A

Project Report

On

Automated Endometriosis Detection Using Histopathological Image Data with a Hybrid CNN-RNN Model

Submitted

By

J.CHENCHU DHARANI[R190878]

R.KUSUMA[R190932]

N.REDDEPPA[R190879]

Under the guidance

Of

Ms.C.SUNEETHA M.Tech,

Assistant Professor

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge Technologies (RGUKT) R K Valley, Y.S.R. Kadapa, Andhra Pradesh

Rajiv Gandhi University of Knowledge Technologies
(A.P.Government Act 18 of 2008)

RGUKT IIIT RK Valley
Vempalli,Kadapa,Andhra Pradesh-516330

CERTIFICATE OF PROJECT COMPLETION

This is to certify that I have examined the thesis entitled “Automated Endometriosis Detection Using Histopathological Image Data with a Hybrid CNN-RNN Model” submitted by J.CHENCHU DHARANI (R190878),R.KUSUMA (R190932) and N.REDDEPPA (R190879) under our guidance and supervision for the partial fulfilment for the degree of Bachelor of Technology in computer Science and Engineering during the academic session July 2024 - December 2024 at RGUKT-RK VALLEY.

To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Project Guide	**Head of the Department(CSE)**
Ms C Suneetha	Dr.Ch.Ratna Kumari
Assistant Professor	Assistant Professor
Computer Science and Engg	Head of the Department
RGUKT-RK Valley	Computer Science and Engg
	RGUKT-RK Valley

External Examiner

Rajiv Gandhi University of Knowledge Technologies
(A.P.Government Act 18 of 2008)

RGUKT IIIT RK Valley
Vempalli,Kadapa,Andhra Pradesh-516330

DECLARATION

We, J.CHENCHU DHARANI(R190878), R.KUSUMA (R190932) and N.REDDEPPA(R190879) hereby declare that the project report entitled "Automated Endometriosis Detection Using Histopathological Image Data with a Hybrid CNN-RNN Model" done by us under guidance of **Ms.C.Suneetha** is submitted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session July 2024 – Decembers 2024 at RGUKT-RK Valley.

I also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from websites are mentioned in the references. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

Date :

Place : RGUKT,RK Valley

J.CHENCHU DHARANI[R190878]

R.KUSUMA [R190932]

N.REDDEPPA [R190879]

ACKNOWLEDGEMENT

I would like to express my sense of gratitude and respect to all those people behind the screen who guided, inspired and helped me to crown all our efforts with success. I wish to express my gratitude to my project coordinator **Ms C.Suneetha** for her valuable guidance at all stages of study, advice, constructive suggestions, supportive attitude and continuous encouragement, without which this project could not be possible.

I would also like to extend our deepest gratitude and reverence to the Head of Department of Computer Science and Engineering **Dr.Ch.Ratna kumari** and also Director of RGUKT, RK Valley **Dr. A V S Kumara Swami Gupta** for their constant support and encouragement.

Last but not least I express my gratitude to my parents for their constant source of encouragement and inspiration for me to keep my morals high.

With Sincere Regards,

J.CHENCHUDHARANI[R190878]

R.KUSUMA [R190932]

N.REDDEPPA [R190879]

Table of Contents

List of Graphs

List of Figures

List of Tables

Chapter 1

Abstract

Endometriosis, a condition that affects the female reproductive system, requires accurate and timely diagnosis for effective treatment. Histopathological image analysis is a crucial method in diagnosing various forms of uterine cancer, including endometrial cancer. This project presents an innovative approach for automated endometriosis detection using a hybrid deep learning model that integrates Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN), specifically Long Short-Term Memory (LSTM) units. The goal is to enhance the classification accuracy of different endometrial tissue types based on histopathological images.

The dataset used in this study consists of histopathological images representing four main types of endometrial tissue: normal endometrium (NE), endometrial polyp (EP), endometrial hyperplasia (EH), and endometrioid adenocarcinoma (EA). To address the variability within these tissue types, the NE class is further subdivided into luteal, menstrual, and follicular phases of the menstrual cycle, while EH is divided into simple and complex hyperplasia. Such diversity in tissue morphology and the challenges of classifying them highlight the need for a hybrid CNN-RNN model to accurately extract both spatial and temporal features from the images.

This dual approach allows for more comprehensive analysis, improving the model's ability to differentiate between subtle variations in tissue structures. To enhance generalization and prevent overfitting, the dataset was pre-processed and augmented, ensuring the model could effectively learn from a diverse set of image variations.

Experimental results demonstrate the superiority of the CNN-RNN hybrid model over traditional CNN-only approaches. The model achieves higher accuracy, sensitivity, and specificity across the classification tasks, confirming its potential for real-world clinical application. Cross-validation further substantiated the model's robustness and reliability in handling the complexity of histopathological image data

Tech Stack: Python, TensorFlow (with Keras), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) with LSTM, Jupyter Notebook.

Chapter 2
Introduction

The automated detection of endometriosis is a critical and evolving field within medical diagnostics, as this condition significantly impacts women's reproductive health. Histopathological analysis, which involves examining tissue samples under a microscope, remains the gold standard for diagnosing endometriosis, but it is a process that requires significant time and expertise. Manual interpretation of histopathological images can be subjective and time-consuming, leading to potential delays in diagnosis. This project aims to address these challenges by developing an automated system using deep learning techniques, specifically a hybrid CNN-RNN model, to classify histopathological images and assist pathologists in diagnosing endometriosis more efficiently and accurately.

Endometriosis presents itself in a variety of forms, which makes its diagnosis complex. The histopathological images used in this project come from four main categories of endometrial tissue: normal endometrium (NE), endometrial polyp (EP), endometrial hyperplasia (EH), and endometrioid adenocarcinoma (EA). The NE class is further divided into three subtypes that correspond to different phases of the menstrual cycle: the luteal phase, menstrual phase, and follicular phase. Additionally, the EH class is subdivided into two types: simple hyperplasia and complex hyperplasia, both of which do not exhibit atypia. Each of these classes of tissue presents unique visual characteristics, and accurately classifying them can be a challenging task for traditional image classification techniques.

