Auto-annotation of Pathology Images

Chao Huang (ch3474), Yihao Li (yl4326), Yufeng Ma (ym2764)
Shuo Yang (sy2886), Xiaoyun Zhu (xz2884)

Supervised by Dr. Adler Perotte and Dr. Shana Coley
Instructed by Dr. Adam Kelleher

Columbia University
Data Science Institute

Agenda

- Introduction
- Data and Preprocessing
- Models
- Productionization
- Summary

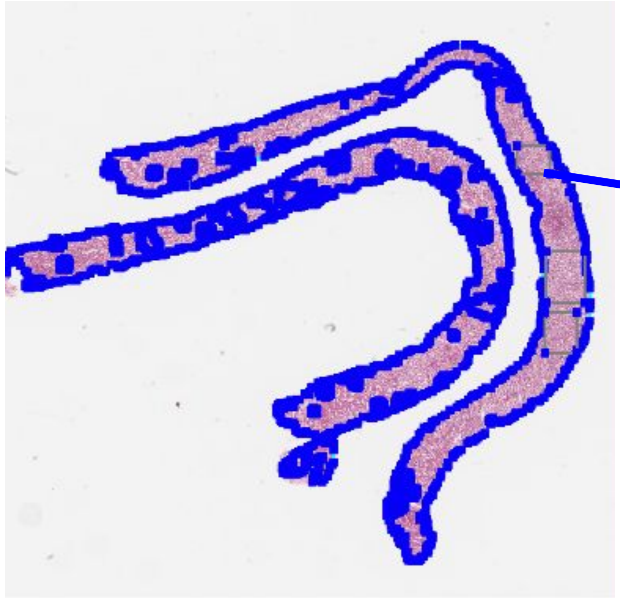
Introduction

Introduction: Background

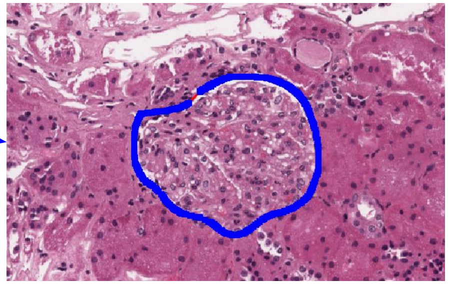
Region of Interest: **small** compared to WSI (Whole Slide Image)

⬇

Time consuming, labor intensive, inconsistent



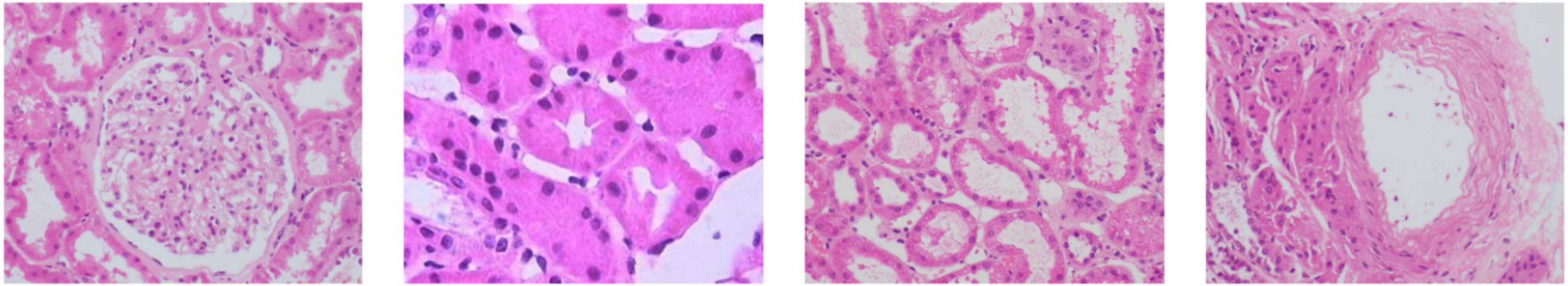
WSI



Region of Interest

Introduction: Goal and Milestone

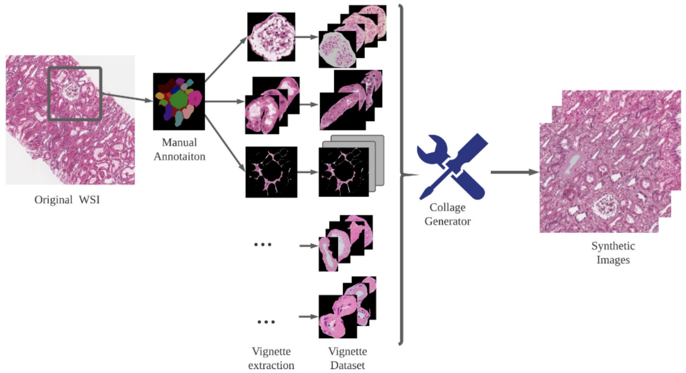
Segmentation of four histologic features



(a) Glomeruli (b) Interstitium (c) Tubules (d) Vessels

Complete Workflow: Human-in-the-loop

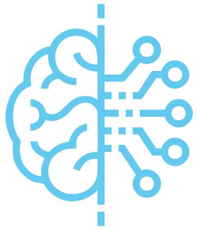
labeled Data from Professionals



Synthesized Data

Data Preparation(Data For Iteration 0)

Models



DL networks
e.g., U-Net
Mask R CNN

Web Application

Predict and Display Masked Slides

Corrected Masks for fine tuning model

Annotation Correction

Interactive Web UI Core(Iteration x)

