PROJECT EVALUATION REPORT

TITLE:

Comprehensive Analysis of Histopathological Images: Classification and Clustering of Lung and Colon Tissue Types

PREPARED BY:

Anoop Bhatia

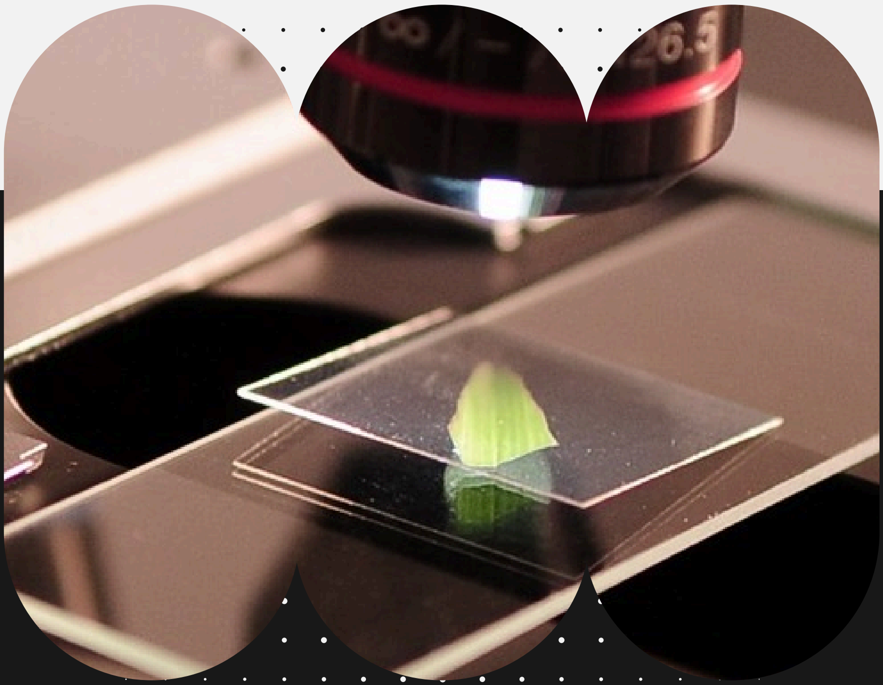
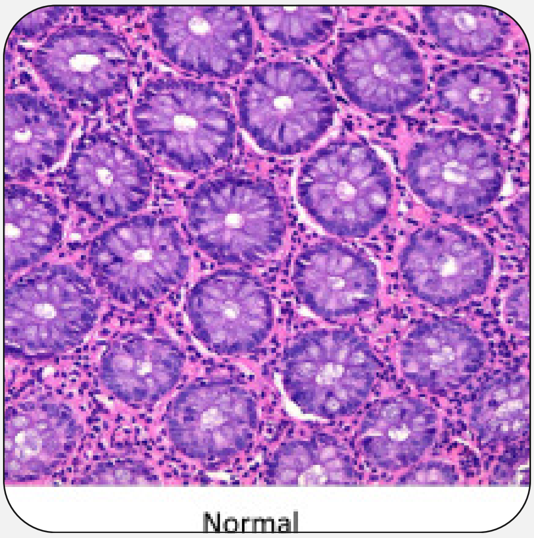


TABLE OF CONTENTS

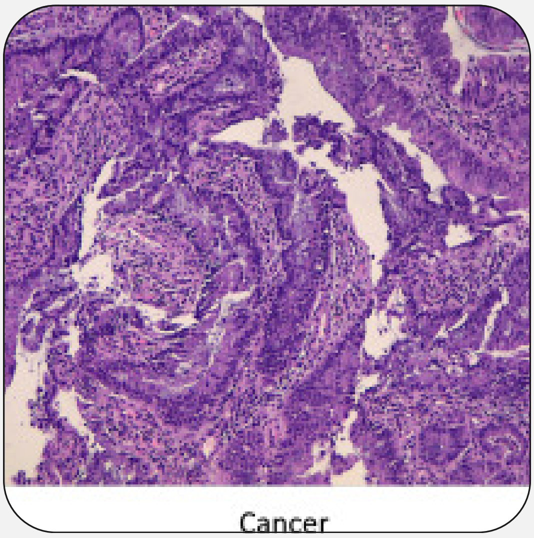
ABOUT THE DATASET

Source : Kaggle
Data Set Link
25000 images of 5 classes including lung and colon cancer and healthy samples.

