Optimizing Cervical Cancer Diagnosis: Image Classification and GenAI

Computer Vision and Deep Learning Final project

Our Team

Liang Gong

Gynecologic Oncologists Pathologists

Colleen Jung

Program Manager

Yanchen Dong

Lead Data Scientist

David Kuo

Machine Learning Engineer

Business Problem: Cervical Cancer

Current Problem	Expected Outcome
Early-stage cervical cancer often goes undetected. Diagnostic inconsistencies. Variability in diagnostic accuracy. Time-consuming conventional diagnostic methods.	Aid healthcare professionals in making informed quick diagnoses.
Not enough cervical cancer cells from pap smear tests for training purpose.	Generate more synthetic cervical cancer cells to train medical students

https://www.researchgate.net/publication/327995161_Sipakmed_A_New_Dataset_for_Feature_and_Image_Based_Classification_of_Normal_and_Pathological_Cervical_Cells_in_Pap_Smear_Images

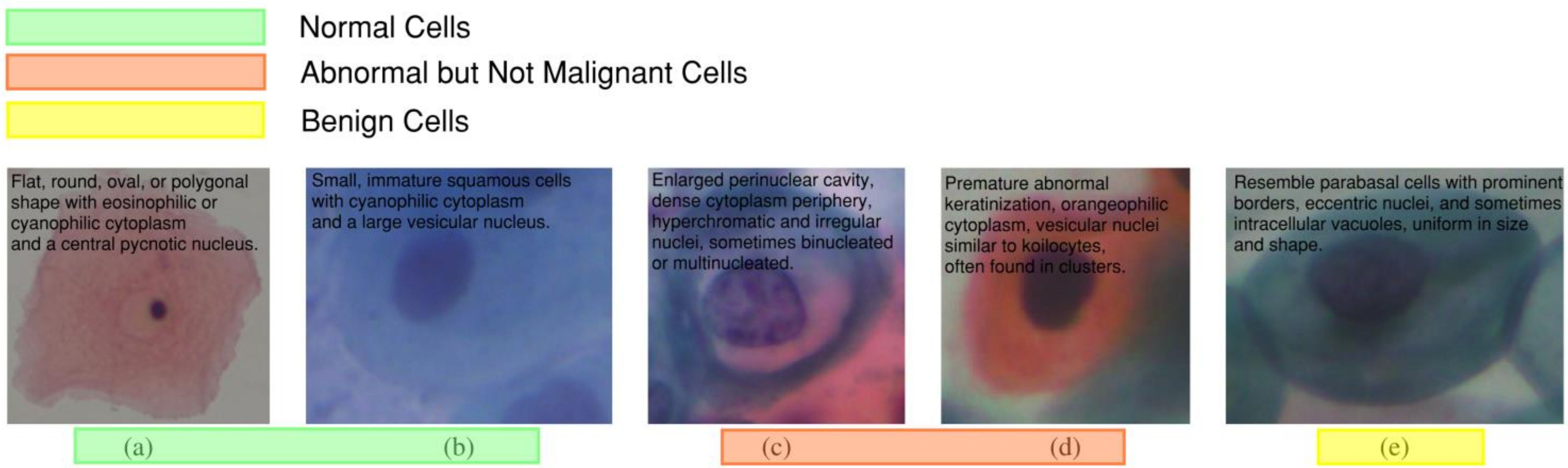


Fig. 1. Cell images of five categories: (a) Superficial-Intermediate, (b) Parabasal, (c) Koilocytotic, (d) Dyskeratotic, (e) Metaplastic.

Treatment Approaches

- **Normal Cells**: Regular monitoring and routine Pap smears as part of standard gynecological care.
- **Abnormal but Not Malignant Cells**:
  - **Koilocytotic Cells**: Close monitoring and follow-up Pap smears. If persistent, may require colposcopy and biopsy to rule out progression to cancer.
  - **Dyskeratotic Cells**: Similar to koilocytotic cells, requiring careful monitoring, follow-up tests, and possible biopsy.
- **Benign Cells**:
  - **Metaplastic Cells**:Regular follow-up to monitor any changes. Generally, no treatment needed unless associated with pre-cancerous lesions, in which case further investigation and possible treatment may be required.

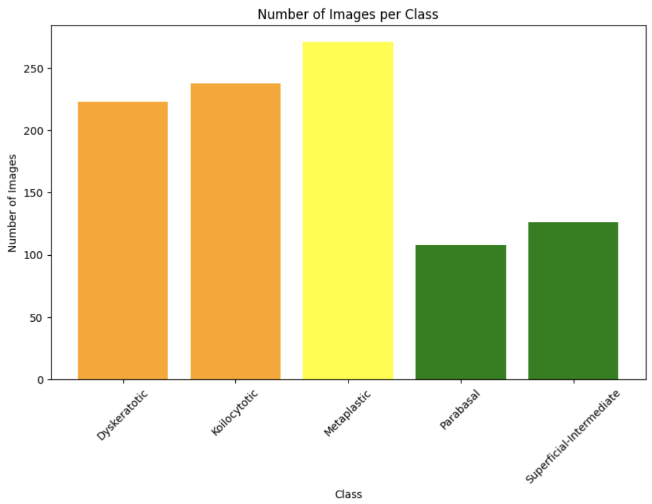
Project Overview - Abstract

Overview

Problem Statement
Cervical cancer, often undetected in its early stages, poses significant diagnostic challenges. Variability in diagnostic accuracy and the time-consuming nature of conventional methods necessitate the development of more reliable and efficient diagnostic tools.

Approach
We used Convolutional Neural Networks (CNN) and Transfer Learning to enhance cervical cancer detection accuracy and efficiency. Synthetic data were generated with Conditional Generative Adversarial Networks (CGAN) and Conditional Variational Autoencoders (CVAE). More advanced GenAI methods were developed by our research engineer.

Data Overview
Dataset: SIPaKMeD Database with 4049 annotated cervical cell images
Categories:
Superficial-Intermediate,
Parabasal,
Koilocytotic,
Dyskeratotic, Metaplastic.



Model Training and Evaluation

#	Cognitive Problem	Models	Results
1	Classification	VGG16 ,ResNet50	96% accuracy
2	Image Generation	CGAN, CVAE	90 loss

Key Deliverables and Timelines

#	Deliverable	Description	Est Due date
1	Data Collection	Collection and preprocessing of cervical cell images.	7/20/2024
2	Model Development (Cognitive Problem 1: Classification)	Development of CNN and Transfer Learning models	7/27/2024
3	Synthetic Image Generation (Cognitive Problem 2: Data Augmentation)	Generation of synthetic data using GAN and VAE to augment the dataset.	8/3/2024
4	Model Evaluation	Detailed evaluation and comparison of model performance for both cognitive problems.	8/6/2024
5	Deployment	Deployment of the web application for model access and usage.	8/17/2024

Dataset Review

Category	Num of Images	Num of Cells
Superficial/Intermediate	126	813
Parabasal	108	787
Koilocytotic	238	825
Metaplastic	271	793
Dyskeratotic	223	813
Total	966	4049

SIPaKMeD Database:

- Contains 4049 annotated cervical cell images from Pap smears.
- Images classified into five categories: Superficial-Intermediate,
- Parabasal, Koilocytotic, Dyskeratotic, Metaplastic.
- Used for feature and image-based classification.

Challenges in Classification:

- Variability in cell morphology and image quality.