Data and Preprocessing

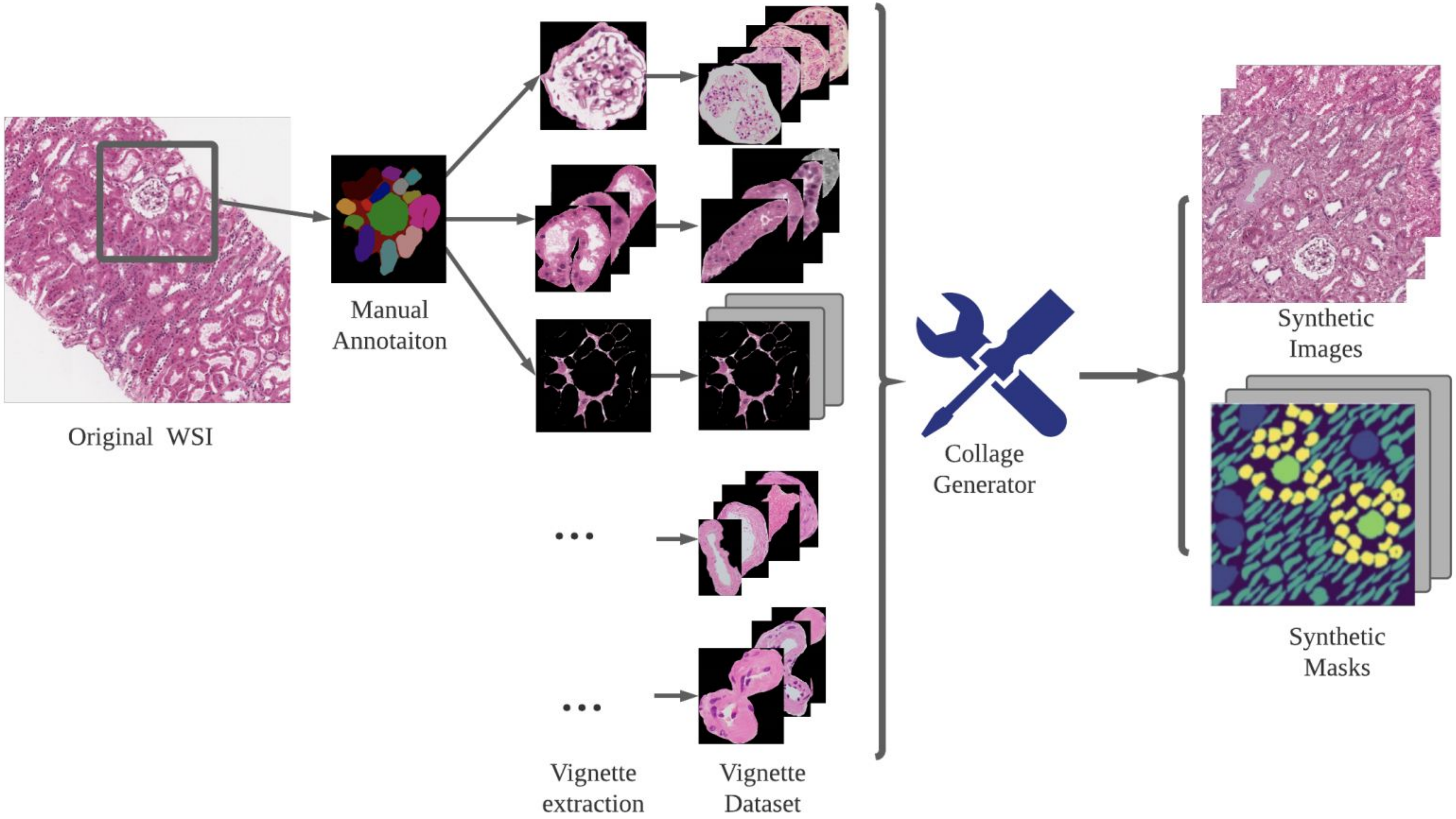
Data and Preprocessing: Datasets

- CUMC Dataset:
    - Target: Annotated WSI (Whole Slide Image) data from pathology residents in our project.

- Auxiliary Datasets:
    - MultiStain: Collected from posted researches by Jayapandian et al.[1]
    - Collage: Generated image from vignettes under the same magnifications

Data and Preprocessing: Collage Generator

- Output:
  - Images that mimic the real ones
- Input:
  - Vignettes: Image of one specific item
  - Parameters: e.g., image size, number of each kind of item



Collage Generator

Step 1:

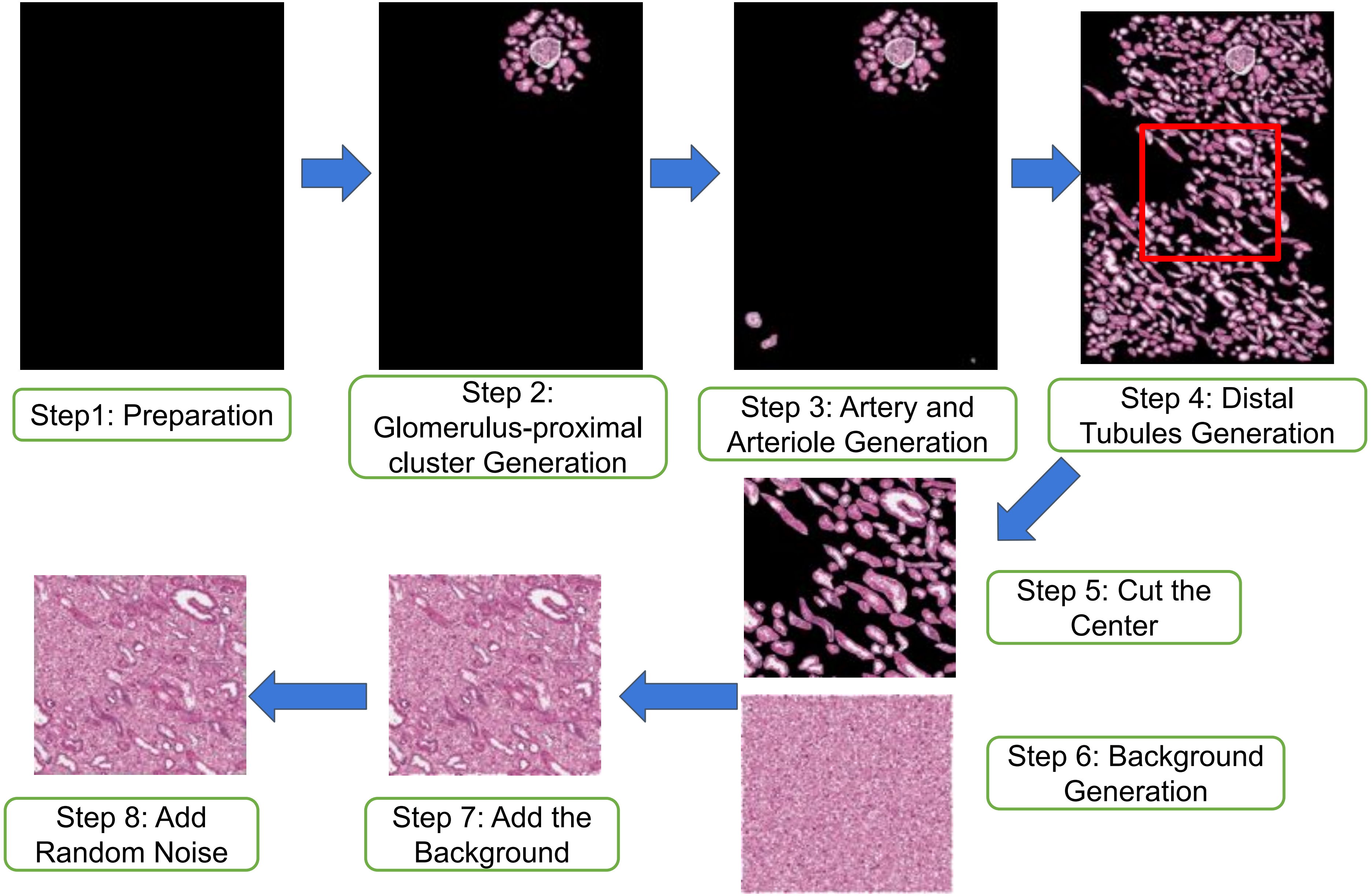
- Prepare items images
- Enlarge the canvas