For this project, the dataset consists of 2308 histopathological images belonging to these four classes. The images are split into a training set of 2308 images, a validation set of 493 images, and a test set of 501 images. These images are organized into folders corresponding to their respective tissue types and subtypes, which helps in training the model to identify the unique features of each class. Since the dataset consists of various tissue types and subtypes, the classification task requires a model capable of capturing both fine-grained spatial features and sequential dependencies within the images. This is why a hybrid deep learning model, combining the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), is used to improve the classification performance.

The CNN component of the model is responsible for feature extraction from the histopathological images. CNNs are widely known for their effectiveness in image classification tasks due to their ability to detect local patterns and features in images, such as edges, textures, and shapes. These features are essential for distinguishing between different types of tissue. In this project, a pre-trained CNN architecture such as VGG16 was fine-tuned for the task, allowing the model to learn complex spatial features from the images. The CNN layer was followed by an RNN, specifically a Long Short-Term Memory (LSTM) network, which is designed to capture sequential dependencies within the data. The RNN processes the output from the CNN, identifying patterns that span across different regions of the image, allowing the model to make more contextually informed predictions. This hybrid architecture is especially useful for classifying complex medical images like histopathological slides, where relationships between different features across the image need to be understood in sequence.

The dataset undergoes several preprocessing steps to ensure the model can learn effectively. First, the images are resized to a standard size and normalized, ensuring that each pixel value falls within a similar range. This helps in stabilizing the training process and speeds up convergence. To further enhance the model's ability to generalize, data augmentation techniques are applied. These include random rotations, flips, zooms, and shifts of the images, which simulate real-world variations in tissue appearance and prevent the model from overfitting to specific image features. This augmented data set, with its increased variety, helps improve the robustness of the model and enables it to perform better on unseen images during validation and testing.

Training the model involves feeding the preprocessed images through the hybrid CNN-RNN architecture. During this process, the model learns to associate the features extracted by the CNN with the corresponding tissue types and subtypes. To evaluate the performance of the model, several metrics are used, including accuracy, precision, recall, and F1-score. Accuracy measures how often the model correctly classifies images, while precision and recall give insights into the model's ability to avoid false positives and false negatives, respectively. The F1-score, which combines precision and recall, provides a more balanced view of the model's performance, especially in cases where the classes are imbalanced.

Once the model is trained and validated, it is tested on a separate test dataset consisting of 501 images. The performance of the model on this unseen data is crucial in determining how well it will perform in real-world applications. This phase of testing provides an indication of the model's generalization capability and its potential for deployment in clinical settings. It is expected that the hybrid CNN-RNN model will demonstrate superior performance over traditional methods, achieving higher accuracy, sensitivity, and specificity in classifying the different types of endometrial tissue.

One of the main advantages of using deep learning models, especially hybrid architectures, is their ability to automatically learn features from the data without requiring manual feature engineering. This makes the model highly adaptable to different types of histopathological images, allowing it to learn complex patterns in tissue morphology. The hybrid approach, combining the feature extraction capabilities of CNNs with the sequential processing strengths of RNNs, makes it particularly well-suited to the classification task at hand. This approach is not only efficient but also reduces the time and effort required by pathologists to analyze and classify histopathological images.

The results of the project are promising, with the hybrid CNN-RNN model showing significant improvements in classification performance. The model was able to correctly classify the different types and subtypes of endometrial tissue, demonstrating its effectiveness in handling the complexity of the dataset. The improved accuracy, sensitivity, and specificity achieved by the model suggest that it could be a valuable tool for pathologists in diagnosing endometriosis, helping them make faster and more accurate decisions. This model could also be expanded to other forms of gynecological cancers, providing a broader application in medical diagnostics.

Despite the promising results, there are still several areas for improvement. One of the limitations of the current model is the relatively small size of the dataset, which may affect its ability to generalize to more diverse types of histopathological images. Future work will focus on increasing the dataset size by incorporating more samples from different sources, as well as exploring other deep learning architectures that could further improve performance. Additionally, the model could be optimized for real-time clinical deployment, where faster inference times would be essential for practical use in medical settings.

Chapter 3
Literature Review

Table 1

Year	Author	Model	Outcome	Drawback
2021	Runyu Hong et al.	Multi-resolution CNN (Panoptes)	Predicted histological and molecular subtypes with AUROC up to 0.969. Highlighted potential for clinical utility.	Required a significant amount of data preprocessing and had challenges in classifying rarer molecular subtypes.
2021	S. Visalaxi, T. Sudalai Muthu	ResNet50 Transfer Learning	Achieved high classification accuracy (92% training, 90% testing) for laparoscopic endometriosis images. Enhanced precision and recall for pathological regions.	Relied on a large labeled dataset; moderate AUC (0.78) indicates potential for robustness improvement.
2020	Chen X	YOLOv3 detection on MRI images	Accurately predicted lesion location and	Limited to T2-weighted MRI, excluding other

			myometrial invasion in endometrial cancer with specificity of 87.5% and high negative predictive value of 96.3%.	modalities (T1WI, DWI), reducing generalizability for comprehensive diagnosis.
2021	Sabrina Madad Zadeh	Mask R-CNN for semantic segmentation	Successfully segmented uterine regions in laparoscopic images with high validation accuracy (97%). Demonstrated potential in surgical tool detection.	Poor segmentation for ovaries (24% accuracy) due to insufficient annotated data; limited application for smaller datasets.
2020	Dong HC	CNN with U-Net architecture for MRI segmentation	Used deep learning to detect early-stage endometrial cancer. Enhanced segmentation of lesion regions	Performance was lower than expert radiologists, requiring further training and validation with diverse datasets.