Feature Extraction:

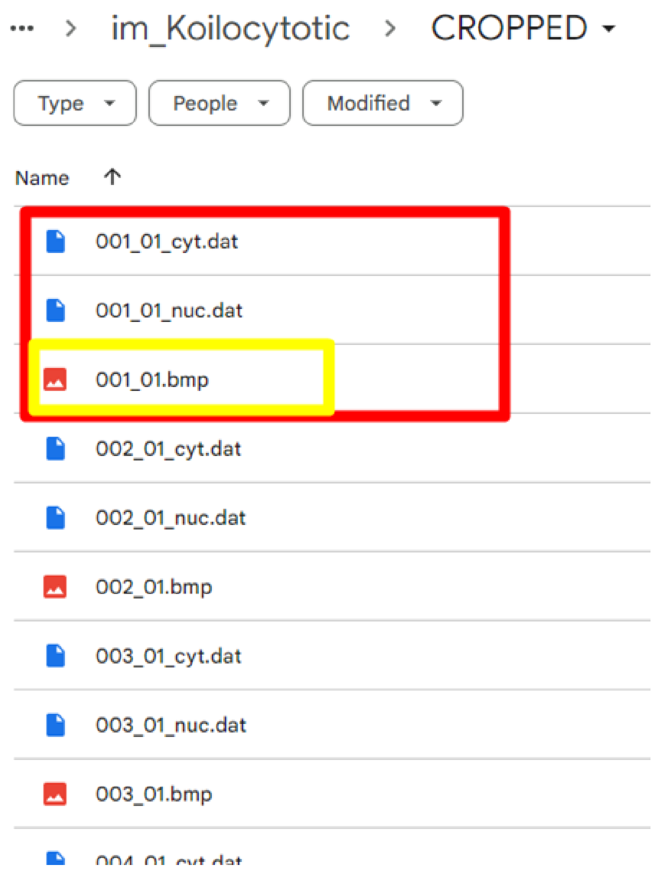
- 26 features calculated: intensity, texture, shape.
- Manual annotation of cytoplasm and nucleus boundaries by experts.

Classification Methods:

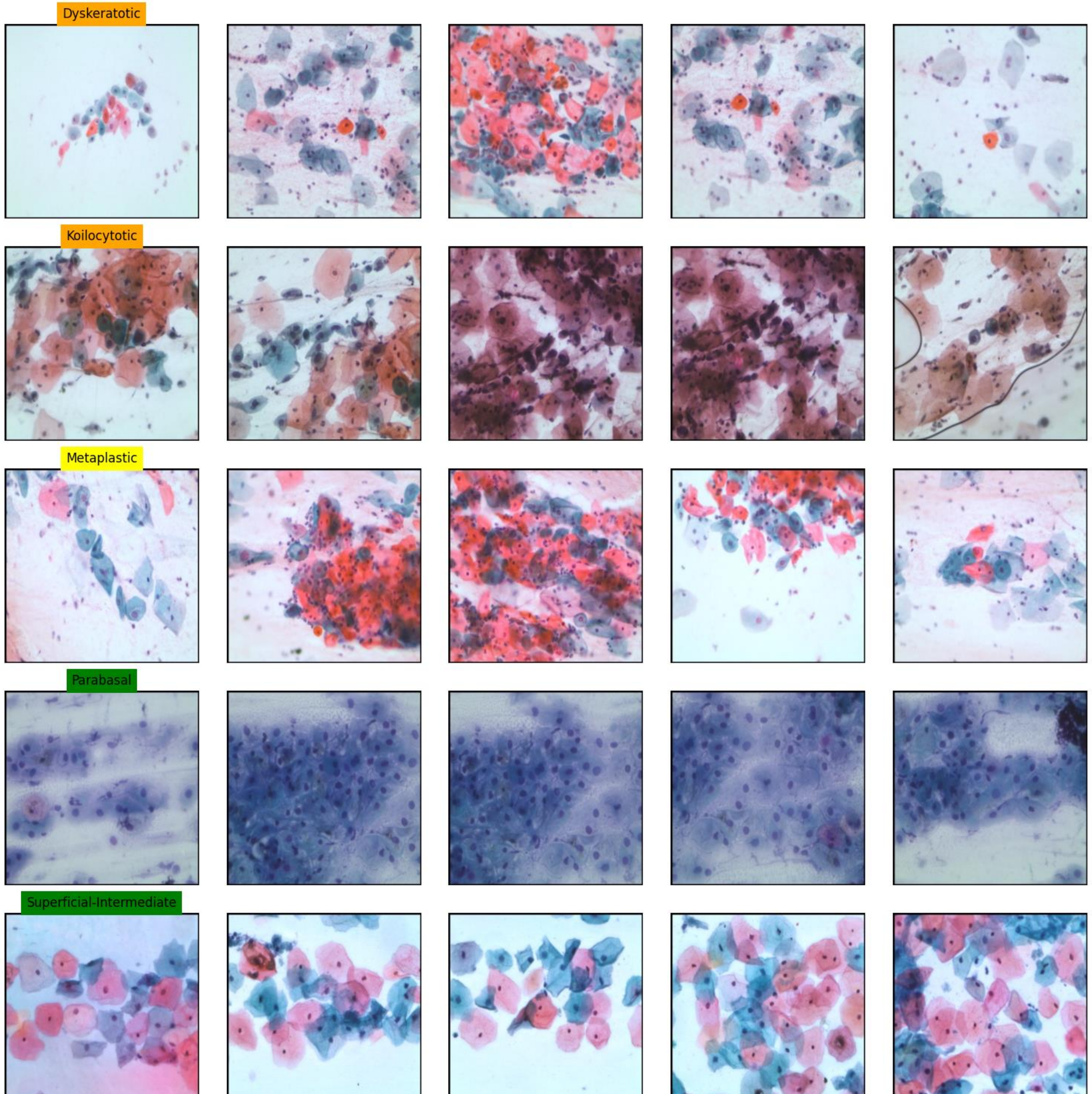
- Support Vector Machines (SVM) with RBF kernel. **- .dat**
- Multi-layer Perceptrons (MLP). **- .dat**
- Convolutional Neural Networks (CNN), adapted VGG-19 architecture. **- .bmp**

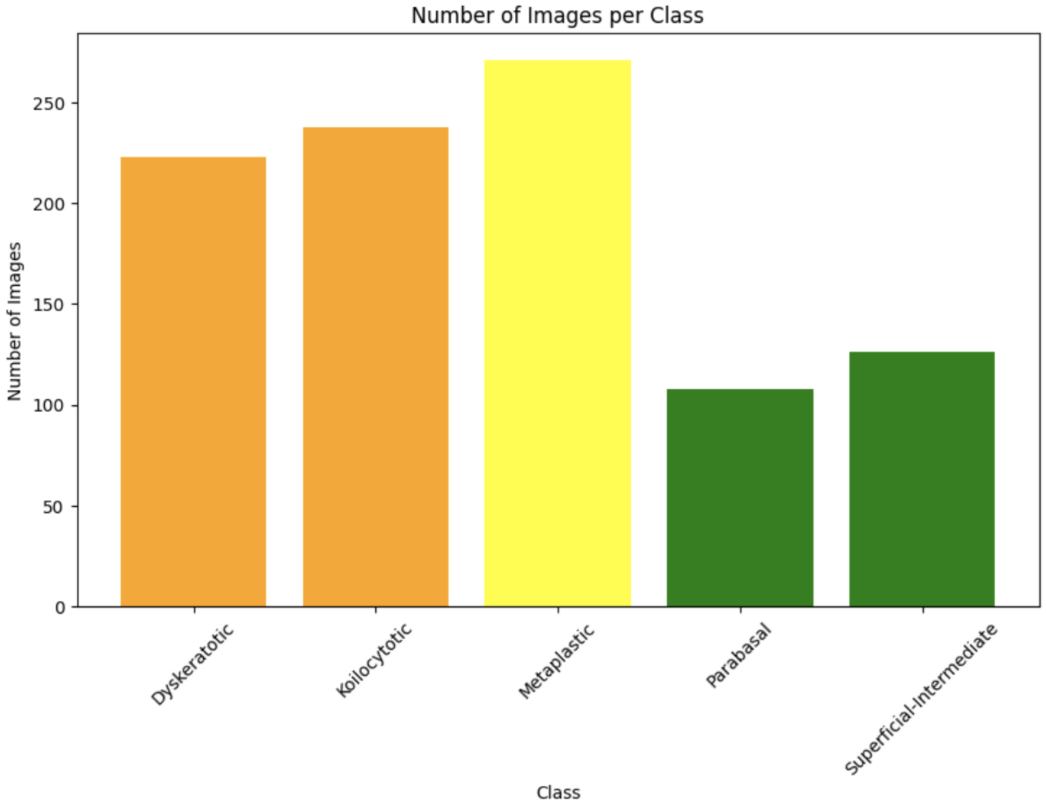
Performance Evaluation:

- Highest accuracy achieved by CNN.
- SVM and MLP showed effective performance with handcrafted features.