Step 2-4:

- Generate
    - Glomeruli
    - Artery and Arteriole
    - Distal Tubules

Step 5-8:

- Postprocess

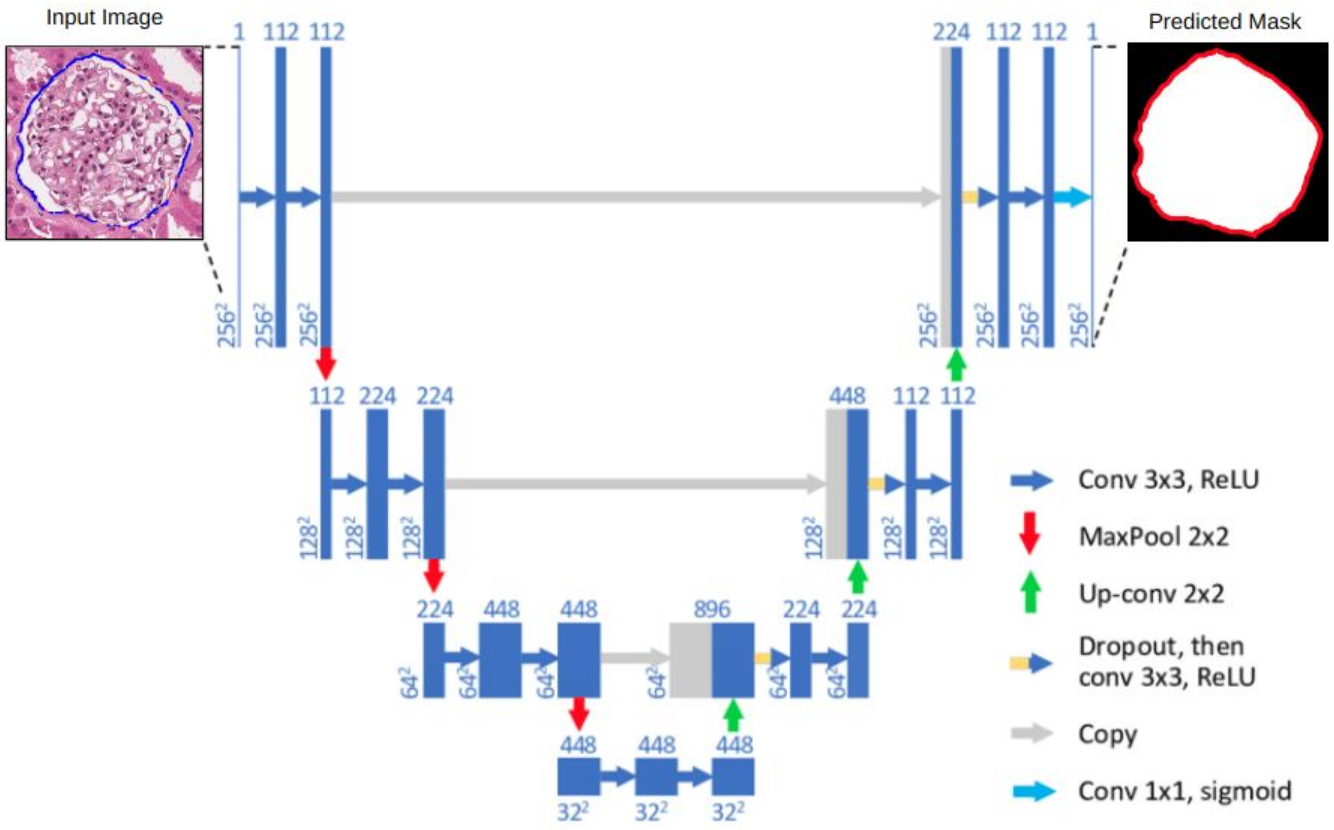


Models

State-of-the-art, and further

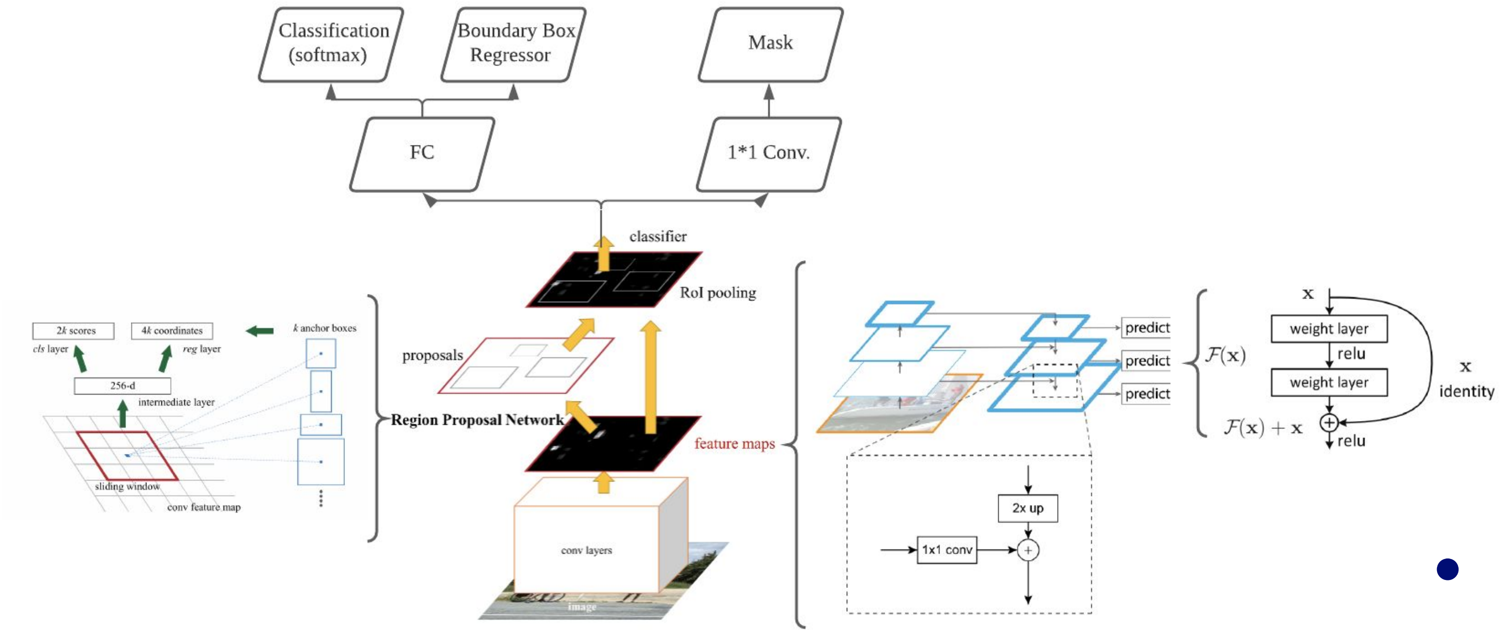
- Field Frontier: Kidney histopathology
  - U-Net: popular model in papers about this topic from late 2019 to Nov. 2020 [1,2,3]
- Field Frontier: Image Segmentation
  - R-50-FPN Mask-RCNN = Faster R-CNN + FPN + ResNet50 (Detectron2, FAIR 2019[4])