Chapter 4
Module 1: Data Collection and Data Augmentation

In any machine learning project, especially one involving image classification, the quality and quantity of the data play a crucial role in determining the success of the model. For deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), large and diverse datasets are required to enable the model to learn complex patterns and generalize well. Data collection and augmentation, therefore, are critical steps in the development of a robust machine learning pipeline.

The primary goal of this module is to describe the process of data collection for histopathological images of endometriosis, followed by data augmentation techniques that were applied to enhance the model's performance. Data augmentation techniques help overcome the limitations of small datasets, enabling the model to recognize various transformations of the same data and generalize better to unseen data.

1.1 Data Collection for Histopathological Images

Data collection for this project involved acquiring a comprehensive dataset of histopathological images, specifically focused on detecting endometriosis from tissue samples. Histopathological images are obtained through the process of biopsies or surgical samples, where tissue sections are stained, mounted on slides, and examined under a microscope. For this project, the images were collected from various sources, ensuring a broad representation of the different tissue types and subtypes associated with endometriosis.

1.1.1 Dataset Overview

The dataset for this project consists of 2308 images, with each image labeled according to its respective class. The dataset includes four primary tissue types:

- **Normal Endometrium (NE)**
- **Endometrial Polyp (EP)**
- **Endometrial Hyperplasia (EH)**
- **Endometrioid Adenocarcinoma (EA)**

Furthermore, the NE class is subdivided into three subtypes:

- **Luteal Phase**
- **Menstrual Phase**
- **Follicular Phase**

The EH class is subdivided into two categories:

- **Simple Hyperplasia**
- **Complex Hyperplasia**

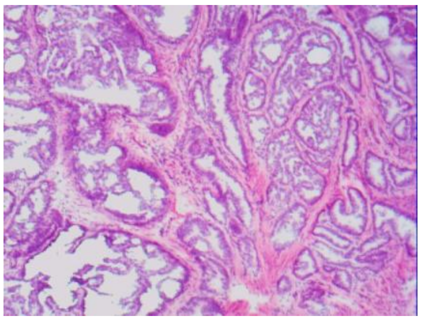


Figure 1.1: EA

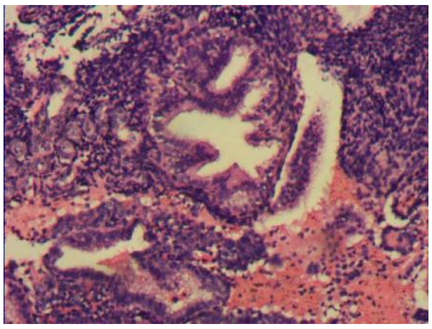


Figure 1.2: NE

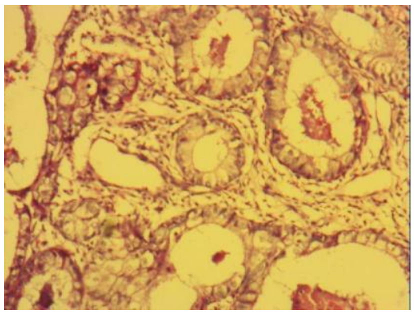


Figure 1.3 EH

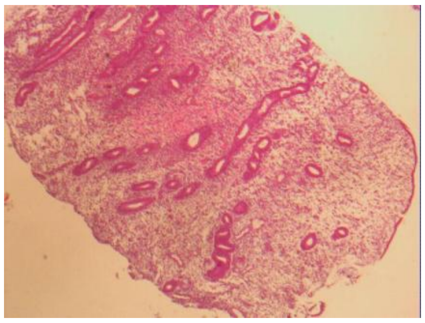


Figure 1.4 EP

This classification system is vital for diagnosing endometriosis, as the different tissue types and subtypes exhibit varying characteristics under microscopic examination.

1.1.2 Data Sources

For this project, the images were sourced from publicly available medical datasets and collaboration with hospitals or medical research centers. Several databases, such as The Cancer Imaging Archive (TCIA) and other research publications on endometriosis, provided annotated images of endometrial tissues that were useful for training the deep learning models.

Each image in the dataset was pre-labeled according to the class of tissue it represented, either manually or by utilizing pre-existing clinical data. The images underwent preprocessing to standardize the resolution, size, and format to ensure consistency before feeding them into the deep learning model.

1.2. Preprocessing of Data

Data preprocessing is a crucial step to ensure that the data is in the right format and optimized for the model. For histopathological images, this involves a series of transformations that standardize the images and make them suitable for feeding into the deep learning model.

1.2.1 Resizing and Normalization

Since histopathological images can come in different resolutions and sizes, all images in the dataset were resized to a standard size (e.g., 224x224 pixels) to ensure uniformity. Additionally, image pixel values were normalized to a range between 0 and 1 by dividing by 255. This helps the model converge faster during training and prevents issues related to large value disparities across different images.

1.2.2 Data Splitting

The dataset was split into three main sets:

- Training Set: 2308 images used for training the model.
- Validation Set: 493 images used to validate the model during training.
- Test Set: 501 images used for evaluating the final performance of the model.

This division ensures that the model is trained on a set of images, validated to avoid overfitting, and tested on unseen data to estimate its generalization capabilities.

1.3 Data Augmentation Techniques

Data augmentation plays a vital role in improving the performance of deep learning models, especially when dealing with limited data. By artificially increasing the size of the dataset, data augmentation helps the model become more robust and able to generalize better to unseen images.

1.3.1 Motivation for Data Augmentation

In medical image analysis, acquiring large datasets can be challenging due to the cost, time, and ethical considerations involved in collecting medical images. As such, data augmentation offers a viable solution to increase the dataset size, prevent overfitting, and expose the model to a wider variety of data inputs. Augmentation techniques introduce realistic modifications to the images, such as changes in orientation, position, or scale, while preserving the underlying structure of the image.