EDA - Clusters



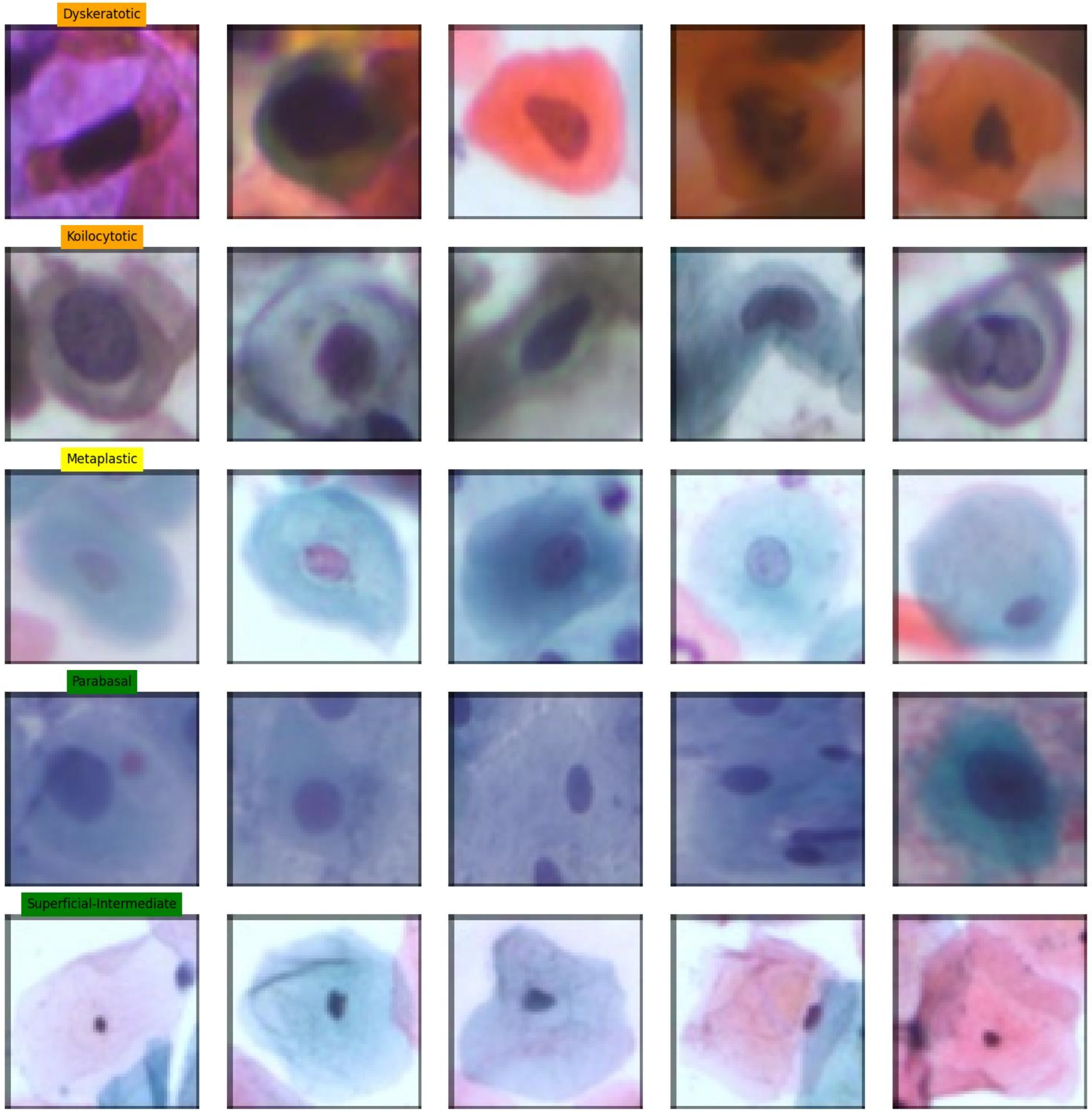


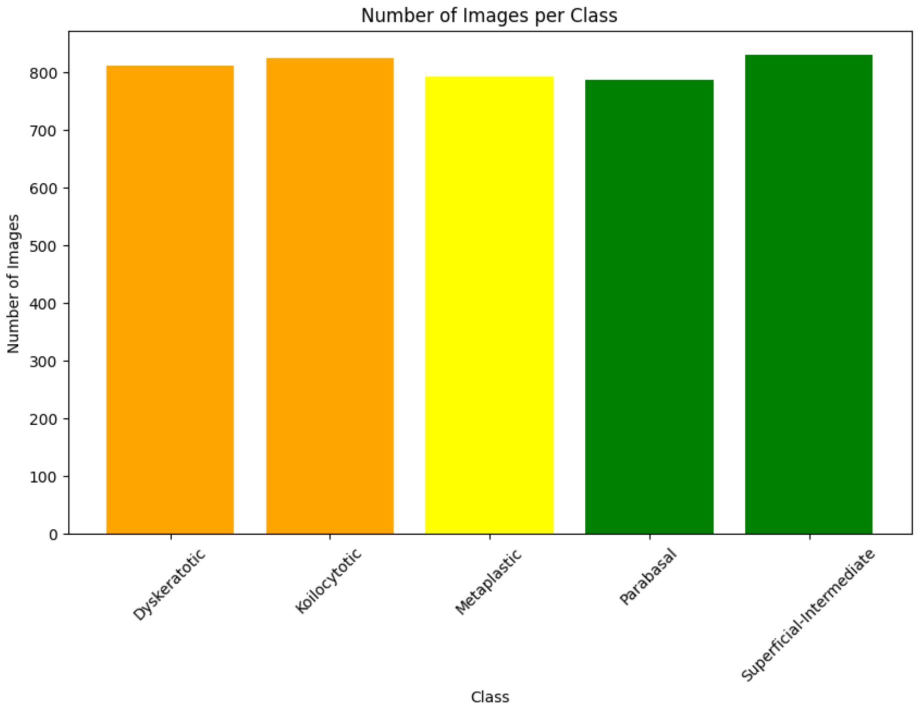
Cluster Image
Total number of images: 966

Number of images per class

- Dyskeratotic: 223 - **Concerning Cell**
- Koilocytotic: 238 - **Concerning Cell**
- Metaplastic: 271
- Parabasal: 108
- Superficial-Intermediate: 126