This dataset comprises 25,000 histopathological images, each 768 x 768 pixels in size and in JPEG format, organized into five distinct classes. Originating from a validated HIPAA-compliant sample, it includes 750 images of lung tissue (250 each of benign tissue, adenocarcinomas, and squamous cell carcinomas) and 500 images of colon tissue (250 each of benign tissue and adenocarcinomas). To reach the total of 25,000 images, the dataset was augmented using the Augmentor package, with each class containing exactly 5,000 images.



Normal



Cancer

PROJECT OVERVIEW

Summary

This project focuses on the classification of lung and colon histopathological images using deep learning techniques. The primary objective is to develop a robust model capable of distinguishing between normal and cancerous tissues, including specific types of cancers such as adenocarcinoma and squamous cell carcinoma, based on histopathological images. This classification task is crucial for aiding in the diagnosis and treatment planning for patients with lung and colon cancer.

Tech Stack

- **Python**: The programming language used for the entire implementation process, including data preprocessing, model development, and evaluation.
- **PyTorch**: A deep learning framework that facilitates the construction and training of the convolutional neural network (CNN) used for image classification.
- **Torchvision**: A package within PyTorch that provides utilities for image transformations, dataset handling, and pre-trained models.
- **Matplotlib and Seaborn**: Libraries employed for visualizing data, including plotting histopathological images and creating confusion matrices to evaluate model performance.
- **Scikit-learn**: A machine learning library used for tasks such as splitting the dataset into training and validation sets, and calculating evaluation metrics like confusion matrices.

METHODOLOGY

1. Data Preparation and Organization

Environment Setup:

- Libraries such as numpy, pandas, torch, torchvision, and matplotlib were imported to handle data processing, image transformations, model operations, and visualization.

Directory Exploration:

- The dataset located in ./lung_colon_image_set was inspected to understand its structure and contents. This involved:
  - Main Directory: Listing contents to ensure subdirectories train and val are present.
  - Subdirectories: Listing images within train and val to confirm dataset distribution.

Data Splitting and Organization:

- Splitting Method: The dataset was split using an 80-20 ratio:
  - Training Set: 80% of the total images (16,000 images).
  - Validation Set: 20% of the total images (4,000 images).
- File Movement: Images were moved into class-specific subdirectories within train and val folders. Class names included colon_aca, colon_n, lung_aca, lung_n, and lung_scc.
- Training Set:
  - colon_aca: 4,000 images
  - colon_n: 4,000 images
  - lung_aca: 4,000 images
  - lung_n: 4,000 images
  - lung_scc: 4,000 images
- Validation Set:
  - colon_aca: 1,000 images
  - colon_n: 1,000 images
  - lung_aca: 1,000 images
  - lung_n: 1,000 images
  - lung_scc: 1,000 images

.

METHODOLOGY

2. Model Training and Evaluation

Model Initialization and Training Setup:

- Model: ResNet-18 was utilized, initialized with pre-trained weights.
- Loss Function: CrossEntropyLoss was used for classification tasks.
- Optimizer: Adam optimizer was chosen for its adaptive learning rate capabilities.

Training Process:

- Epochs and Loss: The model was trained over 15 epochs with the following loss values recorded:
    - Epoch Loss Values: Started from 0.1721 and decreased to 0.0201.
- Performance Trends: The model showed a decrease in loss over epochs, indicating effective learning with minimal overfitting.

Validation and Evaluation:

- Validation Loss and Accuracy:
    - Validation Loss: 0.0175
    - Validation Accuracy: 99.46%
- Interpretation: The low validation loss and high accuracy reflect excellent model performance with effective generalization to unseen data.

Visualization and Model Assessment:

- Image Visualization: Selected validation images were visualized alongside predicted and actual labels for qualitative assessment.
- Confusion Matrix: A confusion matrix was used to evaluate model performance across different classes, revealing accurate predictions and areas for improvement.

