped patches and related intermediate data files, etc.
- inference: storing the CNN inference models.
- pipeline-example for artifact detection: storing the application using CNN inference models for artifact detection in a WSI, that can be ran on a single GPU server.
- PriCE-exps: storing the experimental workflows and /or Jupyter notebooks of the PriCE experiments and simulations for visualization and demonstration.

Question 4: How it works?

To cope with the diverse image samples of the privacy-preserving data-splitting procedure, we abstract the entire image as a grid graph where different patches with pixel size $p \times p$ are cropped from the original image $\mathcal{D}$, containing sensitive image labels and objects. The image label contains sensitive coordinate information to reconstruct the image and guide the outcome.

Let $G = (V, E)$ be a graph extracted from the entire patch dataset D cropped from the original image $\mathcal{D}$. Each patch is represented as a vertex $\upsilon \in V$. Two vertices υ and μ of V such that $(\upsilon, \mu) \in E$ are called to be adjacent. Let $\upsilon = (x_i, y_i)$ and $\mu = (x_{i+1}, y_{i+1})$, we denote all possible adjacent relationships between υ and μ as: (1) horizontal: $|x_{i+1} - x_i| = p$; (2) vertical: $|y_{i+1} - y_i| = p$; and (3) diagonal: $\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} = \sqrt{2} \times p$. With these characteristics, the positions of the patches can be identified in the original image (See in Figure 1).

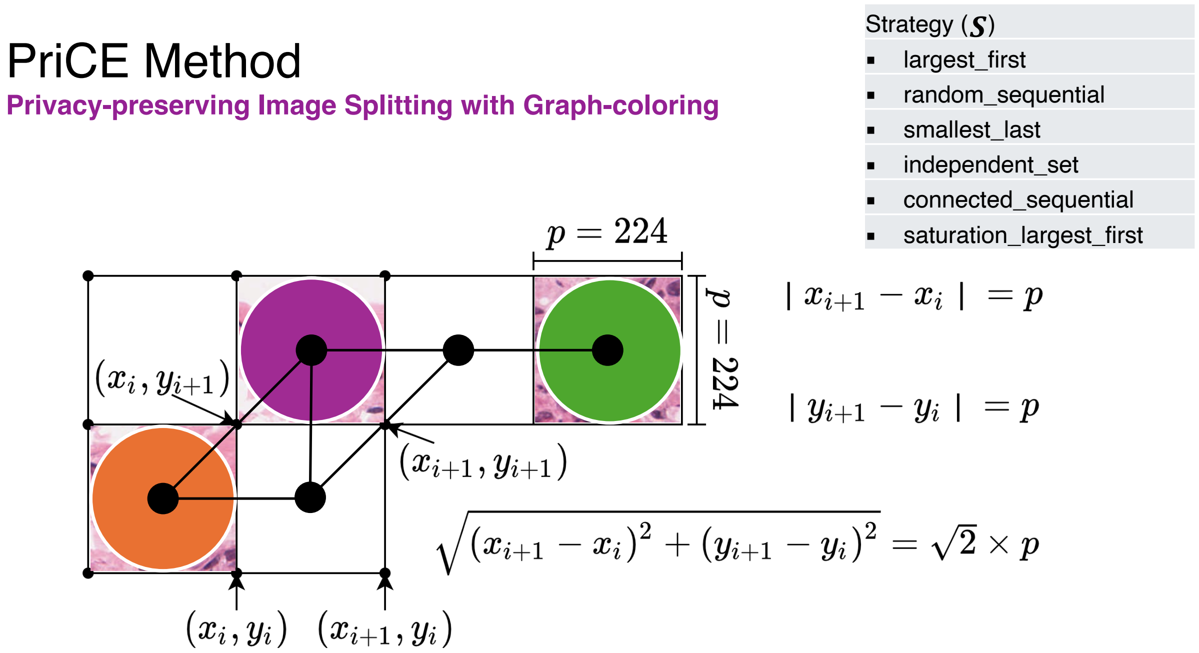
Based on the assumption, if more adjacent patches are placed in the same sub-dataset, the higher the probability that the adversary will restore the entire image. We study different split strategies to scramble these identifications and reduce the risk of restoring the original dataset from the image fragments by the adversary. On the one hand, we adopt the graph-coloring-based split strategies, including 'largest_first', 'random_sequential', 'smallest_last,' 'independent_set,' 'connected_sequential', 'saturation_largest_first', to split the entire dataset D into different sub-datasets $d_{p,1}, ..., d_{p,N}$, such that no two adjacent vertices share the same color or dataset. On the other hand, we introduce a random data perturbation to preserve the sensitive coordinates on split datasets' labels by inserting random noise.

Part (a): how to split a gigapixel medical image?

