Workshop No.2 — Kaggle Systems Engineering Analysis Report

Juan David Buitrago Rodriguez - 20242020194
David Giovanni Aza Carvajal - 20241020137

May 2025

This report lays out the plan for building a convolutional neural network system to analyze histopathological images, specifically tilted to work within Kaggle's computing constraints. It pulls together key takeaways from Workshop 1 to create a practical, scalable, and transparent pipeline for classifying kidney tissue samples.

Workshop No. 1 Findings

In order to proceed with further planning and analysis, let's summarize the main findings, constraints, data characteristics, and any chaos-theory factors.

Main Findings and Constraints

- **Critical Insights**
  The system must handle high biological variability and structural complexity while preserving fine-grained details like capillary walls. The analysis revealed that preprocessing steps and labeling inconsistencies directly impact the ability of models to generalize and segment accurately.
- **Data Fragility**
  Any distortion or over-simplification during processing risks the elimination of valuable microanatomical information critical for accurate classification.

Data Properties

- **Biological Variability**
  Differences in donor tissues and staining methods introduce heterogeneity that complicates model learning.
- **Structural Similarity**
  Capillaries often mimic other tissue elements, increasing false positive rates.
- **Image Complexity**
  High-resolution images include dense and overlapping structures requiring careful filtering.

CNN, Serial Imaging, and Radon Projection

- **CNN Suitability**
  Convolutional Neural Networks (CNNs) are well-suited for biomedical image analysis due to their ability to extract hierarchical spatial features. They help distinguish capillary patterns despite visual noise.

- **Serial Image Analysis**
  Applying CNNs across sequential slices allows tracking of vascular continuity, enhancing spatial coherence in predictions.

- **Radon Transform Projection**
  This preprocessing technique emphasizes linear structures (like vessels) by projecting the image at various angles, helping CNNs to detect vasculature that may otherwise be obscured.

Chaos-Like Dynamics

- **Instability in Results**
  Minor data variations (e.g., from staining or preparation artifacts) can unpredictably alter segmentation outcomes representing a breach for chaos-like sensitivity.

- **Risk Amplification**
  Without careful control, models can fixate on irrelevant visual patterns, leading to overfitting or unreliable performance.

System Requirements

Performance Requirements

- **Performance under limitations**
  All models must be trained and be within Kaggle's compute limits. Optimization through batch size tuning, mixed precision, and pruning will be considered.

- **Risk Amplification**
  Models can sometimes focus on unimportant details in images, which causes poor performance or overfitting (memorizing data instead of learning). We need to prevent this.

- **Serial Image Analysis**
  When looking at a series of images, the model should keep consistency across them, ensuring each image slice matches the others in terms of position and structure.

- **Runtime Efficiency**
  The system must quickly analyze small image sections to process entire tissue samples in a reasonable amount of time.

- **Computational and Accuracy Targets:**
  - **Accuracy** Achieve at least 90% accuracy on the validation dataset.
  - **Training** Finish training within 50 rounds, stopping early if the model stops improving

- **Robustness and Reliability:** The model must handle variability in input images without significant drops in accuracy. The system should maintain stable performance across at least three different test sets from distinct sources.

- **Scalability:** The system must process huge whole-slide images (over 500MB) efficiently, extracting small patches and managing memory to avoid crashes or timeouts.

CNN Implementation Requirements

- **Adaptability to Image Variations**
  The system needs to handle typical issues in histopathological slides, like uneven staining, slight tissue warping, or random noise and artifacts. When tested with artificial distortions, the model's accuracy should only drop by up to 5%, so it stays reliable no matter the input quirks.
- **Consistency in Structural Orientation**
  Before feeding images into the model, preprocessing steps should line up key anatomical features properly. This is especially crucial for things like blood vessels, where consistent alignment makes detection more dependable across different patients and imaging setups.
- **Continuous Monitoring and Error Awareness**
  Throughout both training and inference stages, the system should track essential indicators such as prediction confidence, class balance, and spikes in validation loss. In case of irregular behavior—such as sudden uncertainty or unexpected data shifts—the system must be able to trigger alerts or flag specific cases for further inspection, maintaining operational reliability even in edge cases.
- **Transparency and Result Verification**
  Model outputs should be accompanied by clear visual aids that support expert interpretation. These may include attention-based visualizations or error maps highlighting problematic regions. Moreover, every inference session should be logged in detail, recording performance metrics, input conditions, and other contextual data to ensure traceability and enable thorough post-hoc analysis when needed.