Model Saving:

- Saving: The trained model was saved using torch.save() for future deployment and continued use.

METHODOLOGY

3. Feature Extraction

Loading the Pretrained Model:

- Model Selection: The Model previously saved was chosen for its robust feature extraction capabilities.
- Setup: The model was set to evaluation mode (model.eval()) to ensure accurate feature extraction.

Data Preparation:

- Transformations: Images were processed using a series of transformations:
  - Resize: Images were resized to a uniform dimension (e.g., 224x224 pixels).
  - Crop: Center cropping was applied to focus on the central part of images.
  - Normalization: Pixel values were normalized to match the model's expected input range.
  - Conversion: Images were converted to tensors for processing by the model.
- Data Loaders: Data loaders were created for both training and validation datasets, facilitating efficient batching and shuffling.

Feature Extraction and Saving:

- Extraction Process: Features were extracted by passing images through the pretrained model (excluding the final classification layer).
- Saving: Extracted features were saved as .npy files for both training and validation datasets.
- Statistics:
  - Training Features Shape: (20,000 samples, 512 features per sample)
  - Validation Features Shape: (5,000 samples, 512 features per sample)

METHODOLOGY

4. Clustering Analysis

KMeans Clustering:

- Training Data:
  - Clusters: KMeans clustering with 20 clusters was applied.
  - Cluster Centers: The centroids of the 20 clusters in the feature space were computed.
  - Cluster Labels: Each training sample was assigned to one of the 20 clusters.
  - Statistics:
    - Cluster Centers: Represented as arrays of 512 features each.
    - Example Labels: [4, 14, 14, 14, 14, 14, 16, 16, 4, 12]
- Validation Data:
  - Cluster Centers and Labels: Similar format as training data, with clusters specific to the validation set.
  - Example Labels: [4, 11, 4, 11, 9, 11, 11, 11, 9, 4]

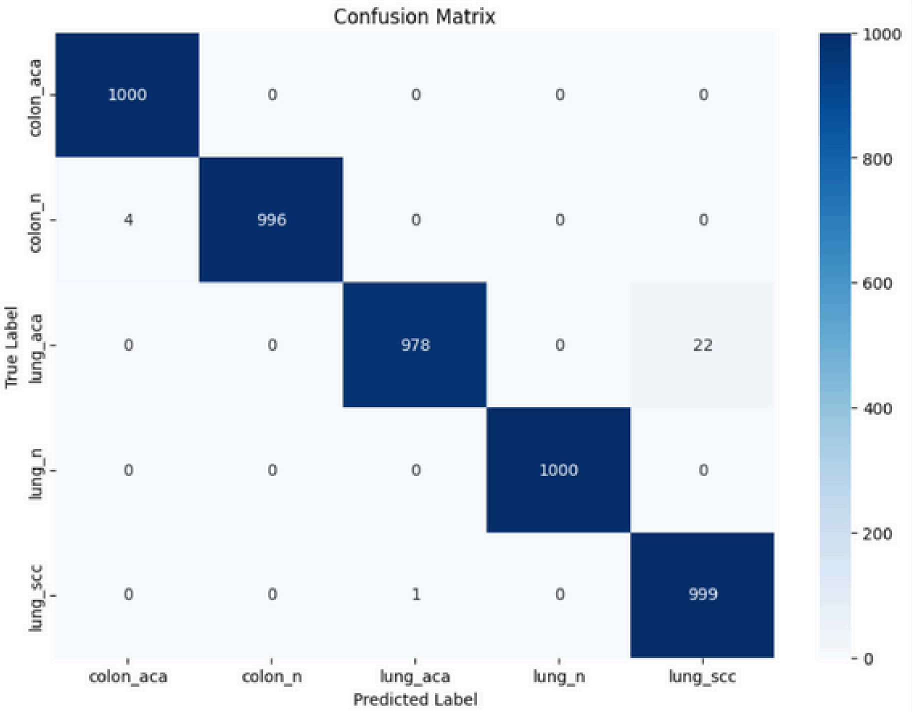
Dimensionality Reduction and Visualization:

- Techniques Used:
  - PCA: Principal Component Analysis was used to reduce dimensionality to 2D.
  - t-SNE: t-Distributed Stochastic Neighbor Embedding provided a more nuanced visualization of clusters.
  - Data Visualization: Scatter plots showed feature distribution and clustering patterns.