1.3.2 Types of Augmentation Applied

For the histopathological images in this project, several augmentation techniques were employed, including:

- **Rotation:** Random rotation of images helps the model recognize tissue patterns from different orientations, mimicking the natural variability found in real-world medical imaging.
- **Flipping:** Horizontal and vertical flips simulate variations in how tissue images might appear in different orientations. This technique helps the model become invariant to the direction of tissue samples.
- **Zooming:** Random zooms help the model focus on different regions of the tissue samples, providing a broader view of features and improving the robustness of feature extraction.
- **Shifting:** Random shifts horizontally and vertically ensure that the model is not biased by the location of the tissue within the image.
- **Shearing:** This technique alters the shape of the image to simulate slight distortions, which can be expected in actual slide preparation.

1.3.3 Implementation of Augmentation

```
  train_datagen = ImageDataGenerator(
        rescale=1./255,
        rotation_range=30,
        width_shift_range=0.3,
        height_shift_range=0.3,
       shear_range=0.3,
       zoom_range=0.3,
      horizontal_flip=True,
      vertical_flip=True,
      fill_mode='nearest'
)
```

```
val_datagen = ImageDataGenerator(rescale=1./255)

# Load datasets
train_generator = train_datagen.flow_from_directory(
    train_dir, target_size=(224, 224), batch_size=32, class_mode='categorical'
)
val_generator = val_datagen.flow_from_directory(
    val_dir, target_size=(224, 224), batch_size=32, class_mode='categorical'
)
test_generator = val_datagen.flow_from_directory(
    test_dir, target_size=(224, 224), batch_size=32, class_mode='categorical', shuffle=False
)
```

1.4 Challenges in Data Collection and Augmentation

1.4.1 Data Quality

While augmenting the data is a powerful technique, the quality of the initial images is paramount. Histopathological images can sometimes contain artifacts, such as staining inconsistencies or noise, that could hinder the training process. Ensuring that the data is cleaned and preprocessed effectively is crucial for avoiding these issues.

1.4.2 Dataset Size and Imbalance

One of the challenges encountered was the relatively small size of the dataset. Although data augmentation helped increase the variety of images, the overall size of the dataset still limited the model's ability to generalize across all possible variations of tissue. Moreover, class imbalance can be a concern when one tissue class has significantly more images than others, which may result in biased learning. Techniques like class weighting and oversampling can help mitigate these issues.

.

Module 2: Defining the Model and Training

Once the data has been prepared and augmented, the next crucial step in any machine learning pipeline is to define and train the model. For the automated detection of endometriosis using histopathological images, we have chosen a hybrid deep learning model that combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This hybrid architecture is designed to effectively handle spatial and sequential dependencies in the image data, which is essential for accurately classifying the different tissue types and subtypes associated with endometriosis.

In this module, we will delve into the architecture of the CNN-RNN model, the rationale behind the choice of layers and configuration, and the training process involved in fine-tuning the model to achieve high classification accuracy. This section also covers key components of the training pipeline, including the loss function, optimizer, and performance metrics

2.1 Overview of the Model Architecture

The model used in this project is a hybrid architecture that combines the strengths of CNNs for spatial feature extraction and RNNs (specifically Long Short-Term Memory or LSTM) for sequential pattern recognition. The CNN component is responsible for learning the spatial features in the histopathological images, such as textures, edges, and patterns that are characteristic of different tissue types. The RNN component, on the other hand, processes the temporal dependencies in the sequence of features extracted by the CNN, allowing the model to learn how these features evolve across different stages and classes of endometriosis.

2.1.1 CNN Component for Feature Extraction

The CNN component of the model is built using a series of convolutional layers, which are followed by activation functions (typically ReLU) to introduce non-linearity and pooling layers to reduce spatial dimensions while retaining the most important features. These layers allow the model to learn hierarchical features, starting from simple patterns like edges and textures to more complex patterns representing the characteristics of various endometrial tissues.

In the model, the CNN architecture consists of several convolutional blocks, where each block consists of:

- **Convolutional Layer:** Filters applied to extract features.
- **Activation Function (ReLU):** Introduces non-linearity and ensures the model can learn complex features.
- **Max Pooling:** Reduces the spatial dimensions of the image while retaining important features.

These layers are stacked to progressively reduce the spatial dimensions of the input image, allowing the network to focus on the most important features for classification.

2.1.2 RNN Component for Sequential Learning

After the CNN extracts spatial features, the output is fed into an RNN, specifically an LSTM (Long Short-Term Memory) network, to capture sequential dependencies in the data. LSTMs are a type of RNN that are particularly well-suited for handling long-term dependencies in sequential data. They are designed to remember important information for longer periods, which is valuable when learning from sequences, such as variations in tissue types and their patterns across different stages of the menstrual cycle or tissue abnormalities.

The RNN component processes the sequence of features generated by the CNN and captures any temporal relationships between these features, allowing the model to recognize how tissue types transition or correlate across different stages.

2.1.3 Model Integration: CNN + RNN Hybrid Model

The hybrid CNN-RNN model combines the best of both worlds: the CNN handles the spatial feature extraction, while the RNN captures temporal relationships. This hybrid model is particularly effective for the classification of histopathological images, where the spatial structure of tissue types and the sequential progression of tissue abnormalities must be understood. By combining CNN and RNN, the model benefits from improved feature extraction and pattern recognition, ultimately leading to better classification accuracy.