Reliability Requirements

- **Model Robustness**
  The system must handle various imaging artifacts and outliers (e.g., noise, irregular lighting, motion blur) through data augmentation and regularization.

- **Data Integrity During Processing**
  There is a known risk of losing valuable anatomical details during steps like resizing or thresholding. Preprocessing must minimize data loss while normalizing inputs.

- **Output Stability Across Slices**
  The output of the CNN must stay consistent across neighboring slices. Any errors or instability in the segmentation should be detected and flagged.

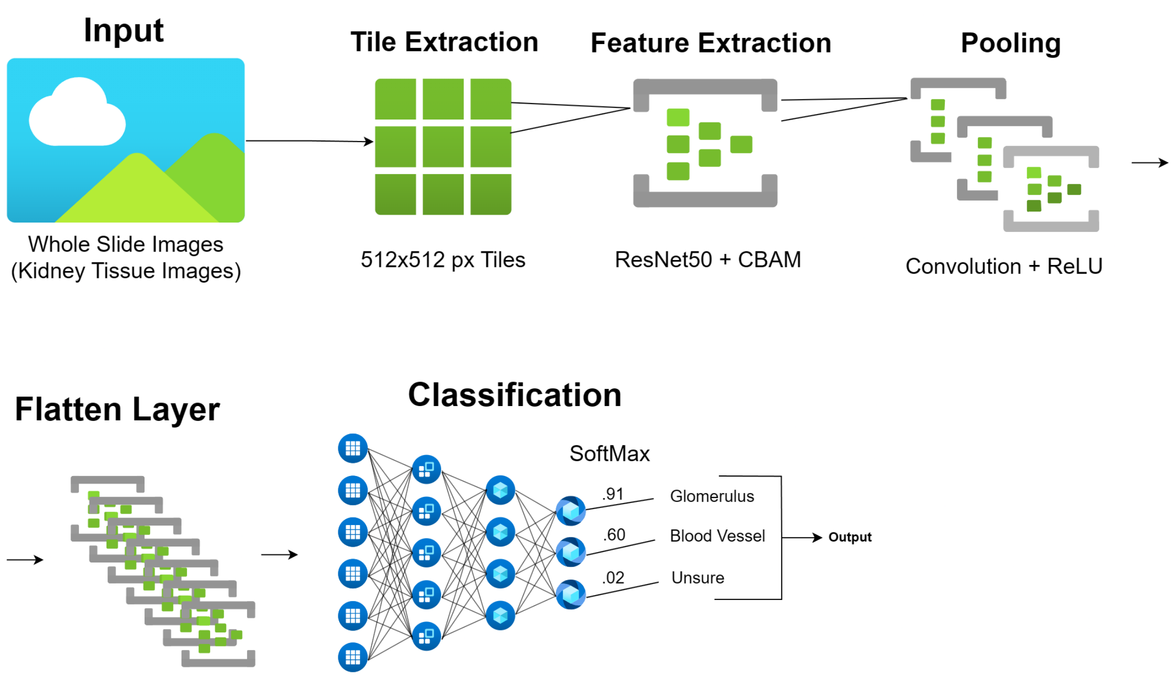
Architecture

Core Workflow of Histological Images

Table 1: Main Pipeline Stages in Histological Image Analysis

Stage	Brief Description
Slide Preparation	Tissue is fixed, sectioned, and stained for histological examination.
Slide Digitization	Prepared slides are scanned to produce high-resolution Whole Slide Images (WSIs).
Region of Interest Selection	Manual or automated selection of relevant tissue areas for analysis.
Patch Extraction	WSIs are divided into smaller image patches suitable for computational processing.
Deep Learning Prediction	CNN-based models analyze patches to classify, segment, or detect features of interest.