Clustering Metrics:

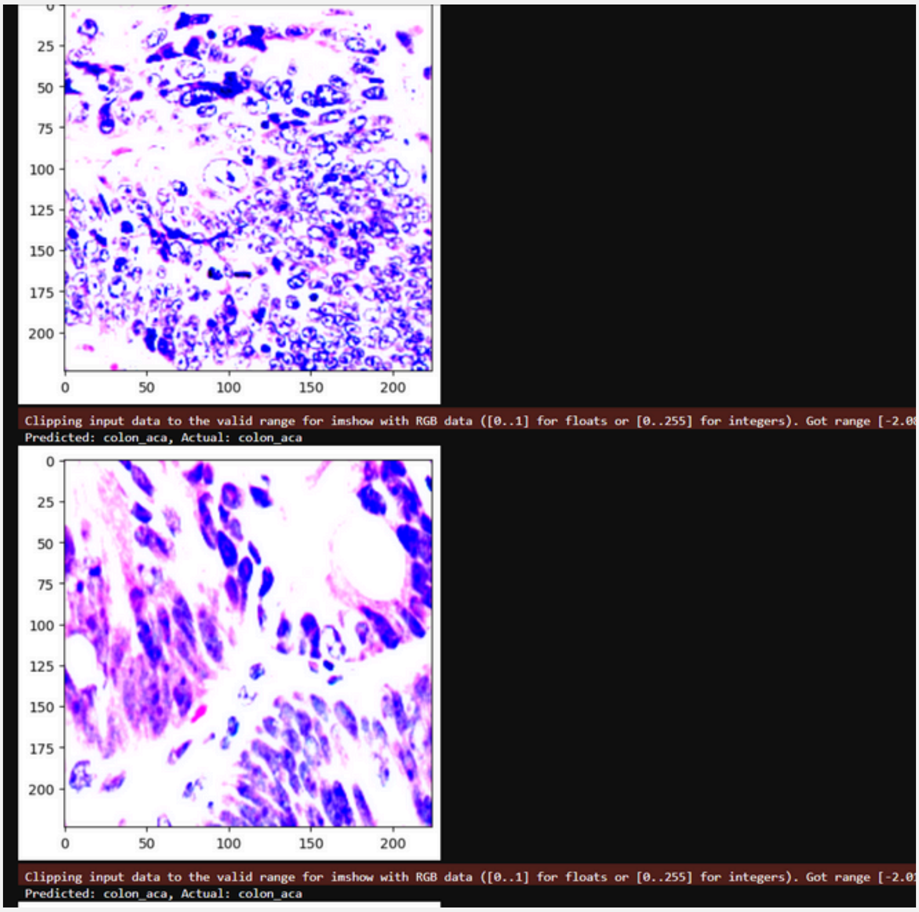
- Training Data:
  - Silhouette Score: 0.381 (moderate cluster separation)
  - Davies-Bouldin Index: 0.969 (indicates reasonable cluster distinctiveness)
  - Calinski-Harabasz Index: 11,044.23 (high score suggests well-separated clusters)
- Validation Data:
  - Silhouette Score: 0.424 (slightly better separation compared to training data)
  - Davies-Bouldin Index: 0.806 (improved distinctiveness)
  - Calinski-Harabasz Index: 3,143.12 (indicates good cluster separation)

KEY FINDINGS



True Label \ Predicted Label	colon_aca	colon_n	lung_aca	lung_n	lung_scc
colon_aca	1000	0	0	0	0
colon_n	4	996	0	0	0
lung_aca	0	0	978	0	22
lung_n	0	0	0	1000	0
lung_scc	0	0	1	0	999