EDA - Cropped





CROPPED Image used
Total number of images: 4049

Number of images per class

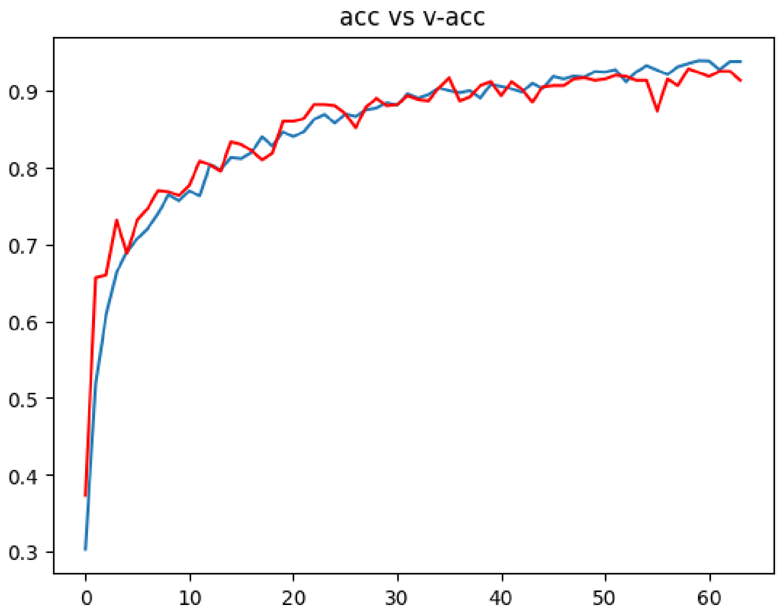
- Dyskeratotic: 813 - **Concerning Cell**
- Koilocytotic: 825 - **Concerning Cell**
- Metaplastic: 793
- Parabasal: 787
- Superficial-Intermediate: 831

Classification Model (1) - CNN

Model Architecture

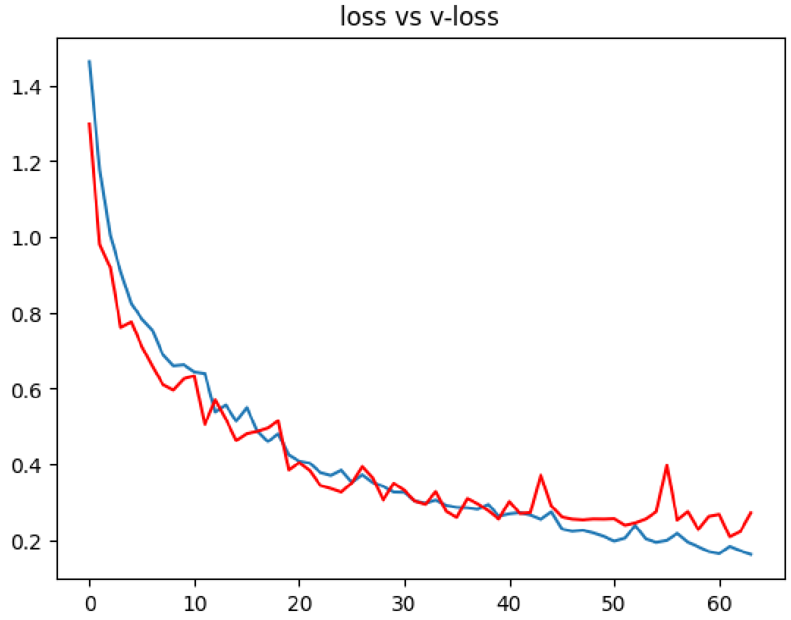
- **Input Layer:** Image input of shape (64, 64, 3).
- **Convolutional Layers:**
  - Three Conv2D layers with increasing filters (16, 32, 64) to extract features.
  - MaxPooling layers to reduce spatial dimensions.
- **Dropout Layers:** reduce overfitting.
- **Flatten Layer:** Flattens the 3D feature maps to 1D.
- **Dense Layers:**
  - One Dense layer with 64 units, activated by ReLU, followed by Dropout.
  - Final Dense layer with 5 units, activated by Softmax for classification into 5 categories.
- **Total Parameters:** 392,741

Accuracy Plot



- **Accuracy : 92.77%**
- Further refinement is possible with more advanced architectures.
- The performance is acceptable but could be enhanced with fine-tuning, or transfer learning approaches.

Loss Plot



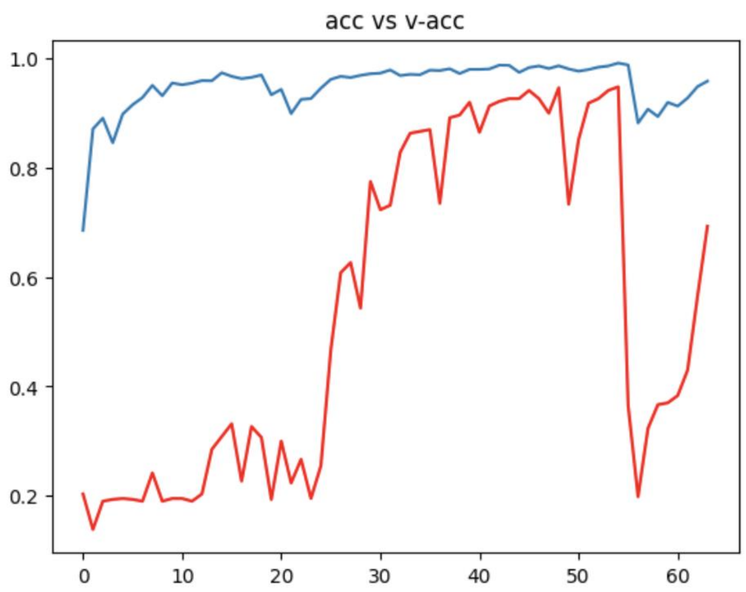
- **Validation loss: : 0.2717**
- The basic CNN shows limited overfitting, as indicated by the close match between training and validation accuracy and loss curves.
- Dropout layers were essential in mitigating overfitting.

Classification Model (2) - ResNet50

Model Architecture

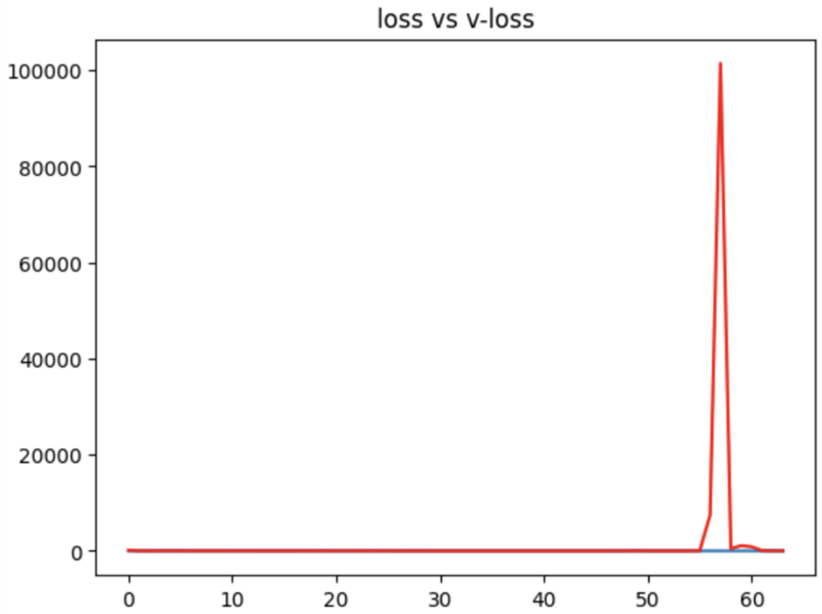
- **Base Model:** ResNet-50 pre-trained on ImageNet
- Custom Top Layers:
  - **Flatten Layer:** The output of ResNet-50 is flattened.
  - **Dense Layer:** A Dense layer with 1024 units and ReLU activation is added for high-level feature representation.
  - **Dropout Layer**
  - **Output Layer:** Final Dense layer with 5 units and Softmax activation for multi-class classification.