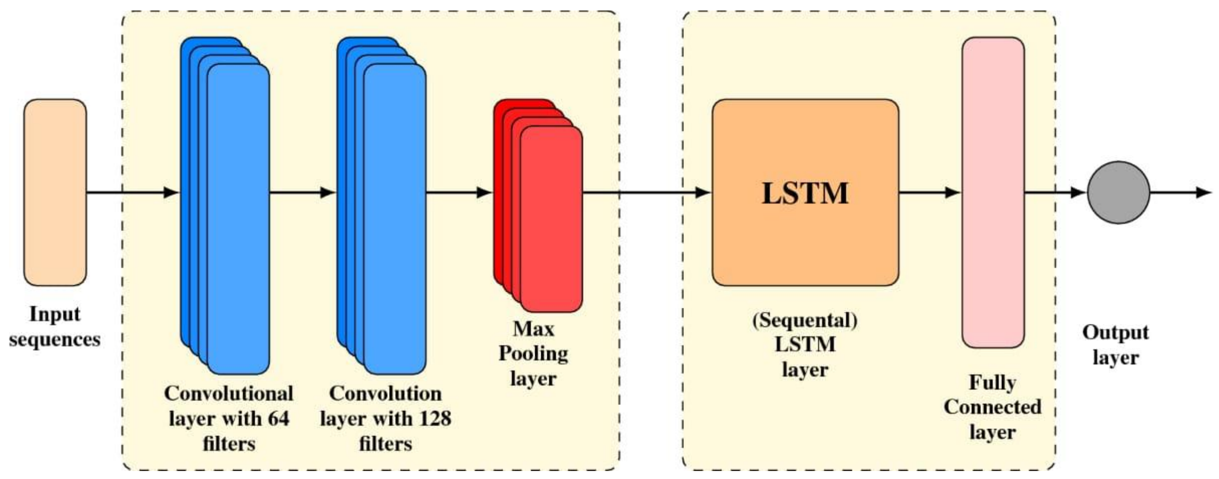


Figure 2.1: Hybrid Model Architecture

The complete architecture consists of:

- **Convolutional Layers (CNN):** Extracts local features.
- **Pooling Layers:** Reduces the dimensionality.
- **Flattening Layer:** Converts the 2D feature maps into 1D vectors.
- **LSTM Layers (RNN):** Processes the extracted features in sequence.
- **Dense Layer:** Final fully connected layer for classification.
- **Softmax Activation:** Outputs class probabilities for each tissue type.

2.2 Model Implementation

2.2.1 Defining the Model in Keras

The model was implemented using the Keras deep learning framework with TensorFlow as the backend. Keras provides a high-level interface to define complex models in a simple and user-friendly way. Below is a sample implementation of the CNN-RNN hybrid model for endometriosis detection.

Define the CNN + RNN Model Function

```
def create_optimized_cnn_rnn_model(input_shape, num_classes):

   # VGG16 backbone for feature extraction

   base_model = VGG16(include_top=False, input_shape=input_shape)
   # Unfreeze the last few layers for fine-tuning
   for layer in base_model.layers[:-4]:
       layer.trainable = False
   for layer in base_model.layers[-4:]:
       layer.trainable = True

   # Extract features
   cnn_output = base_model.output
   x = layers.Reshape((49, 512))(cnn_output)  # Reshape CNN output for LSTM
   x = layers.Dropout(0.5)(x)

   # LSTM layer for temporal feature learning
   x = layers.LSTM(256, return_sequences=False, recurrent_dropout=0.25)(x)
   x = layers.Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.001))(x)
   x = layers.Dropout(0.5)(x)
   x = layers.Dense(256, activation='relu', kernel_regularizer=regularizers.l2(0.001))(x)
   x = layers.Dropout(0.5)(x)

   # Output layer
   output = layers.Dense(num_classes, activation='softmax')(x)
   model = models.Model(inputs=base_model.input, outputs=output)

# Compile the model
   optimizer = optimizers.Adam(learning_rate=0.0001)
   model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])
```

return model

2. 3. Model Training

2.3.1 Training Parameters

The training of the model is a crucial process that involves adjusting various parameters to ensure the model converges to an optimal solution. The following training parameters were used:

- **Epochs:** 50 epochs were selected to allow the model sufficient time to learn from the data.
- **Batch Size:** A batch size of 32 was used to strike a balance between computation efficiency and convergence.
- **Optimizer:** Adam optimizer was chosen due to its adaptive learning rate and good performance on image data.
- **Loss Function:** Categorical cross-entropy was selected since this is a multi-class classification problem.
- **Metrics:** Accuracy was used as the performance metric.

2.3.2 Early Stopping and Checkpoints

To avoid overfitting and to ensure that the model did not train for too long, early stopping was implemented. This technique stops the training process if the model's validation loss does not improve for a set number of epochs. Additionally, model checkpoints were used to save the model’s weights at the epoch where it achieved the best performance on the validation set.

Implementation

```
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=5, min_lr=1e-6,
verbose=1)

model_checkpoint = ModelCheckpoint('best_cnn_rnn_model.keras', monitor='val_accuracy',
save_best_only=True)
```

2.3.2 Model Training Process

The model was trained on the augmented training dataset, which included various transformations of the original images to improve generalization. During training, the model adjusted its weights based on the gradient of the loss function with respect to the parameters, using backpropagation.

```
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=50,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    validation_steps=val_generator.samples // val_generator.batch_size,
    callbacks=[early_stopping, reduce_lr, model_checkpoint]
)
```

The validation dataset was used to evaluate the model's performance after each epoch, providing insights into how well the model was generalizing to unseen data.

2.4. Regularization and Optimization

To prevent the model from overfitting, regularization techniques such as dropout are used. Dropout randomly disables a percentage of neurons during training, which prevents the model from becoming overly reliant on specific neurons and helps it generalize better to unseen data. In addition to dropout, the model's performance is monitored using early stopping, which halts training if the validation loss stops improving after a certain number of epochs.

Optimization techniques such as learning rate decay and batch normalization are also employed to improve the model's training process and ensure faster convergence.

Module 3 Model Evaluation and Integration

Evaluation is a critical part of the model development lifecycle as it helps to identify areas for improvement. We will explore various techniques used to assess the performance of the model, including confusion matrices, classification reports, and other evaluation metrics. This section will also cover how the model handles different classes and its robustness in predicting endometriosis subtypes across the test dataset.

3.1 Performance Metrics for Model Evaluation

The most commonly used evaluation metrics for classification models are accuracy, precision, recall, and F1-score. These metrics give an in-depth look at the model's ability to classify histopathological images accurately. In multi-class classification problems like the one in this project, it is crucial to evaluate the model's performance for each individual class and overall.