Confusion Matrix for Model Predictions on Validation Set

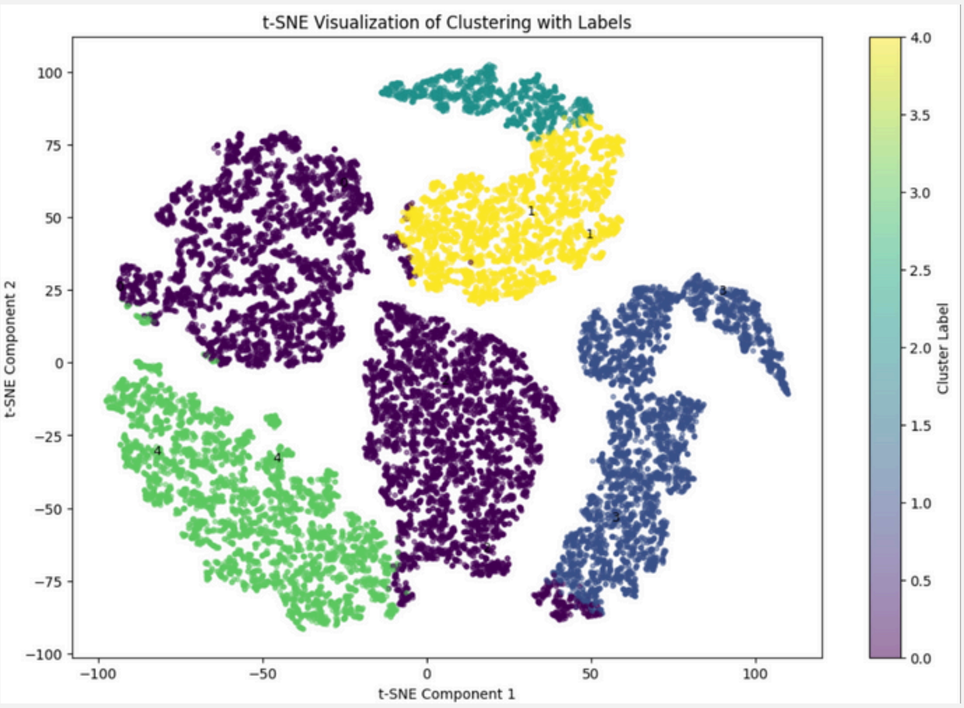


```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-2.0
Predicted: colon_aca, Actual: colon_aca
```

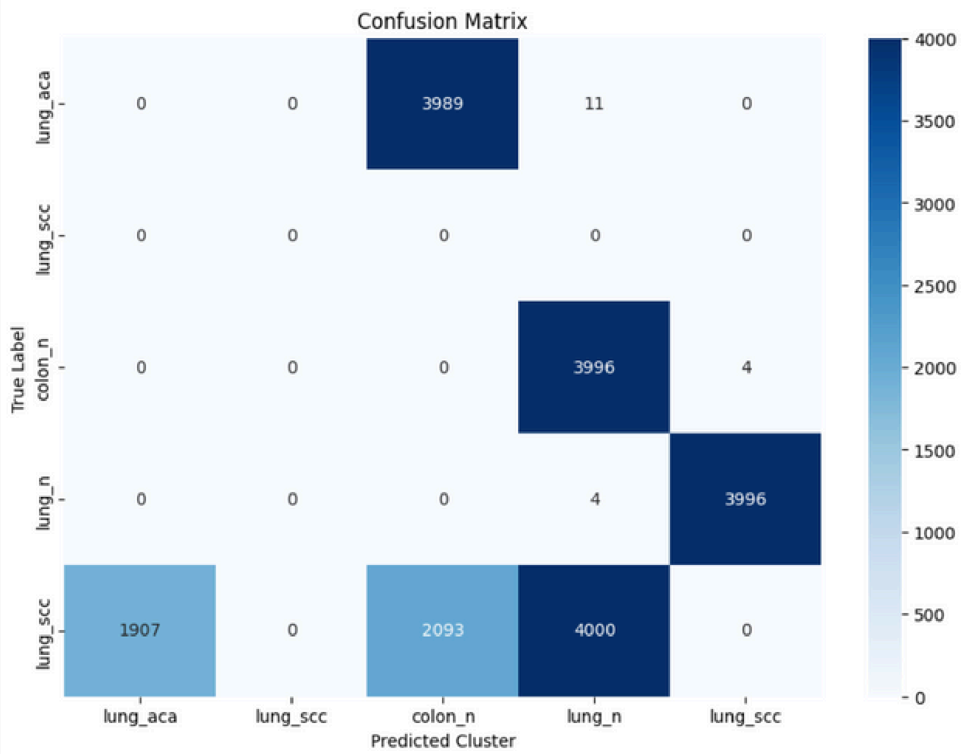
```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers). Got range [-2.0
Predicted: colon_aca, Actual: colon_aca
```