The diagram below shows the system's CNN architecture, highlighting the main components and how data moves through the pipeline to process whole-slide kidney images.



Modules description

- **Whole Slide Images (Input)**
  Receives high-resolution kidney tissue images, typically in .tiff format, as the input for analysis.
- **Tile Extraction**
  Divides the large WSI into smaller images "tiles" of size 512x512 pixels to make computation manageable and localized.

- **Feature Extraction**
  Uses ResNet50 as a backbone CNN to extract spatial features from the tiles. CBAM (Convolutional Block Attention Module) enhances attention to relevant tissue regions (e.g., glomeruli, blood vessels).

- **Convolution + ReLU + Pooling**
  Refines features using convolutional layers followed by ReLU activation and downsampling with pooling layers.

- **Flatten Layer**
  Transforms the multi-dimensional feature maps into a one-dimensional feature vector suitable for input into dense layers.

- **Classification (Fully Connected + Softmax)**
  Uses fully connected layers to perform classification into categories such as:

  - **Glomerulus**
  - **Blood Vessel**
  - **Unsure**

  Final output is produced using a Softmax activation for probability scores.

Sensitivity and Chaos

- **Handling High Sensitivity and Noise:** The CNN uses CBAM (Convolutional Block Attention Module) to help the model focus on important parts of the image and ignore irrelevant details or noise. This helps the model perform better even when images vary slightly or have inconsistent colors.

- **Managing Unexpected Changes:** To prepare the model for unusual inputs, we can introduce to different image inputs such as changing contrast, or slightly deforming the images. This teaches the model to handle a wide variety of cases.

- **Uncertain Predictions and Class Imbalance:** We monitor the model's output confidence. If it's not sure about a prediction (e.g., when all class probabilities are similar), that result is flagged for review or sent through an additional process (like an extra model or voting system).

- **Error Detection and Monitoring:** We use tools like TensorBoard and custom alerts to detect problems during training or testing. If the model suddenly performs worse or behaves strangely, we get notified to investigate the issue.

- **Tracking Model Stability:** During use, we track things like average prediction confidence and variability. If the model behaves inconsistently or produces unusual results, we can spot and review those specific cases.

Technical Stack and Implementation Sketch

As it was explained along the document we are focusing on the CNN implementation since they are a specialized class of deep neural networks, designed to efficiently process grid-like data structures such as images. They excel in capturing spatial hierarchies and extracting features from input data using layers of learnable filters and operations.(Mienye et al., 2025)

CNN Implementation Tools

- **PyTorch**
  Core deep learning framework used for model construction, training, and deployment. Offers flexibility and access to pretrained models such as ResNet50.

- **Feature Extraction Models**
  - **ResNet50**
    Deep residual network used for initial feature extraction from image tiles.
  - **CBAM (Convolutional Block Attention Module)**
    Attention mechanism added to ResNet50 to enhance focus on relevant regions (e.g., glomeruli, blood vessels).

- **OpenSlide + Pydicom**
  OpenSlide is used for reading Whole Slide Images (WSIs), while Pydicom handles DICOM-formatted medical images.

- **Radon Transform**
  Used for vertical projection and alignment analysis, enhancing the detection of vascular structures following tissue orientation.

- **NumPy / pandas**
  Employed for image preprocessing, patch extraction (e.g., 512x512 px), dataset organization, and metric evaluation.

- **Torchvision / Albumentations**
  Libraries used for data augmentation and image transformation during training.

- **Matplotlib / Seaborn / TensorBoard**
  Visualization tools for plotting training metrics, confusion matrices, and model interpretability outputs.

References

Mienye, I. D., Swart, T. G., Obaido, G., Jordan, M., & Ilono, P. (2025). Deep convolutional neural networks in medical image analysis: A review. *Information*, *16*(3). https://doi.org/10.3390/info16030195