U-Net: The success of Encoder-Decoder



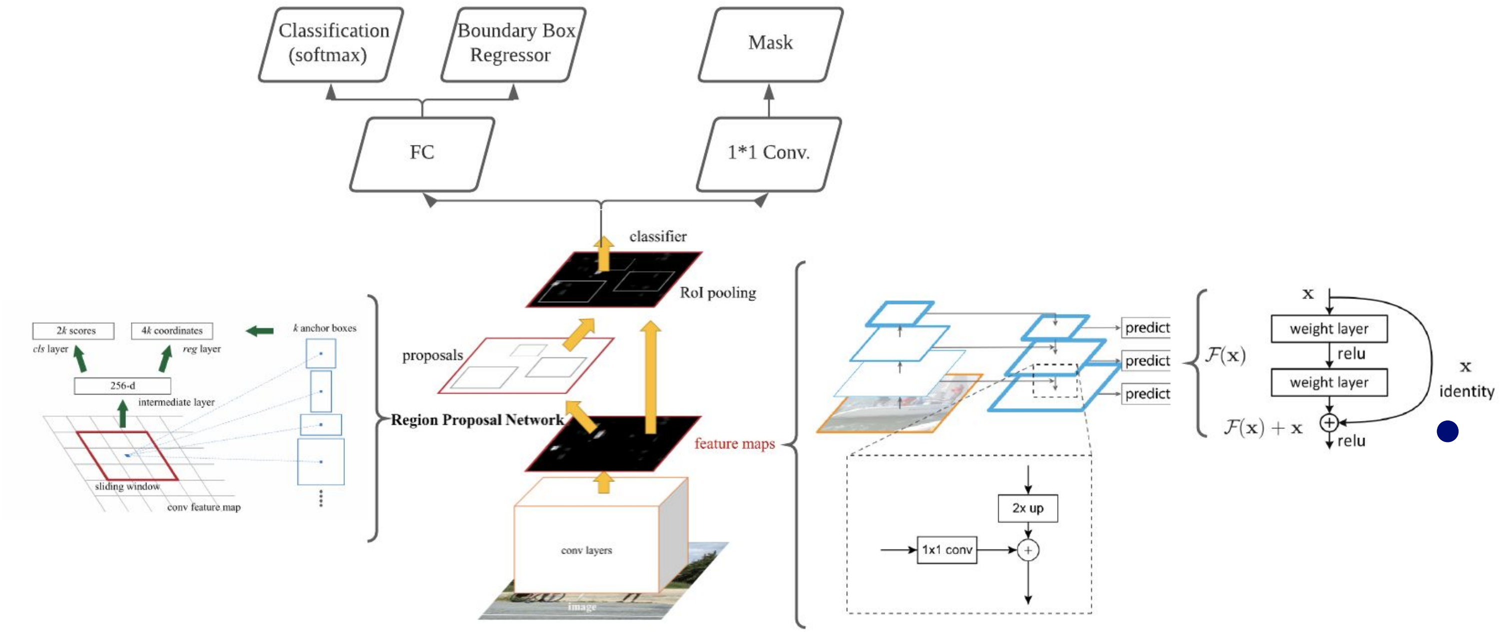
- Encoder-Decoder
  - Using same-size convolution to have a better control on size
- Loss function
  - Using an “one vs. rest” binary class strategy, penalize on element-wise cross entropy
  - Following the practice in [1], give an additional 20% penalty on the edge(red line)

Mask R-CNN: Successful comb. of successes



- Core components
  - ResNet: well-known Vision Model
  - FPN: Feature Pyramid Networks
  - Faster R-CNN: Object Detection
  - Mask generated from 1*1 conv. layer from the output
- Model Config. & Management
  - Detectron2 by FAIR

Mask R-CNN: Successful comb. of successes



- Input Format
  - COCO(Common object in Context[5])-like json-based format
  - Initial model completely trained on the synthesized collages
- Configuration:
  - Detection mode rather than Panoptic mode.
  - Using ResNet-50
  - Upsampling the minority class.

Result: Quantitative Evaluation

- Evaluation Metrics
  - Using pixel-wise F1 score (Dice Coefficient) as the criterion with the supplementary of average precision and recall
  - With iterative pipeline, the evaluation for the initial model does not count 100% for the models

	U-Net @ MultiStain			**U-Net @ Collage**			**Mask R-CNN @ Collage**		
Class	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Glomeruli	0.8295	0.7378	0.9883	0.7750	0.6770	0.9343	0.5569	0.4167	0.9724
Vessels	0.5757	0.4471	0.8842	0.3842	0.3622	0.4964	0.1760	0.1800	0.1750
Tubules	0.9093	0.9476	0.8745	0.7727	0.9441	0.6541	0.5138	0.9961	0.3511

Result: Quantitative Evaluation

- Result:
  - The model trained from real-scenario data (MultiStain) performed a very impressive result.
  - Even for models trained from purely synthetic images, the transfer learning result is still acceptable for Glomeruli and Tubules, which enables the a pre-trained Mask R-CNN's deployment onto our iterative pipeline.