Accuracy Plot



- **Accuracy: 93.43%**
- Training accuracy (blue line) steadily increases and approaches 1.0, indicating that the model is fitting the training data well.
- However, the validation accuracy (red line) fluctuates significantly and drops drastically towards the end. This suggests that the model is overfitting the training data and failing to generalize to the validation set.

Loss Plot



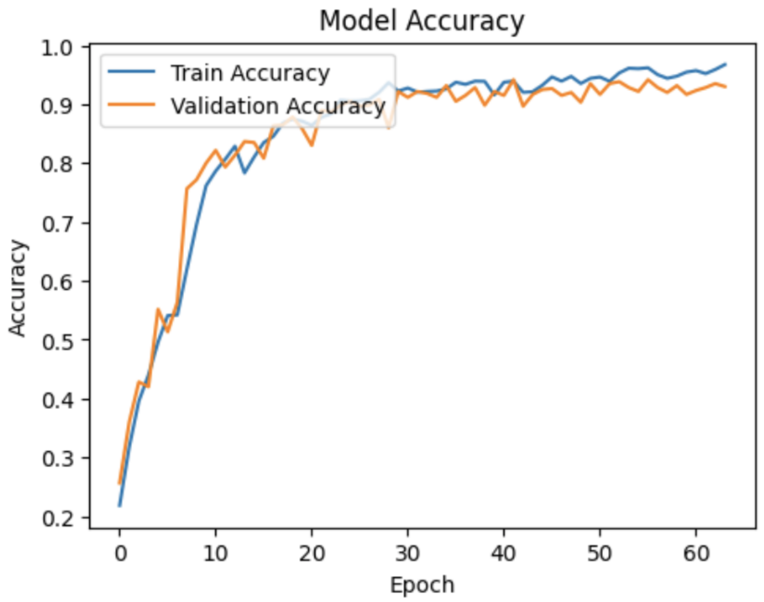
- **Validation loss : 6.0849**
- The loss chart further reinforces this observation, with the validation loss (red line) skyrocketing towards the end, while the training loss remains low. This is a clear indicator of overfitting.

Classification Model (3) - VGG 16

Model Architecture

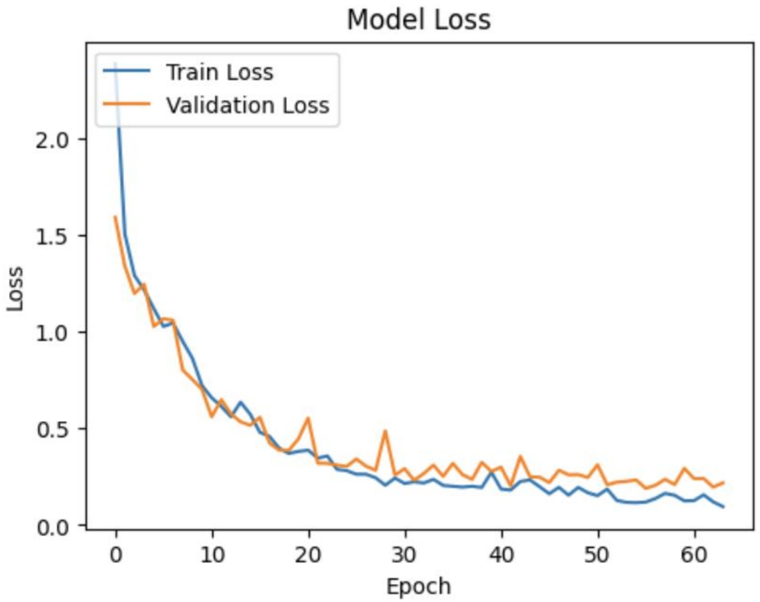
- **Base Model:** VGG16 pre-trained on ImageNet.
- **Custom Layers:** The VGG16 model was fine-tuned with custom layers, including:
    - Flatten layer to flatten the 3D outputs from the convolutional layers.
    - Dense layer with 1024 units and ReLU activation.
    - Output layer with 5 units (for the 5 classes) and softmax activation.
- **Optimizer:** Adam optimizer was used with categorical cross-entropy loss function.
- **Metrics:** The model was compiled with accuracy as the evaluation metric.

Accuracy Plot



- **Accuracy : 95.83%**
- The model achieved high accuracy on both the training and validation datasets, with accuracy nearing 0.9 after about 20 epochs.
- The validation accuracy closely followed the training accuracy, indicating that the model did not overfit significantly.

Loss Plot



- **Validation loss: 0.14**
- The training and validation loss both decreased significantly during the early epochs, stabilizing around 0.2 to 0.4 after 20 epochs, with some fluctuations.
- The validation loss also mirrored the training loss, suggesting good generalization.

Model Evaluation: Basic CNN and Transfer Learning

	Precision	Recall	f1-score	support
Dyskeratotic	0.89	0.9	0.89	315
Koilocytotic	0.88	0.87	0.87	237
Metaplastic	0.86	0.89	0.88	218
Parabasal	0.87	0.85	0.86	217
Superficial-Intermediate	0.91	0.89	0.9	240
accuracy			0.88	1227
macro avg	0.88	0.88	0.88	1227
weighted avg	0.88	0.88	0.88	1227

Transfer Learning 2: VGG16

- Deep network with 16 layers; pre-trained on ImageNet.
- Simplified by excluding the top classification layer and adding custom layers for the specific task.
- Reliable performance; strong generalization; utilizes pre-trained weights for enhanced feature extraction.

Test Accuracy: 95.83%
Test Loss: 0.14

Generate Synthetic Image with GenAI - Conditional GAN (cGAN)

Koilocytotic cells

Cyanophilic, very lightly stained and they are characterized by a large perinuclear cavity. The periphery of the cytoplasm is very dense stained. The nuclei are usually enlarged, eccentrically located, hyperchromatic and exhibit irregularity of the nuclear membrane contour.

Generator:

- **Inputs**:
  - **Noise Vector**: 100-dimensional random noise.
  - **Class Label**: Integer representing the target class.
- **Architecture**:
  - **Label Embedding**: Embeds the class label into a 100-dimensional vector.
  - **Element-wise Multiplication**: Combines noise with label embedding.
  - **Dense Layer**: Expands to 256 * 8 * 8 units, reshaped to 8x8x256.
  - **UpSampling and Conv2D Layers**: Gradually upscales to 64x64x3 with layers (256, 128, 64 filters).
  - **Output**: 64x64x3 image with "tanh" activation.

Discriminator:

- **Inputs**:
  - **Image**: 64x64x3 input image.
  - **Class Label**: Integer representing the target class.
- **Architecture**:
  - **Label Embedding**: Embeds the class label into a vector matching the flattened image size.
  - **Element-wise Multiplication**: Combines flattened image with label embedding.
  - **Dense Layers**: Three layers of 512 units with LeakyReLU and Dropout.
  - **Output**: Validity score with "sigmoid" activation.

cGAN Model:

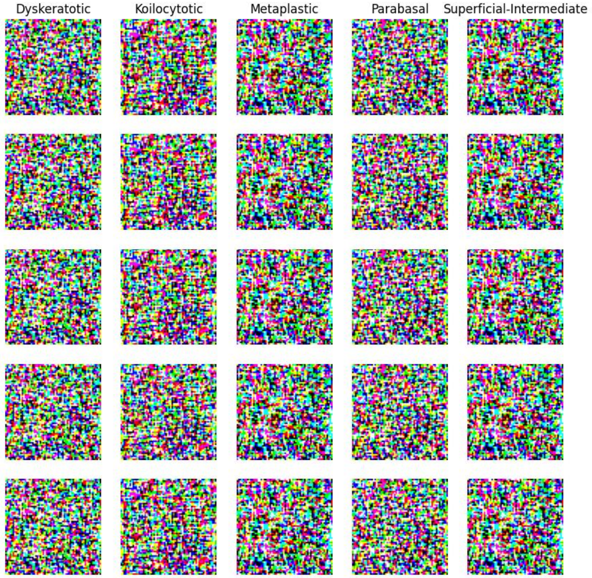
- **Training**:
  - **Generator** is trained to produce realistic images conditioned on class labels.
  - **Discriminator** classifies images as real or fake, conditioned on class labels.
  - **Optimization**: Both networks are optimized using binary cross-entropy loss.