We extract $(x, y)_{\text{coord}}$ as a data matrix A_p of size ($a \times b$), $a < b$, from $d_{p,k} \subset D$. After normalization, we compute the covariance matrix of the normalized matrix $A_{x,c}$, and then computed the eigenvalues λ and eigenvectors $\vec{V}$ so that we can get the top-k eigenvectors $\vec{V_k}$ to calculate A_e. Moreover, we transform the data into a new coordinate system and encrypt it into datasets $\{d_{e,1}, d_{e,2}, ..., d_{e,N}\}$. From the perturbed data, since we know the noise variance, we obtain the estimate coordinates $\hat{Y}$ from decryption by inversely transforming the eigenvector matrix $\vec{V}$ and A_e. Besides, since we know the mappings of original labels and their corresponding encrypted labels, it is easy to measure the output utility (Please see the visualized results, e.g., in Figure 2).
1. Check the code cells and markdowns in the Jupyter Notebook named `PriCE/PriCE-exps/graph_coloring_based_image_splitting.ipynb`
2. Check the code cells and markdowns in the Jupyter Notebook named `PriCE/PriCE-exps/evenly_split_w_wo_shuffle.ipynb`

Part (b): how to encrypt/decrypt sensitive information of medical images? How to quantify the privacy-preserving goals?

Check the code cells and markdowns in the Jupyter Notebook named `PriCE/PriCE-exps/pertubedata_privacy_risk_evaluation.ipynb` (data perturbation and its privacy-preserving algorithm evaluation)



PriCE Method

Privacy-preserving Image Splitting with Graph-coloring

Strategy (S)

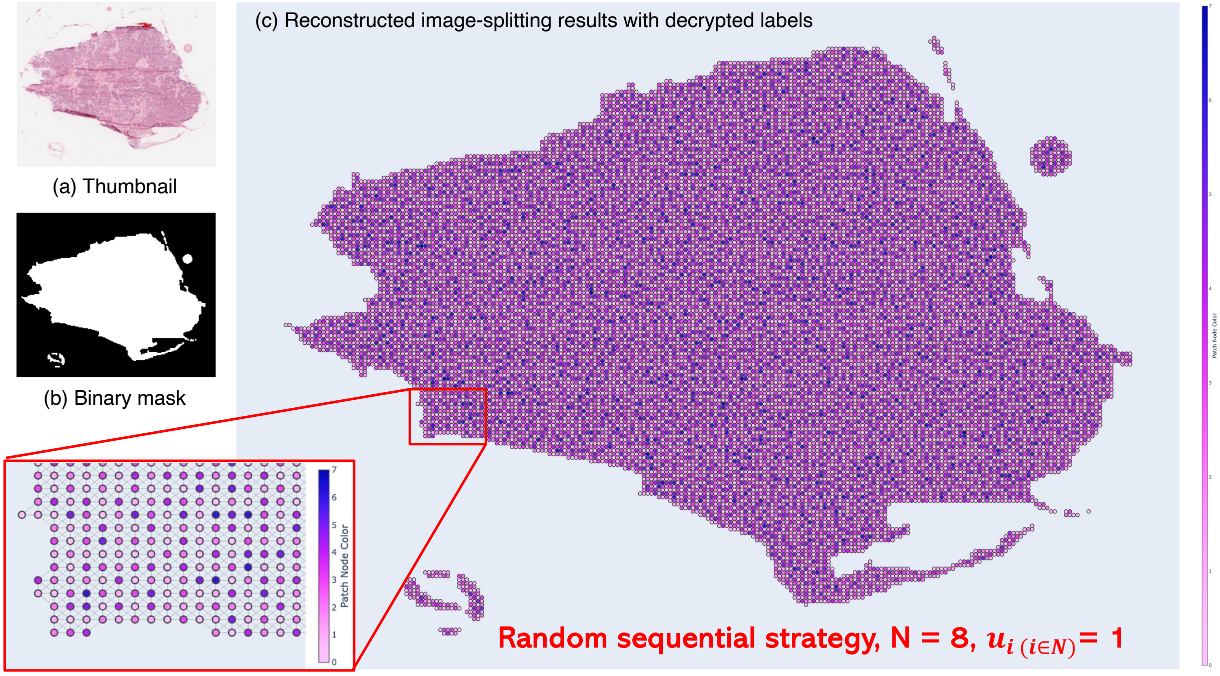
- largest_first
- random_sequential
- smallest_last
- independent_set
- connected_sequential
- saturation_largest_first

$$| x_{i+1} - x_i | = p$$

$$| y_{i+1} - y_i | = p$$

$$\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} = \sqrt{2} \times p$$

Figure 1: Abstraction of the graph-coloring-based-image-splitting: (a) crop the patches from the original image, e.g., a WSI, (b) identify the position $(x, y)_{\text{coord}}$ of each cropped patch, and (c) identify the adjacent edges.



(a) Thumbnail
(b) Binary mask
(c) Reconstructed image-splitting results with decrypted labels
Random sequential strategy, N = 8, $u_{i\,(i \in N)}$ = 1
Patch Node Color
0
1
2
3
4
5
6
7

Figure 2: Visualization of the image-splitting: (a) the thumbnail picture of the original medical image, (b) the binary mask picture, and (c) the reconstructed graph from estimated coordinates after decryption.

Part (c): how to seek the 3D Pareto optimal resource planning?

Check the code cells and markdowns in the Jupyter Notebook named `PriCE/PriCE-exps/Pareto_3D_evaluation.ipynb`