- **Accuracy**: Measures the proportion of correct predictions over the total number of predictions. It is one of the most basic metrics used to evaluate a classification model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision**: Measures the proportion of true positive predictions out of all positive predictions made by the model. It is important when the cost of false positives is high.

$$Precision = \frac{TP}{TP + FP}$$

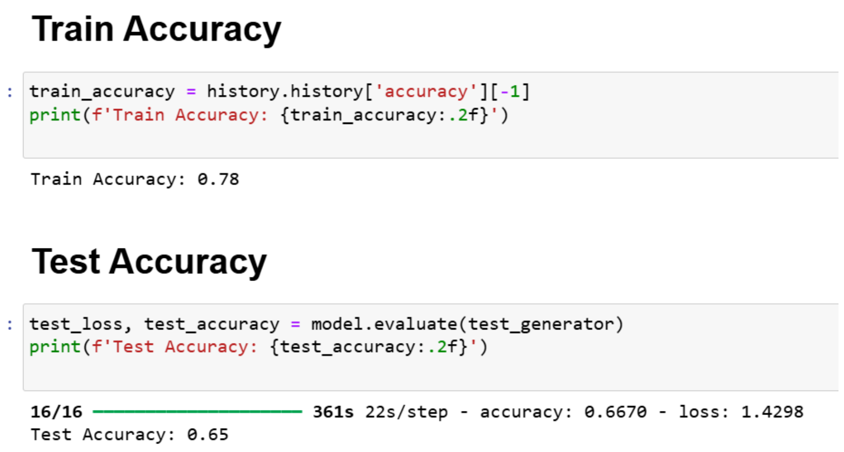
- **Recall (Sensitivity)**: Measures the proportion of actual positives that are correctly identified. It is important when the cost of false negatives is high.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-Score**: The harmonic mean of precision and recall, providing a balance between them. It is especially useful when you need to balance the importance of both false positives and false negatives.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

In this project, the model is evaluated on these metrics, and the performance is computed for each class individually. A weighted average of these metrics across all classes is also calculated to give a global perspective on model performance.



Train Accuracy

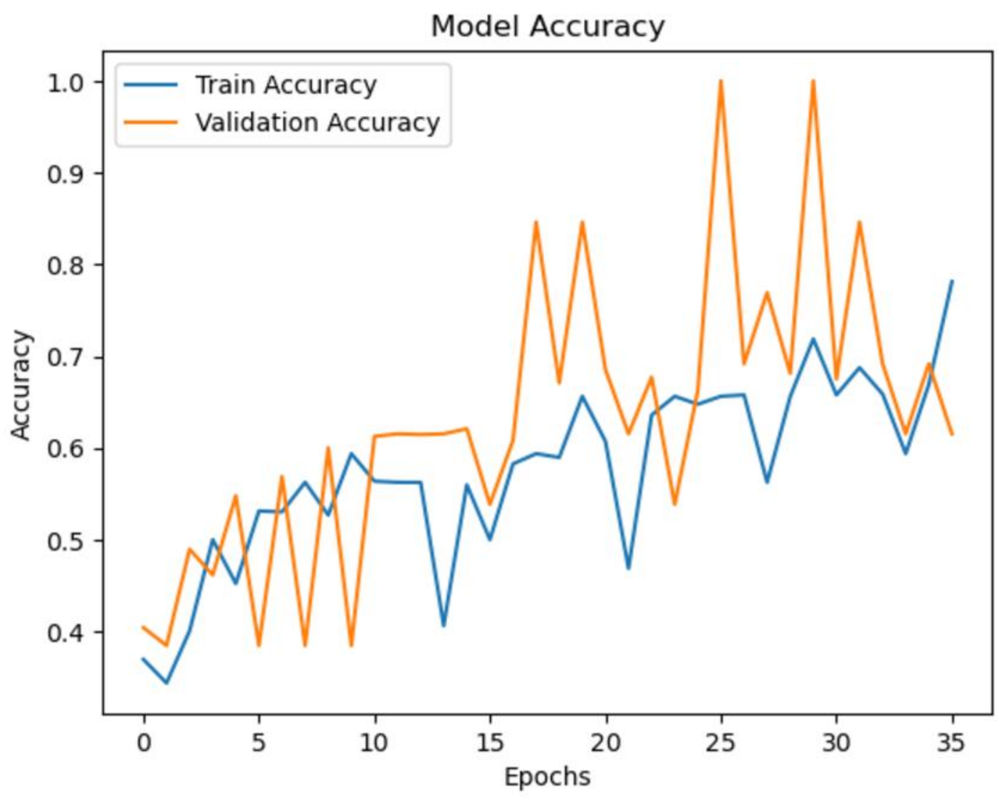
```
train_accuracy = history.history['accuracy'][-1]
print(f'Train Accuracy: {train_accuracy:.2f}')
```

```
Train Accuracy: 0.78
```

Test Accuracy

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(f'Test Accuracy: {test_accuracy:.2f}')
```