Training Loop:

- **Alternating Updates**: Discriminator and Generator are updated alternately.
- **Performance Monitoring**: Losses for both networks are tracked, and generated images are periodically saved.

Visualization:

- **Generated Images**: Images are saved at intervals to monitor progress.
- **Loss Curves**: Discriminator and Generator losses are plotted over epochs to analyze training behavior.



Generate Synthetic Image with GenAI - Conditional Variational Autoencoder (CVAE)

Latent Dimension: **64** - Balances complexity and performance.

Input Dimensions:

- **Image Input**: 64x64x3 (RGB).
- **Label Input**: One-hot encoded vector (5 classes).

Encoder:

- **4 Conv2D Layers**: Filters 64, 128, 256, 512.
- **Batch Normalization**: After each Conv2D layer.
- **Dropout**: 0.3 rate for regularization.
- **Dense Layer**: 256 units, integrates features with class label.

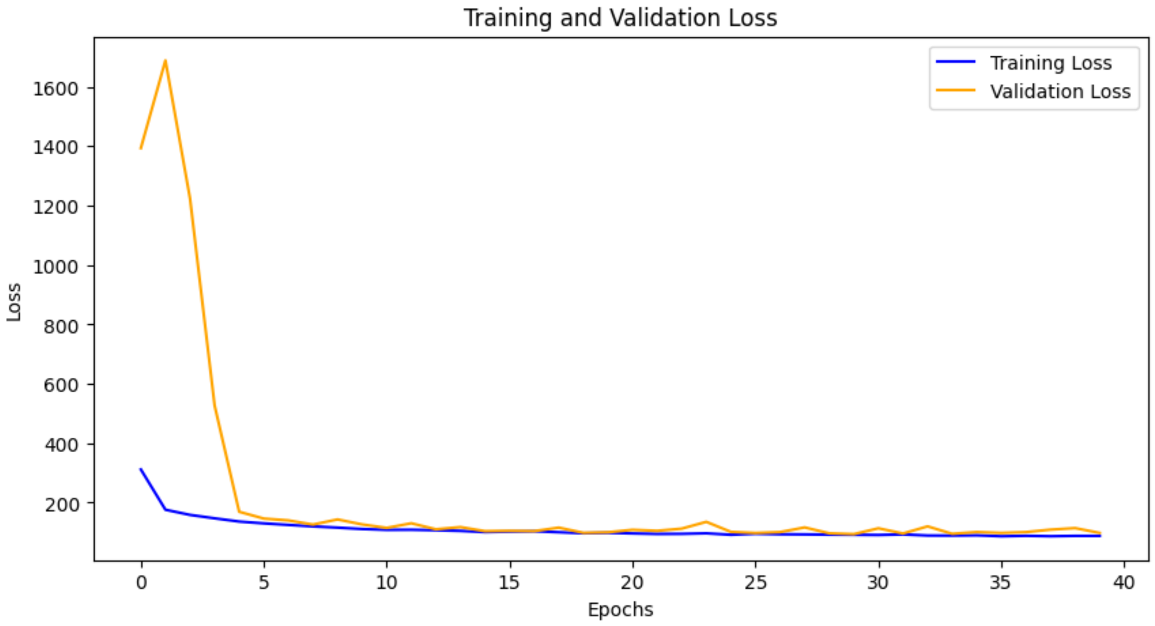
Latent Space:

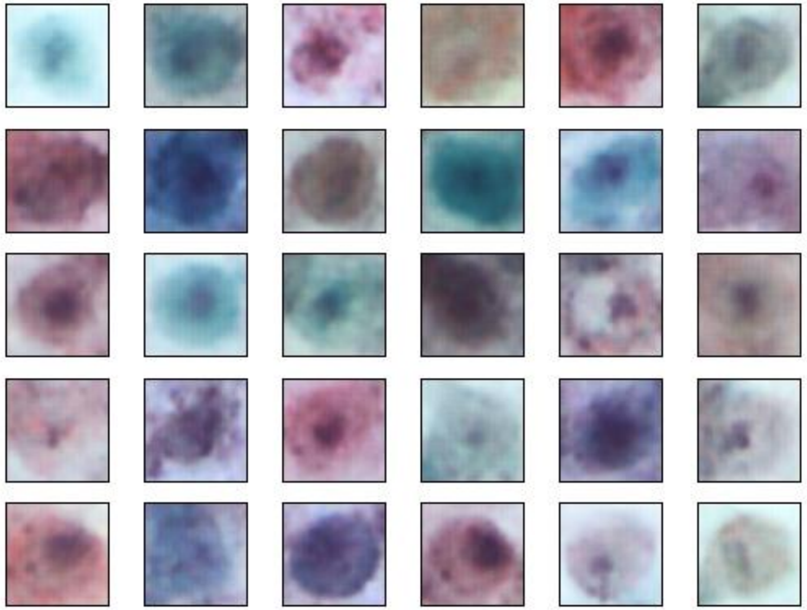
- **z_mean & z_log_var**: Captures the distribution.
- **Sampling Layer**: Reparameterization trick for gradient flow.

Decoder:

- **Dense Layer**: 16x16x512, reshaped for upsampling.
- **4 Conv2D Transpose Layers**: Filters 512, 256, 128, 64.
- **Output Layer**: Reconstructs 64x64x3 image.

Conditional Input: Integrates class labels into both encoder and decoder.