"Validation Set: Predictions vs. Actual Labels"

KEY FINDINGS

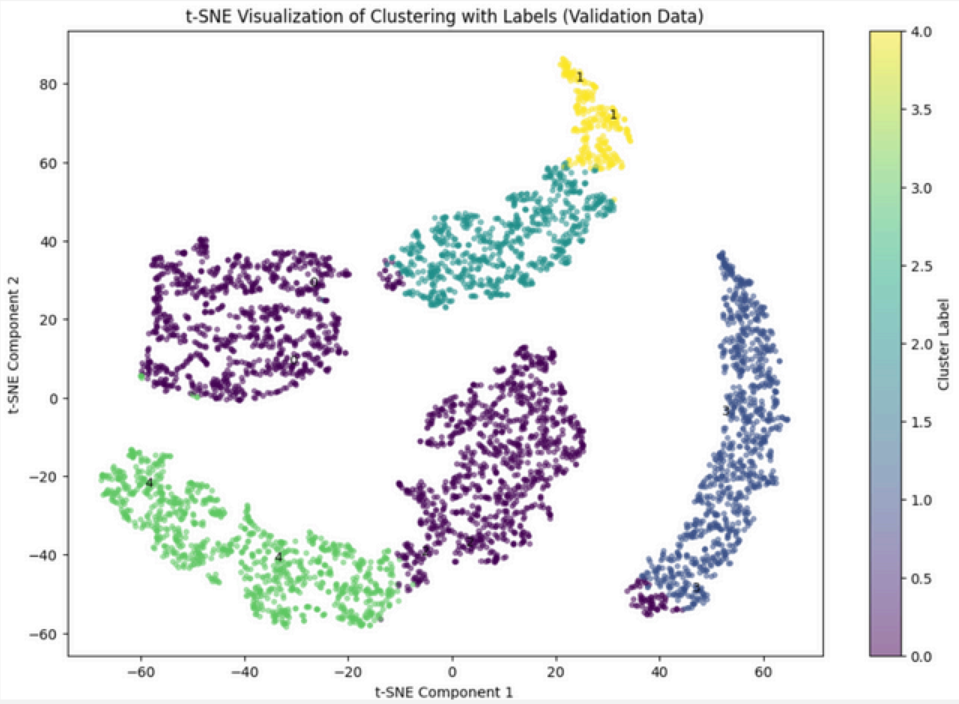


t-SNE Clustering Visualization with Labels
of training data

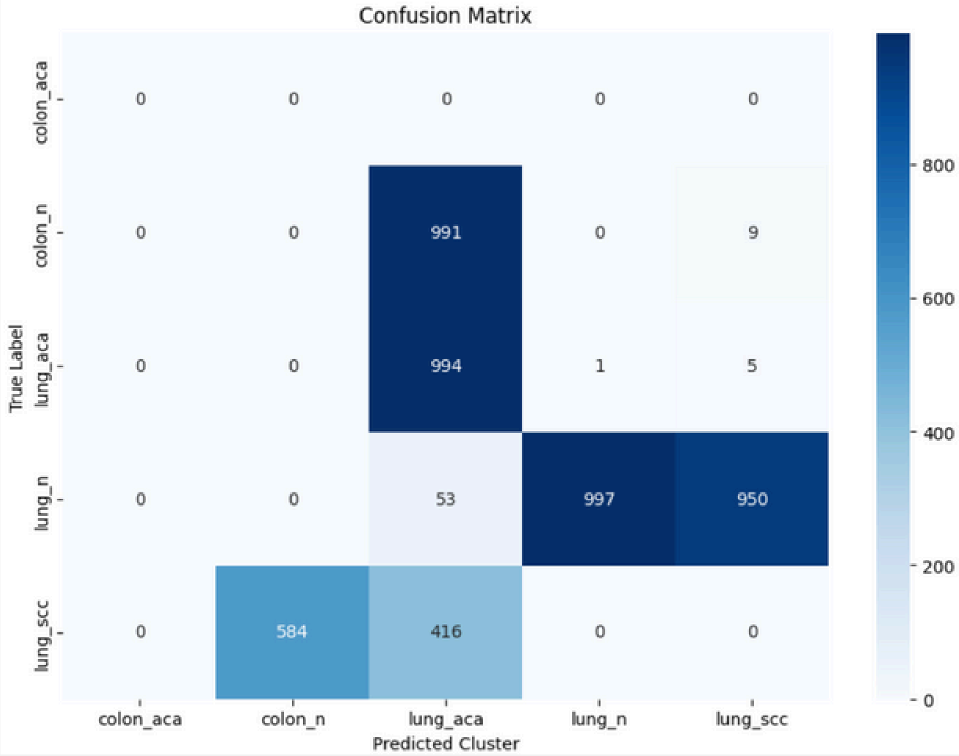