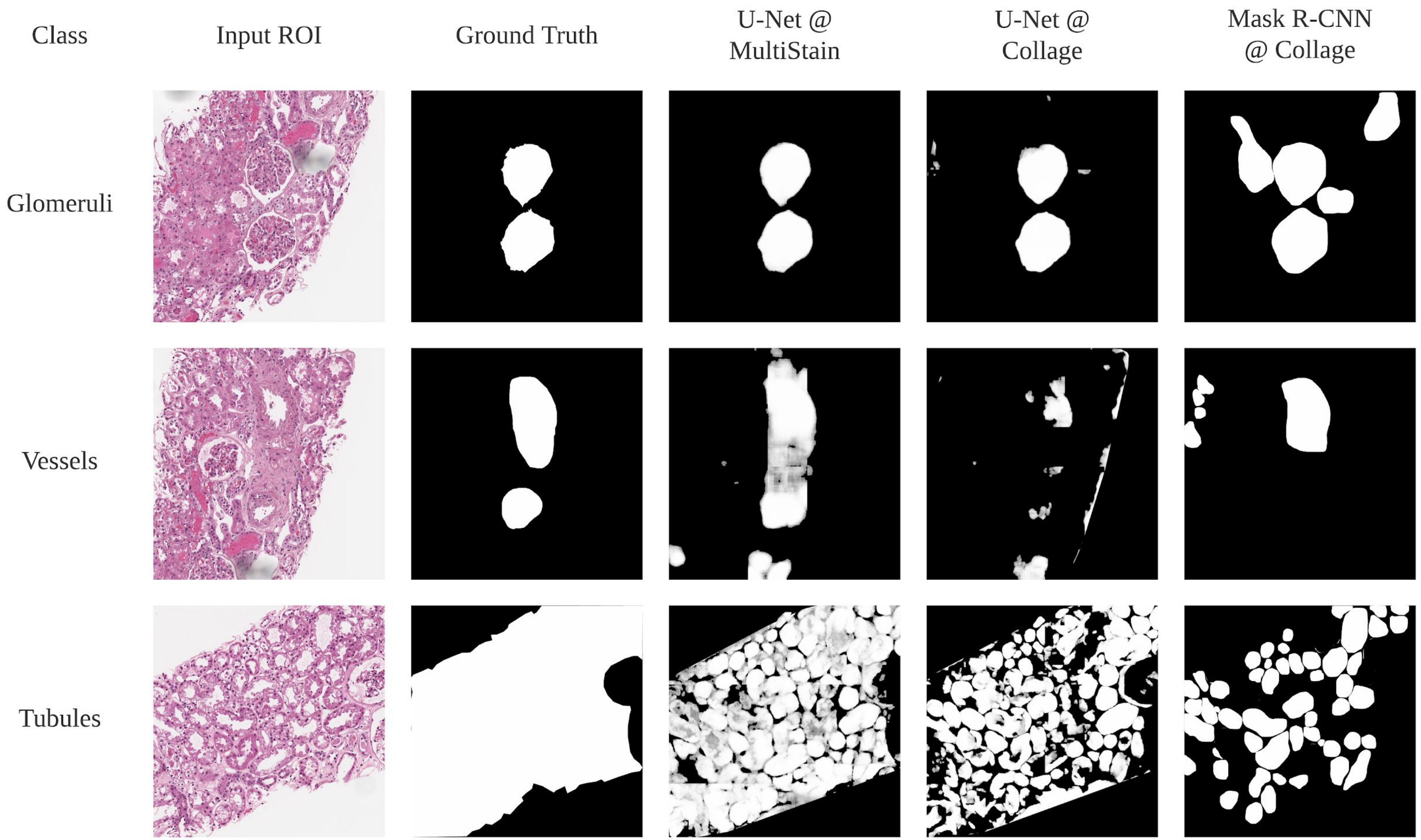
	U-Net @ MultiStain			**U-Net @ Collage**			**Mask R-CNN @ Collage**		
Class	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Glomeruli	**0.8295**	0.7378	0.9883	**0.7750**	0.6770	0.9343	**0.5569**	0.4167	0.9724
Vessels	**0.5757**	0.4471	0.8842	0.3842	0.3622	0.4964	0.1760	0.1800	0.1750
Tubules	**0.9093**	0.9476	0.8745	**0.7727**	0.9441	0.6541	**0.5138**	0.9961	0.3511

Result: Quantitative Evaluation

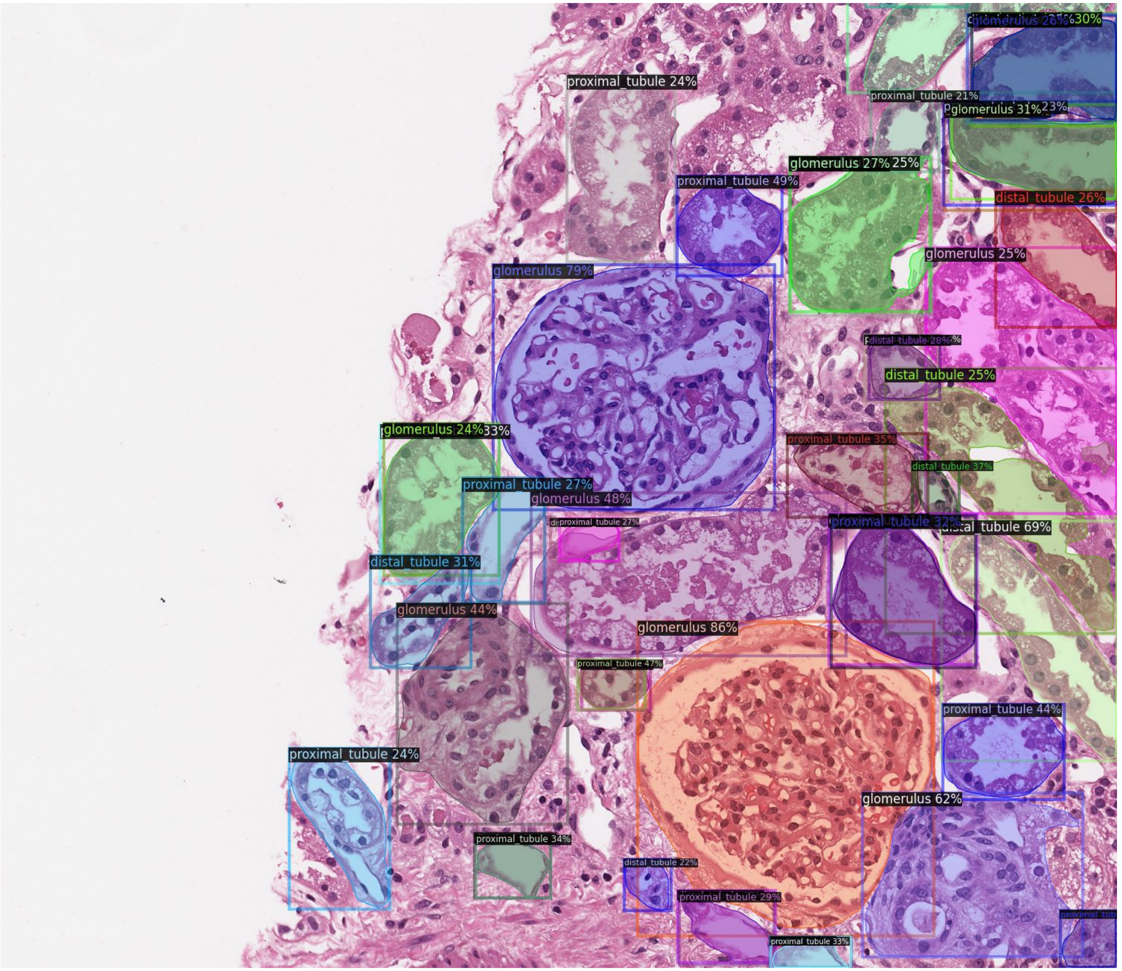
- Can be further improved
  - Vessel is the most difficult structure to be recognized, because of its dynamic shape.
  - The classifier inside Mask R-CNN is suffering from the imbalanced dataset and lack of data. This will be solved in the 1st iteration (we are currently at iteration 0 initializing the loop)