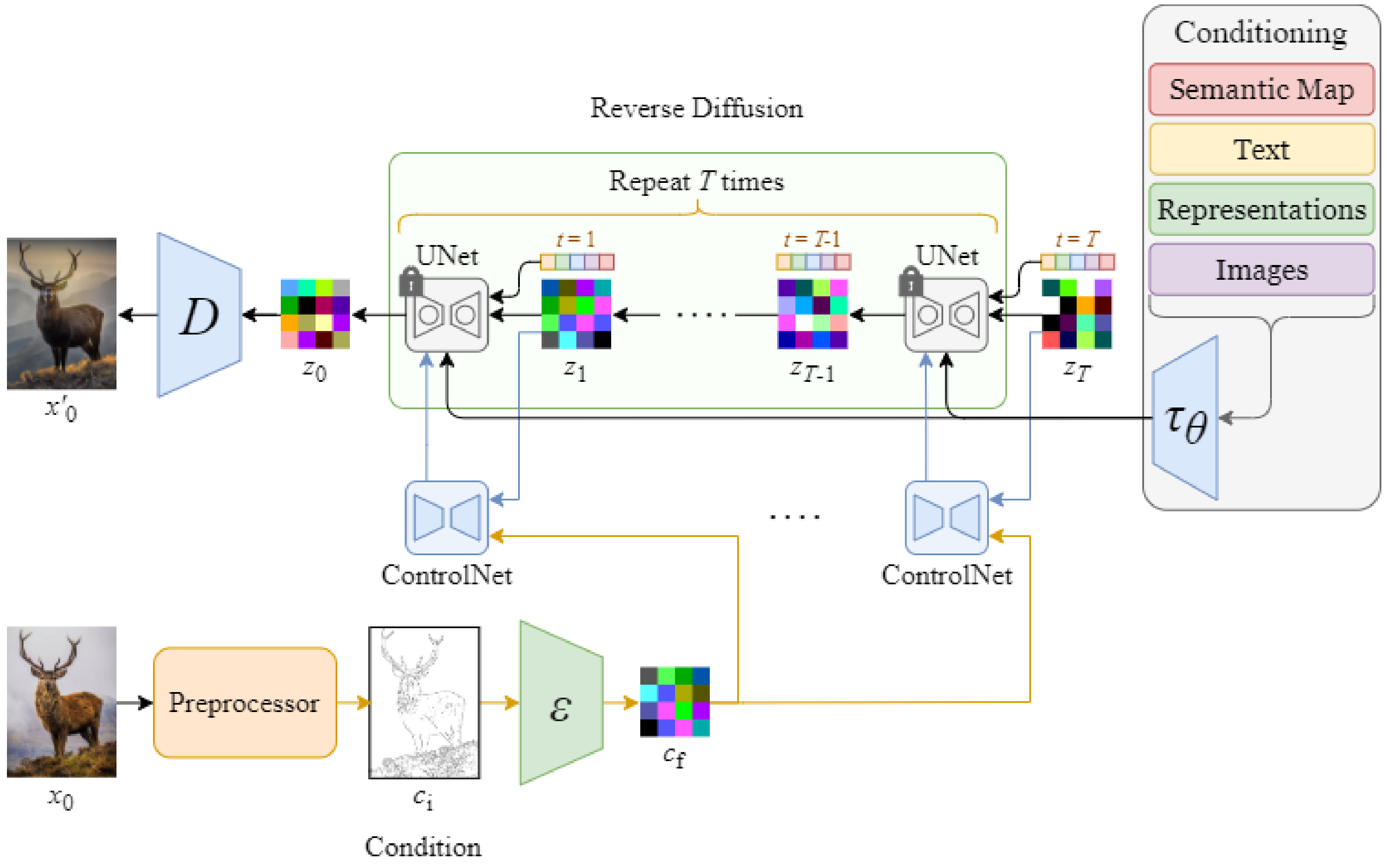




Loss Function:

- **Reconstruction Loss**: Measures pixel-wise differences between the input and output images.
- **Perceptual Loss**: Uses a pre-trained VGG16 model to compare high-level features between the input and output images, enhancing visual quality.
- **KL Divergence**: Ensures the latent space follows a standard normal distribution, promoting a well-structured latent space.

Pretrained Generative Models: Controlnet X-flux

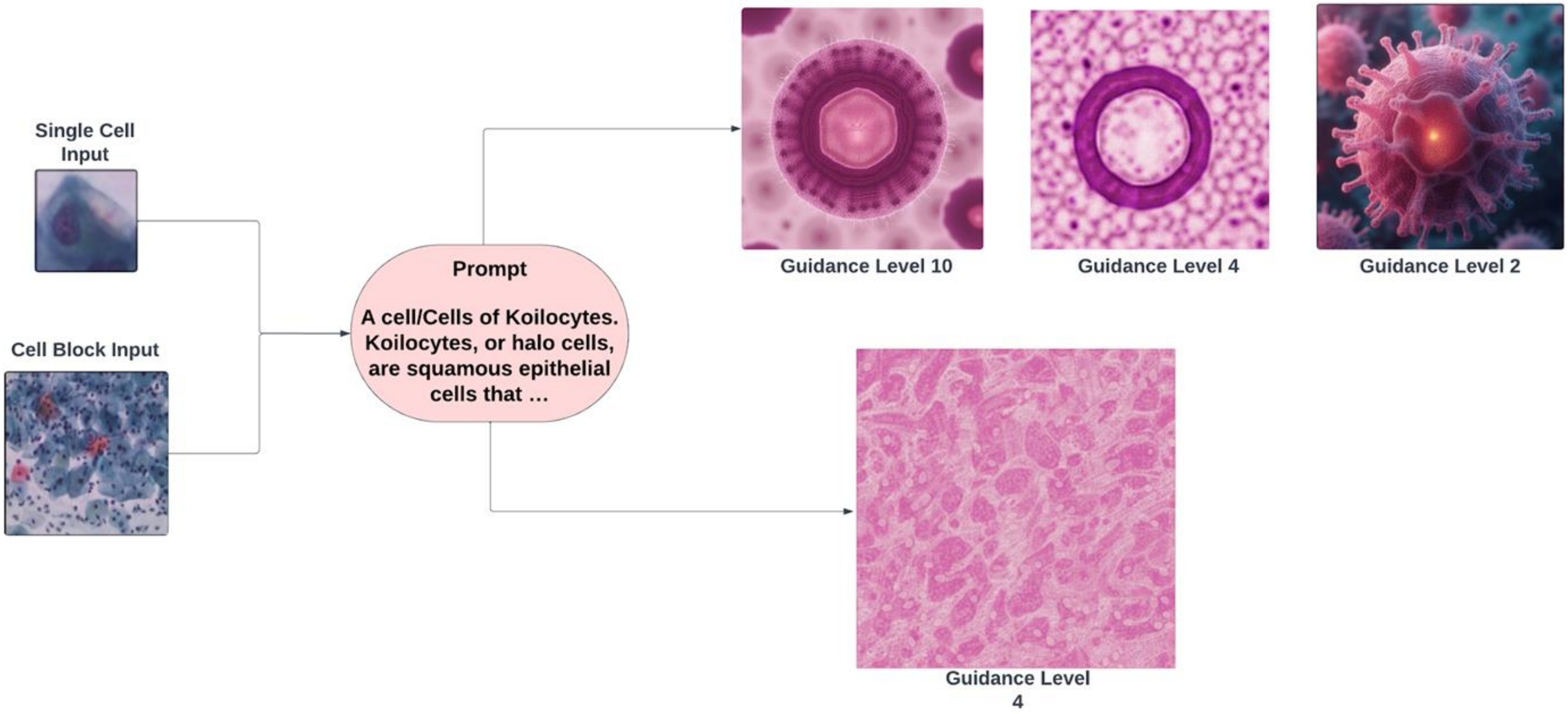


Model Parameters:

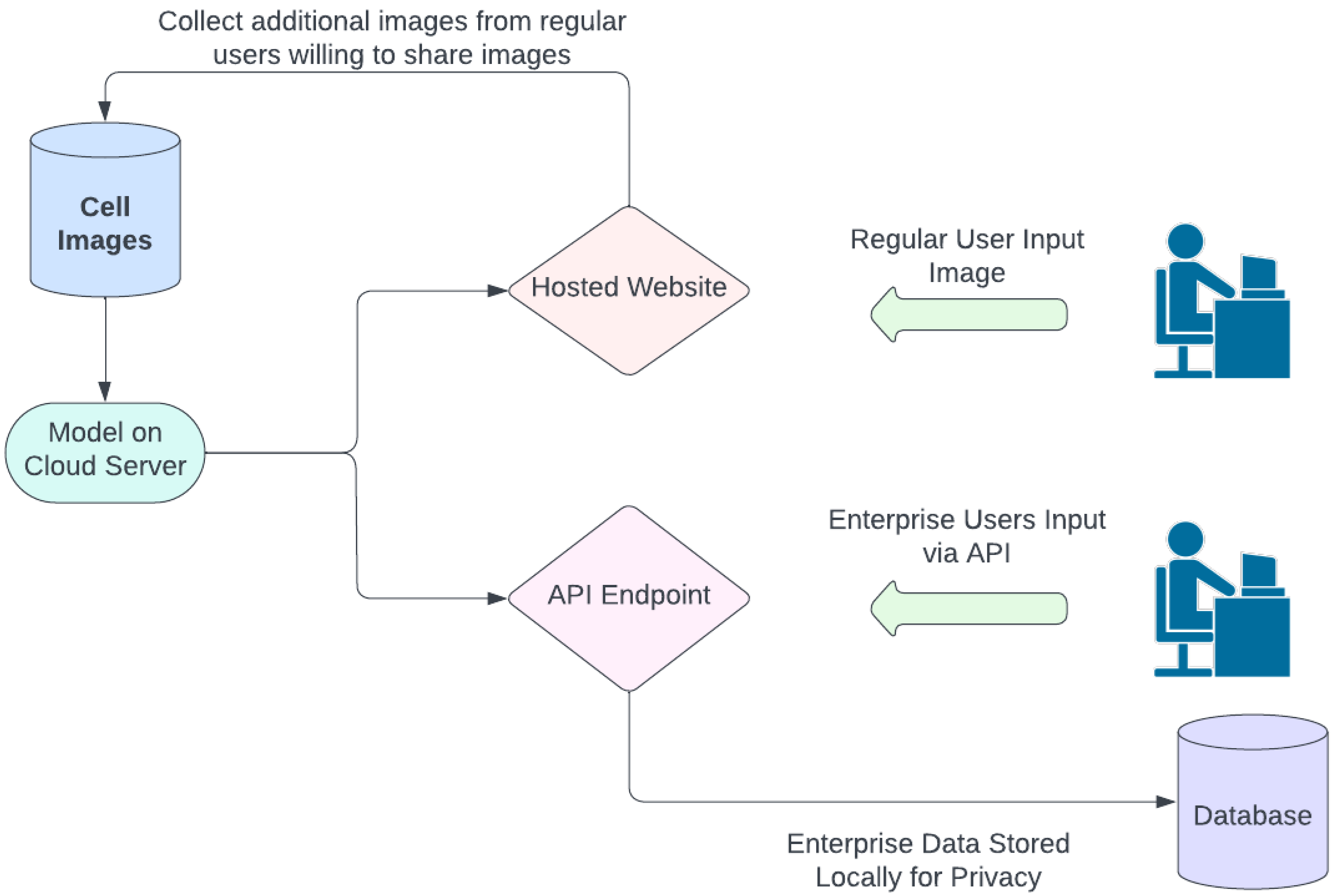
- Input image
  - Cell blocks or single cell
- Prompt
  - Descriptions of cancer cells
- Guidance
- Base Model
  - Controlnet or SORA

Source: https://medium.com/@steinsfu/stable-diffusion-controlnet-clearly-explained-f86092b62c89

Pretrained Generative Models: Controlnet X-flux



Model Operations - Privacy Matters



Model Operations - Refine Existing Model

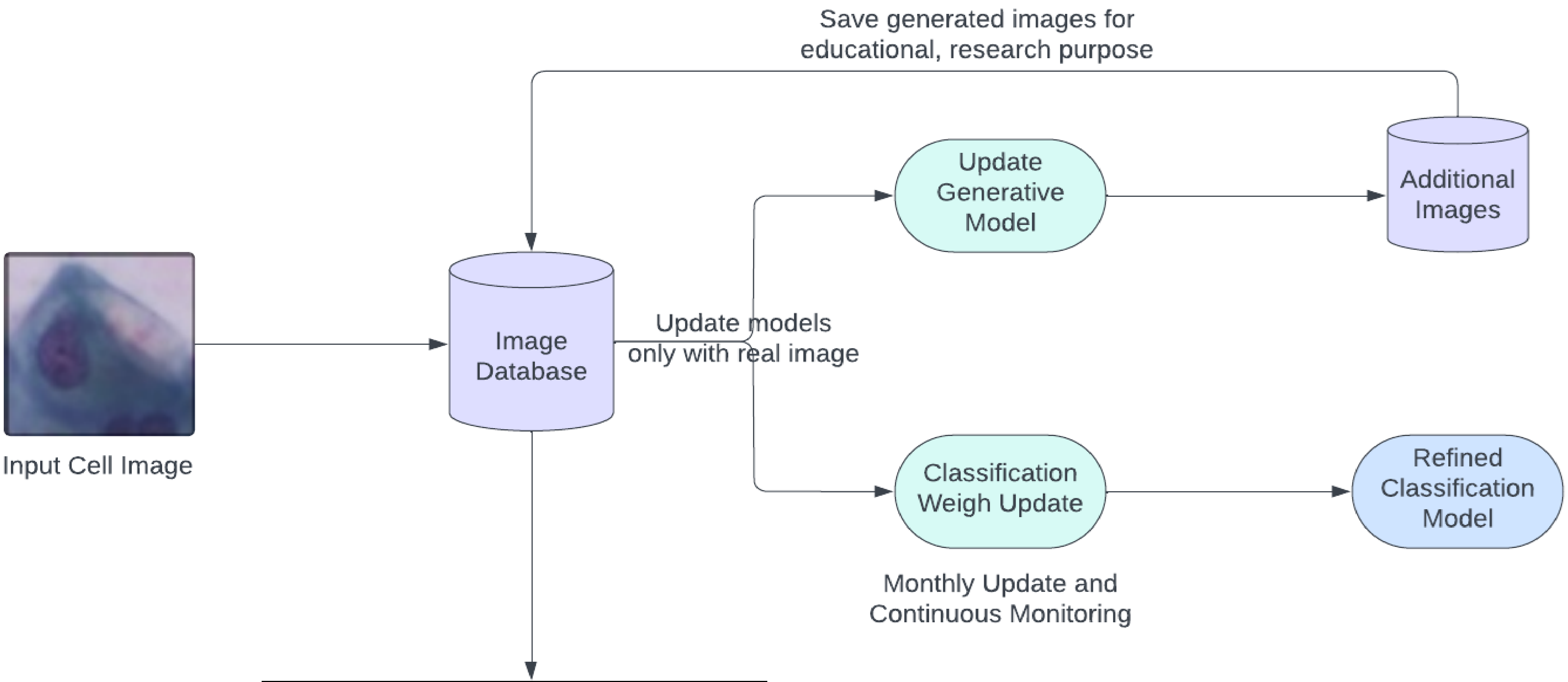


Image ID	Description	Cell Type	Generated
1	Sample from...	Koliocy	False
2	...	...	

Conclusion

Classification of Cancer cells

- Overall, we were able to achieve decent accuracy for classification task (96% accuracy), and we believe that with more data and refined conditions, we will be able to achieve better results
- Among the models we tried, transfer learning based on VGG and ResNet outperformed our customized CNN model

Generation of Cancer Cell Images

- Our customized VAE model performs better than the GAN model, but the generated image is still blurry, and we are hitting some training plateau
- While large-scale text-to-image generative model were able to generate cell images with decent resolution, the generated images are often “cartoonish”, which is likely because those models were not trained with cell images. We attempted to fine tune the models with our dataset, but the memory requirement is beyond our current compute capacity.
- More images with better resolution for training and fine tuning.
- Get HPV vaccines!

Reference:

X-flux Model:https://github.com/XLabs-AI/x-flux?tab=readme-ov-file

- Flux Model Architecture: https://github.com/black-forest-labs/flux
- https://huggingface.co/XLabs-AI/flux-RealismLora
- https://huggingface.co/XLabs-AI/flux-controlnet-canny

Data: https://www.kaggle.com/datasets/prahladmehandiratta/cervical-cancer-largest-dataset-sipakmed