Confusion Matrix of True Labels vs. Predicted Clusters (Training Data)

KEY FINDINGS



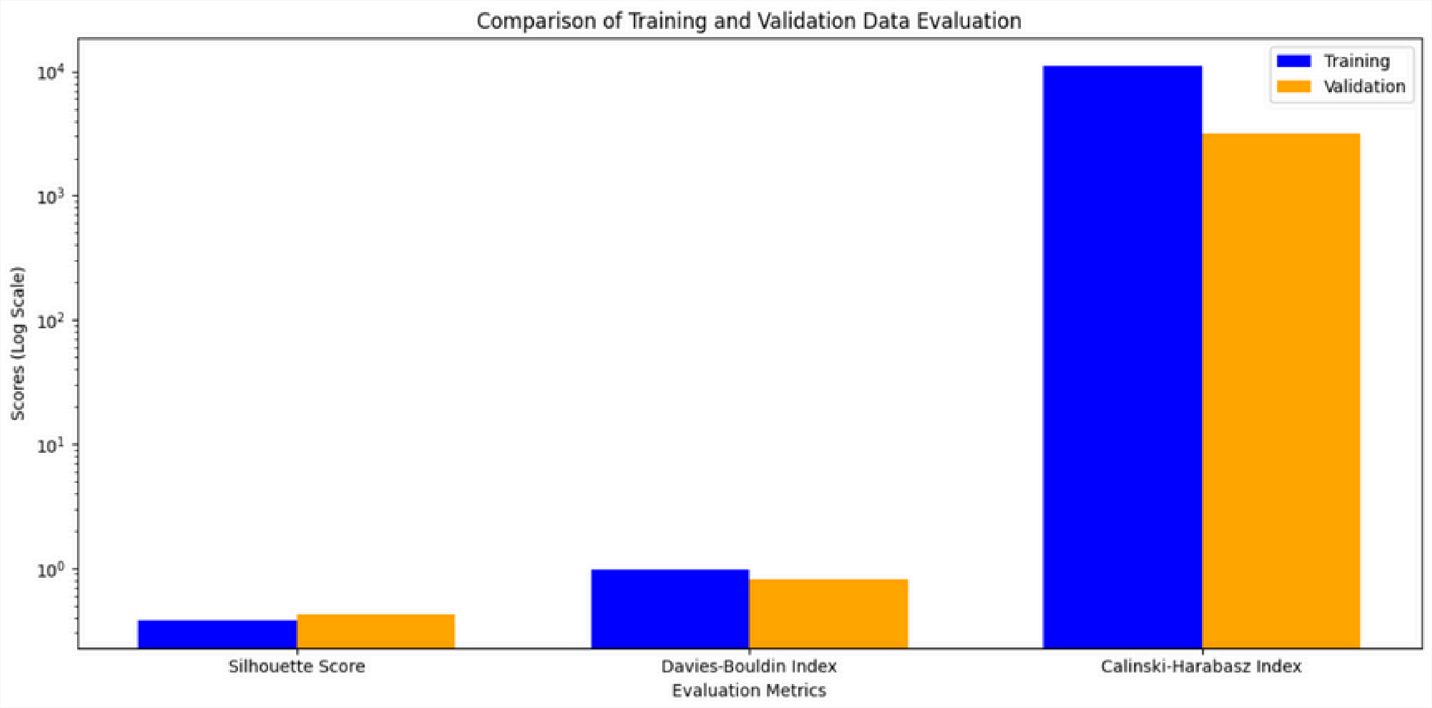
t-SNE Clustering Visualization with Labels of validation data