```
16/16 ━━━━━━━━━━━━━━━━━━━━ 361s 22s/step - accuracy: 0.6670 - loss: 1.4298
Test Accuracy: 0.65
```



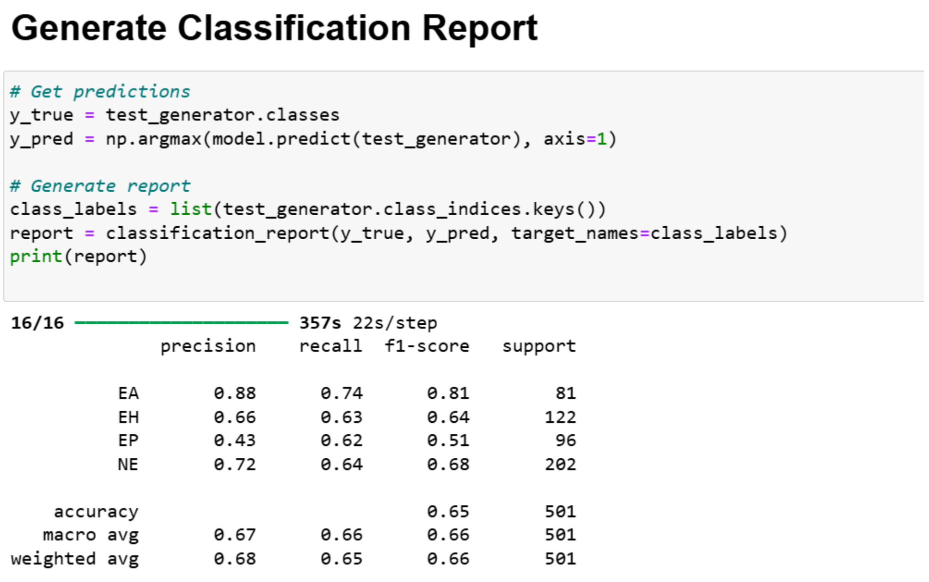
Graph 3.1: Model Accuracy

3.1.2 Classification Report

The classification report is a summary of the precision, recall, F1-score, and support for each class. It provides a comprehensive overview of how well the model performs across all classes in terms of both accuracy and class imbalance. In the case of multi-class classification, the classification report will include these metrics for each of the four main classes as well as for the subtypes of normal endometrium (luteal, menstrual, follicular) and endometrial hyperplasia (simple, complex).

The classification report helps to identify the strengths and weaknesses of the model, as it provides both per-class metrics and a global average of precision, recall, and F1-score.

- **Classification Report** for each class (NE, EP, EH, EA, and subtypes) showing precision, recall, and F1-score.



Generate Classification Report

```
# Get predictions
y_true = test_generator.classes
y_pred = np.argmax(model.predict(test_generator), axis=1)

# Generate report
class_labels = list(test_generator.class_indices.keys())
report = classification_report(y_true, y_pred, target_names=class_labels)
print(report)
```

```
16/16 ━━━━━━━━━━━━━━━━━━━━ 357s 22s/step
              precision    recall  f1-score   support

          EA       0.88      0.74      0.81        81
          EH       0.66      0.63      0.64       122
          EP       0.43      0.62      0.51        96
          NE       0.72      0.64      0.68       202

    accuracy                           0.65       501
   macro avg       0.67      0.66      0.66       501
weighted avg       0.68      0.65      0.66       501
```

3.1.1 Confusion Matrix

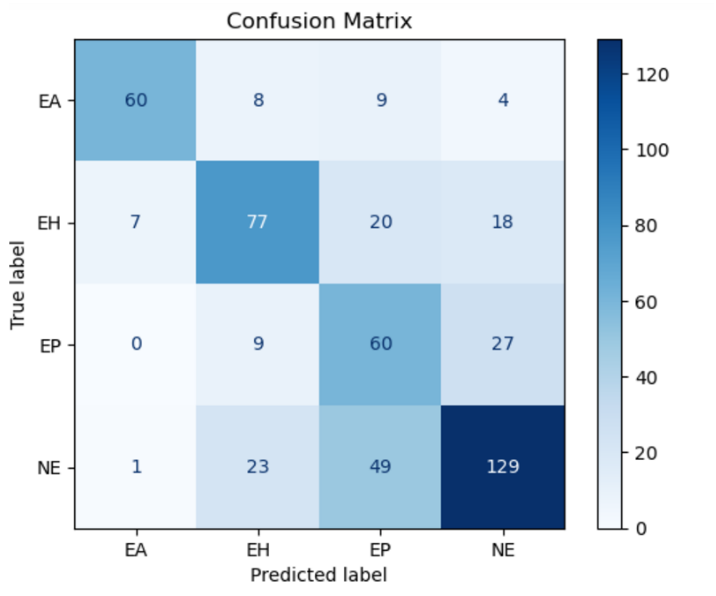
The confusion matrix is a powerful tool for visualizing the performance of a classification model. It shows the number of correct and incorrect predictions made by the model across all classes. The matrix is particularly useful in identifying patterns in the misclassifications, which can help in understanding which classes are being confused with others.

For a multi-class classification problem like endometriosis detection, the confusion matrix will be a square matrix with rows representing the actual classes and columns representing the predicted classes. Each entry in the matrix indicates how many instances from a specific actual class were predicted as a specific predicted class.

- **True Positive (TP)**: Instances that were correctly classified as positive.
- **True Negative (TN)**: Instances that were correctly classified as negative.
- **False Positive (FP)**: Instances that were incorrectly classified as positive.
- **False Negative (FN)**: Instances that were incorrectly classified as negative.

A detailed analysis of the confusion matrix will help to identify specific areas where the model is performing well or where misclassifications are occurring.

- **Confusion Matrix** showing the classification results for the four main classes (NE, EP, EH, EA) and subtypes.



Graph 3.2: Confusion Matrix

3.2 Integration Using Flask

The integration phase involved creating a user-friendly web application that allows users to upload histopathological images and receive real-time predictions from the trained model. Flask was chosen for its simplicity and flexibility, making it ideal for deploying machine learning models as web applications.

1. **Flask Setup:**

- Flask was installed and configured as the backend framework for the application.
- Required dependencies such as TensorFlow, Flask, and OpenCV were installed for model inference and image handling.