	U-Net @ MultiStain			**U-Net @ Collage**			**Mask R-CNN @ Collage**		
Class	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Glomeruli	0.8295	0.7378	0.9883	0.7750	0.6770	0.9343	**0.5569**	0.4167	0.9724
Vessels	**0.5757**	**0.4471**	**0.8842**	**0.3842**	**0.3622**	**0.4964**	**0.1760**	**0.1800**	**0.1750**
Tubules	0.9093	0.9476	0.8745	0.7727	0.9441	0.6541	**0.5138**	0.9961	0.3511

Result: Visualization



Result: Visualization

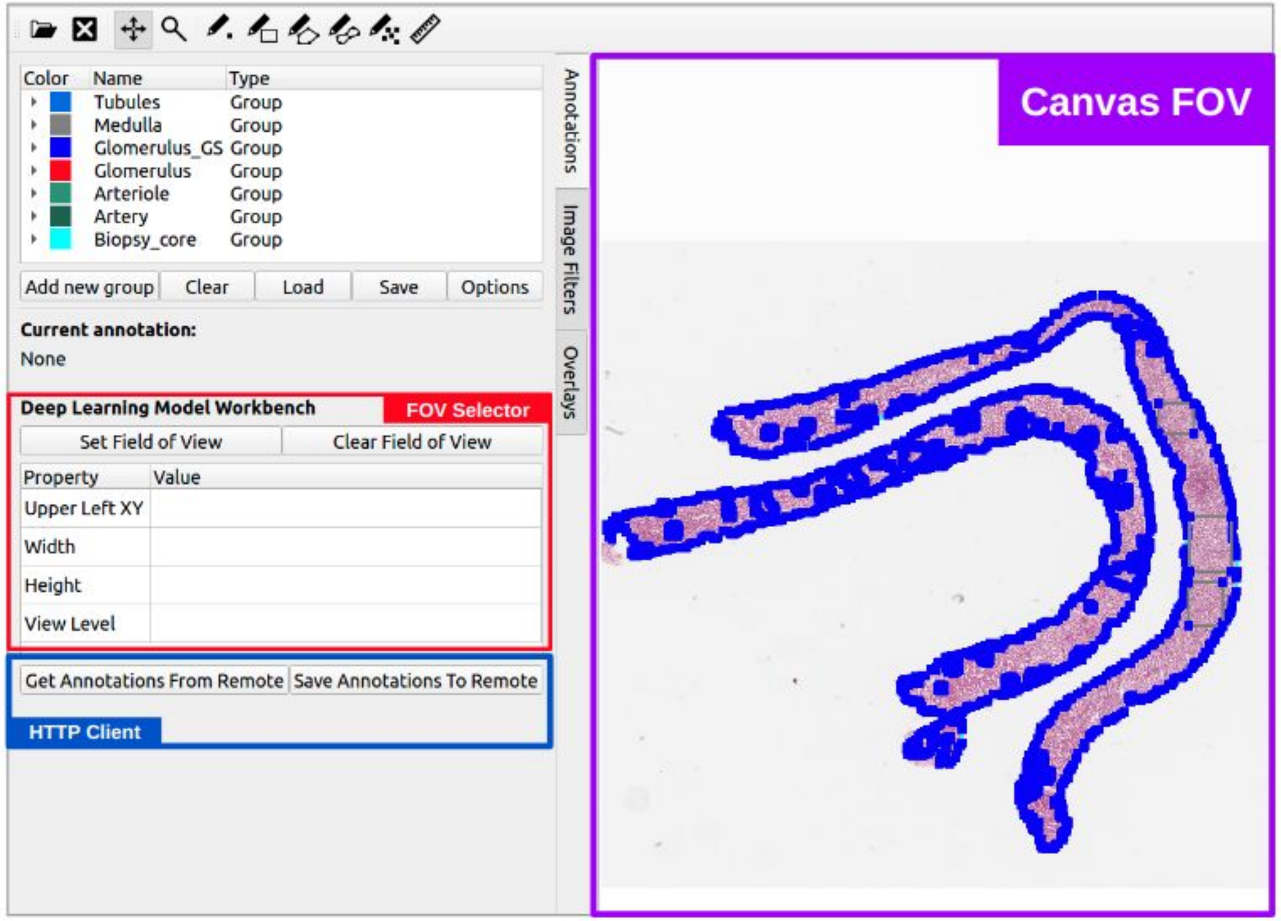


Productionization

Design of the Human-in-the-Loop Platform

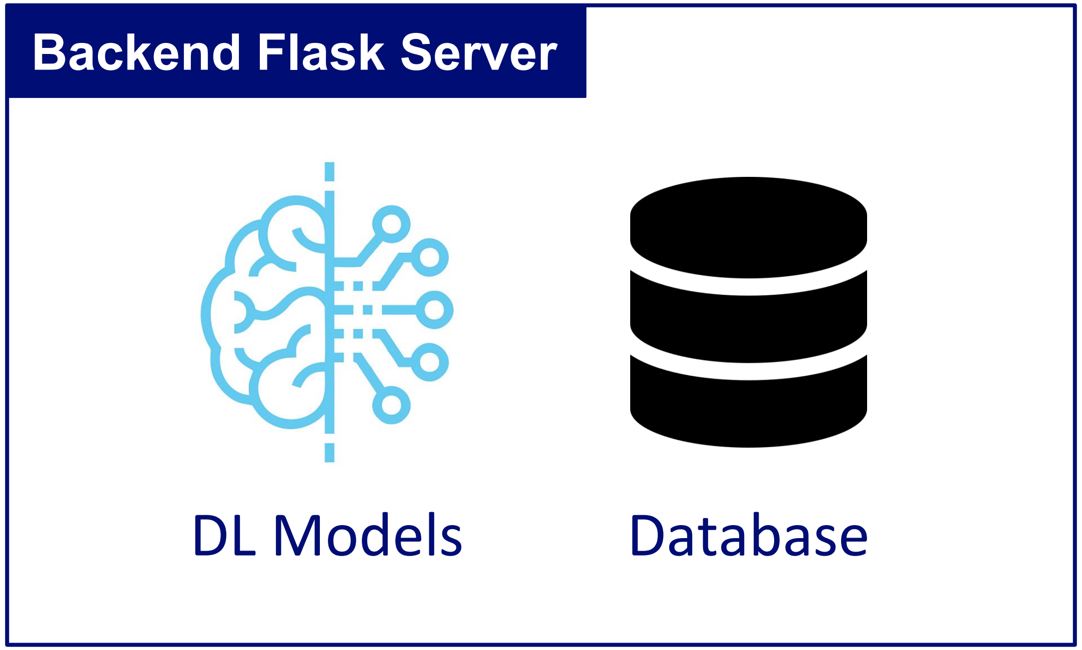
- Functionalities
  - Render predicted annotations from DL models on whole slide images
  - Correct annotations using interactive tools and update models with calibrated annotations
- Architecture
  - Frontend: a desktop Qt application based on ASAP[6]
  - Backend: a Flask server with a MySQL database and pretrained PyTorch models
- Deployment
  - Hosted on an AWS EC2 Linux machine
  - The client can be accessed through X11 forwarding