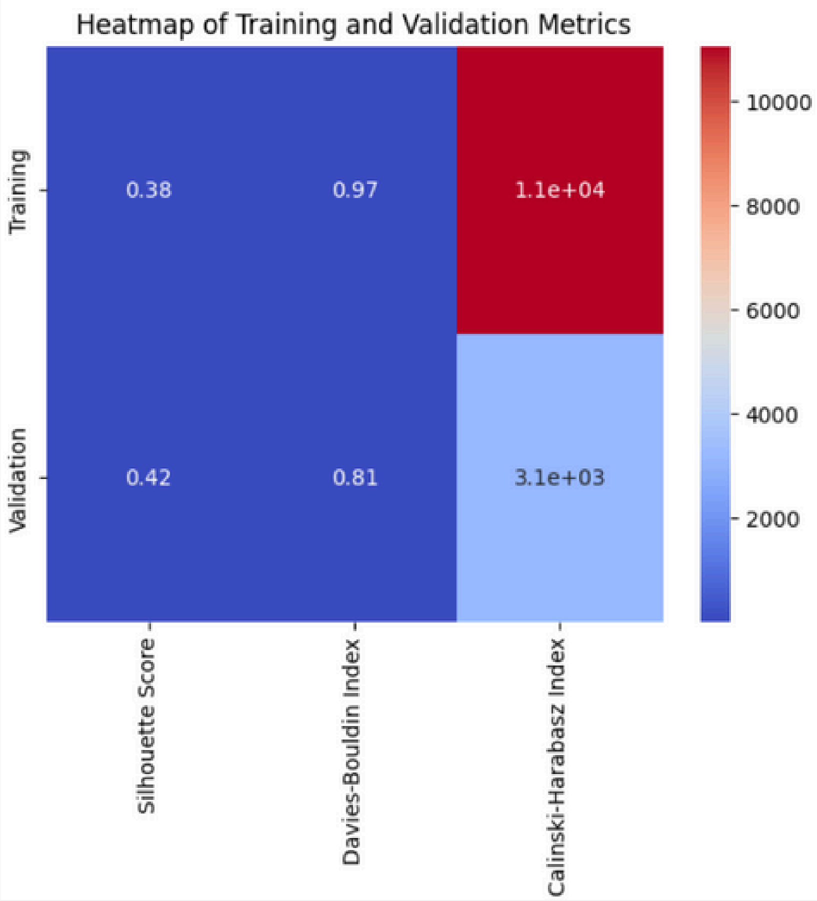


True Label \ Predicted Cluster	colon_aca	colon_n	lung_aca	lung_n	lung_scc
colon_aca	0	0	0	0	0
colon_n	0	0	991	0	9
lung_aca	0	0	994	1	5
lung_n	0	0	53	997	950
lung_scc	0	584	416	0	0

Confusion Matrix of True Labels vs. Predicted Clusters (Validation Data)

KEY FINDINGS





Comparison of Clustering Evaluation Metrics: Training vs. Validation Data

CONCLUSION

The feature extraction and clustering process effectively organized and analyzed the image data, providing valuable insights into the dataset's structure. However, the evaluation metrics indicate room for improvement. Specifically, the Calinski-Harabasz Index suggests that while the clustering model performs adequately on the training data, it may not generalize as well to the validation data.

To enhance the model's generalization performance, future work could focus on tuning the clustering algorithm and exploring regularization techniques. Additionally, incorporating more advanced feature extraction methods or refining the preprocessing steps could further optimize the results. This report lays a solid foundation for these improvements, guiding future efforts in achieving more robust and generalizable clustering outcomes.