Human-in-the-Loop: Frontend Design



- Extensions added to ASAP
  - Field-Of-View (FOV) Selector
  - HTTP Client
- Functionalities
  - Select an FOV as the input region for segmentation
  - Transfer annotations with the back end using HTTP requests

Human-in-the-Loop: Backend Design

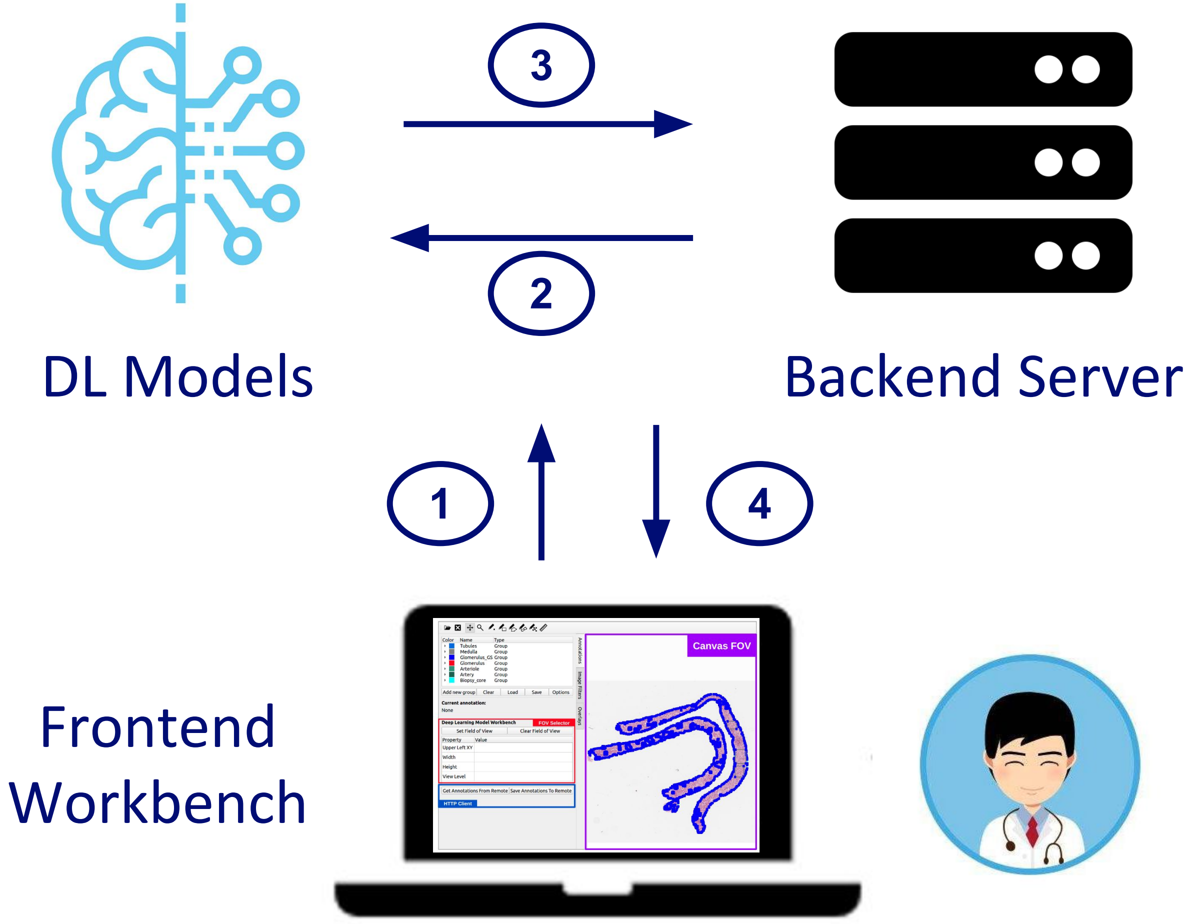



```
API Endpoint:
GET/POST http://127.0.0.1:5000/annotation/<slide_id>

Request Parameters:
<API_Endpoint>?x=<x>&y=<y>&width=<width>&height=<height>
```

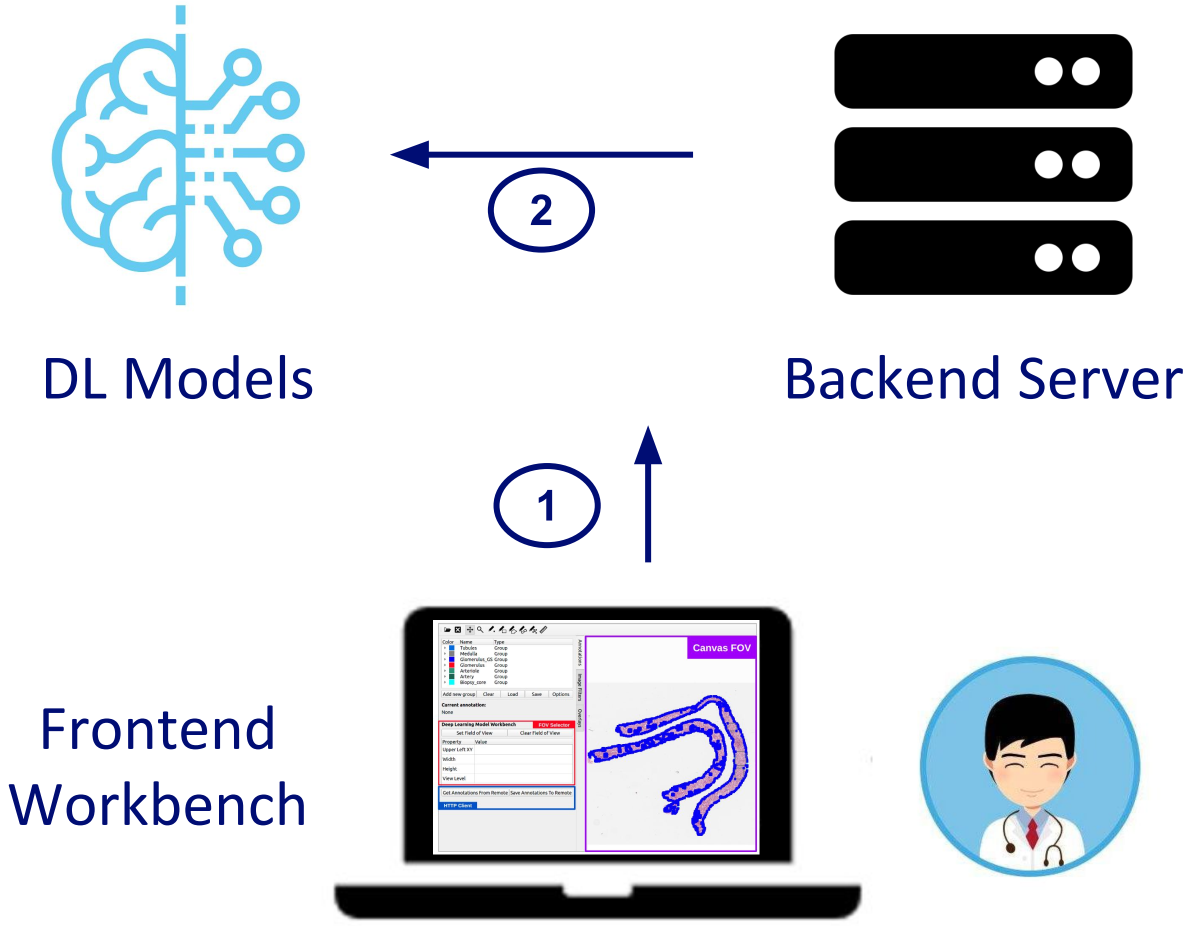
- Core components
  - A database saving annotations
  - An interface calling pretrained DL models for on-the-fly inference
- Functionalities
  - Expose API for retrieving and updating annotations
  - Periodically update models using user-calibrated annotations

Human-in-the-Loop: Retrieve Model Predictions



1. The client sends GET request for annotations of the target FOV
2. The server triggers the inference function of DL models
3. DL models generate on-the-fly segmentation results
4. The server sends back GET response with model-generated annotations

Human-in-the-Loop: Save User Modifications



1. The client sends PUSH request with modified annotations
2. The server periodically updates DL models using modified annotations saved in the database

Summary

Summary: Conclusion

- Interleave annotation with inference
- Minimal amount of labeling required
- Improve efficiency

Summary: Future Work

- Model Refining
- Ensemble

Thank You

Appendix

References

[1] **Development and evaluation of deep learning–based segmentation of histologic structures in the kidney cortex with multiple histologic stains**
Kidney International DOI: https://doi.org/10.1016/j.kint.2020.07.044
[2] **Deep Learning–Based Histopathologic Assessment of Kidney Tissue**
JASN October 2019, 30 (10) 1968-1979; DOI: https://doi.org/10.1681/ASN.2019020144
[3] **Deep Learning–Based Segmentation and Quantification in Experimental Kidney Histopathology**
JASN November 2020, ASN.2020050597; DOI: https://doi.org/10.1681/ASN.2020050597

Related Code and Format Reference

[4] **Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2”**
https://github.com/facebookresearch/detectron2, 2019.
[5] **Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., ... & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. arXiv 2014. arXiv preprint arXiv:1405.0312.**
https://cocodataset.org/#home
[6] **ASAP - Automated Slide Analysis Platform**
https://github.com/computationalpathologygroup/ASAP

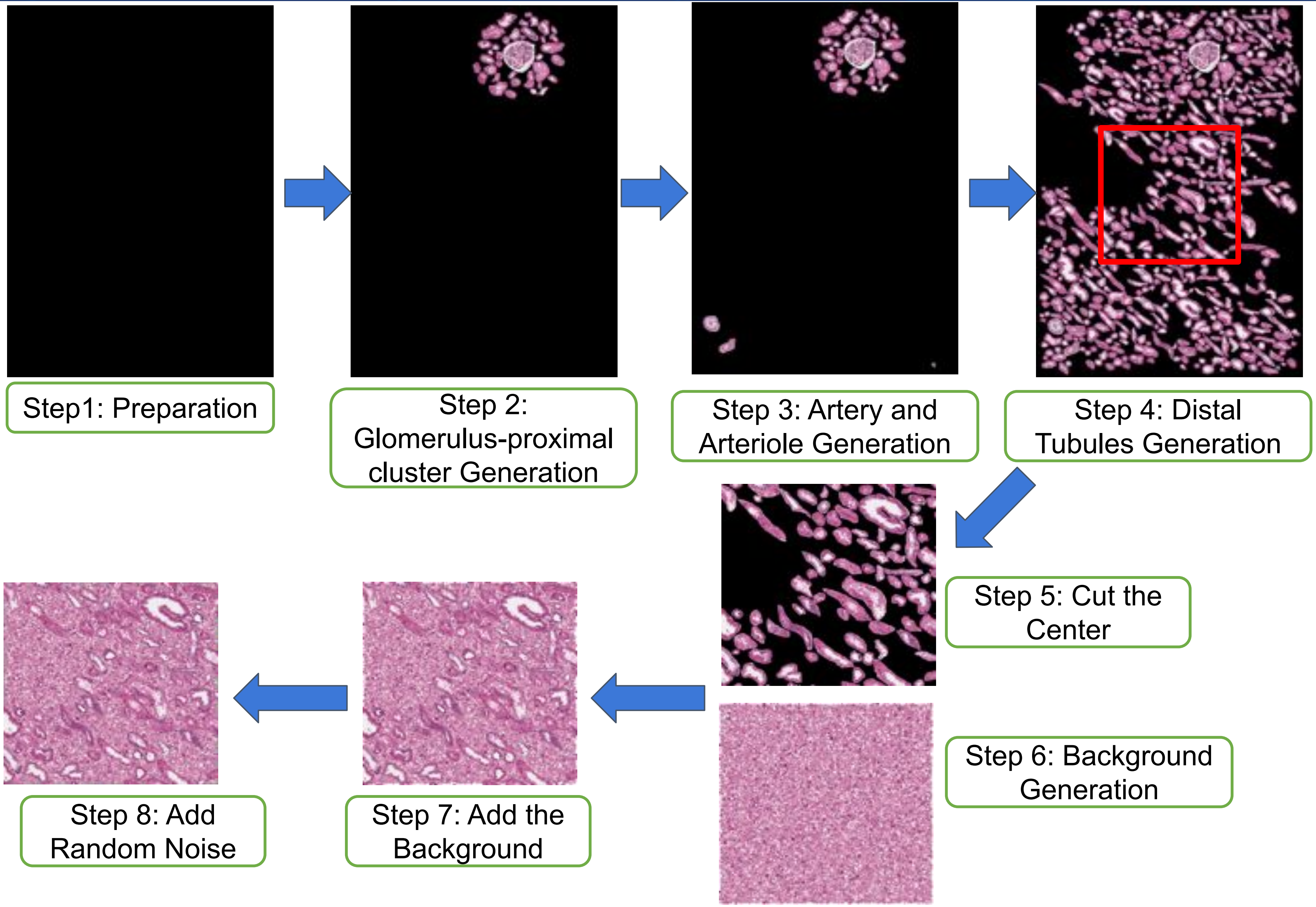
Code Repo:

https://github.com/Auto-annotation-of-Pathology-Images/AAPI_code

Collage Generator

Details:

1. Item Generation:
   - Randomly Picking: we pick the item randomly without replacement
   - Augmentation: we augment the item by randomly doing:
     - flip
     - grid distortion
     - transpose
     - translating
     - scaling
     - rotation
2. Insertion
3. Background Generation



Columbia University

Data Science Institute