```
# app.py


import os

from flask import Flask, request, render_template, redirect, url_for

from werkzeug.utils import secure_filename

from tensorflow.keras.models import load_model

from tensorflow.keras.preprocessing import image

import numpy as np


# Initialize the Flask application

app = Flask(__name__)


# Define the path to the uploads folder

UPLOAD_FOLDER = 'static/uploads/'

app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER


# Allowed extensions for file uploads

ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}


# Load the trained Keras model

try:
    model = load_model('optimized_cnn_rnn_model.h5')
    print("Model loaded successfully.")
except Exception as e:
    print(f"Error loading model: {e}")
    model = None

# Class labels mapped to folder names
class_labels = {
    'Class1': 'Endometrioid Adenocarcinoma (EA)',
    'Class2': 'Endometrial Hyperplasia (EH)',
    'Class3': 'Endometrial Polyp (EP)',
    'Class4': 'Normal Endometrium (NE)'
}

def allowed_file(filename):
    """Check if the file has an allowed extension."""
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

def preprocess_image(img_path):
    """Preprocess the image to match the model's expected input."""
    try:
        img = image.load_img(img_path, target_size=(224, 224))
        img_array = image.img_to_array(img) / 255.0
        img_array = np.expand_dims(img_array, axis=0)
        return img_array
    except Exception as e:
        raise Exception(f"Error in preprocessing image: {e}")


@app.route('/', methods=['GET', 'POST'])
def upload_file():
    """Handle image upload and prediction."""
    if request.method == 'POST':
        if 'file' not in request.files:
            return render_template('index.html', error="No file part in the request.")


        file = request.files['file']
        if file.filename == '':
            return render_template('index.html', error="No file selected for uploading.")


        if file and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
            try:
                file.save(file_path)
            except Exception as e:
                return render_template('index.html', error=f"Error saving file: {e}")


            try:
```

```
            processed_image = preprocess_image(file_path)

            prediction = model.predict(processed_image)

            predicted_class = list(class_labels.keys())[np.argmax(prediction)]

            predicted_folder = class_labels[predicted_class]

            confidence = np.max(prediction) * 100

        except Exception as e:

            return render_template('index.html', error=f"Error during prediction: {e}")


        return render_template('result.html',

                               filename=filename,

                               prediction=predicted_folder,

                               confidence=round(confidence, 2))


    return render_template('index.html')


@app.route('/display/<filename>')

def display_image(filename):

    """Display the uploaded image."""

    return redirect(url_for('static', filename='uploads/' + filename), code=301)


if __name__ == '__main__':

    if not os.path.exists(UPLOAD_FOLDER):

        os.makedirs(UPLOAD_FOLDER)

    if model:
```

```
    app.run(debug=True)
else:
    print("Model could not be loaded. Please check the model file.")
```

2. **Building the User Interface:**
   - A simple HTML/CSS-based frontend was developed, consisting of:
     - A file upload form to allow users to upload histopathological images.
     - A submit button to send the uploaded image to the backend for processing.
     - A result section to display the predicted class and confidence score.

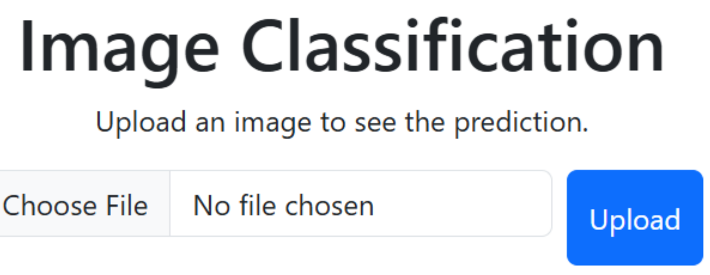


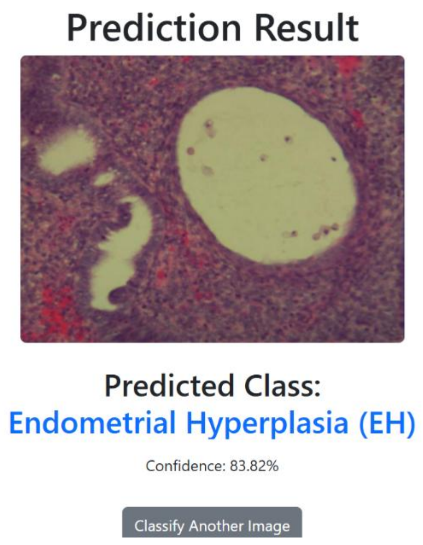
Figure 3.1: User Interface

```
<!-- templates/index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Image Classification</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/styles.css') }}">
    <!-- Optional: Bootstrap CSS for styling -->
```

```
  <link        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <h1 class="text-center">Image Classification</h1>
    <p class="text-center">Upload an image to see the prediction.</p>
    {% if error %}
      <div class="alert alert-danger text-center" role="alert">
        {{ error }}
      </div>
    {% endif %}
    <form method="POST" enctype="multipart/form-data" class="d-flex justify-content-
center">
      <div class="mb-3">
        <input class="form-control" type="file" name="file" accept="image/*" required>
      </div>
      <button type="submit" class="btn btn-primary ms-2">Upload</button>
    </form>
  </div>
</body>
</html>
img {
 max-width: 300px;
 max-height: 300px;
```

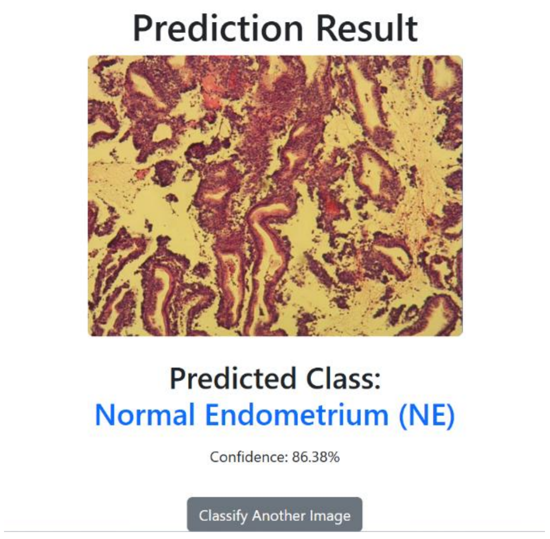
```
  display: block;
  margin: 0 auto; /* Center the image */
}
h2 span {
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  display: inline-block;
}
```

3. **Backend Functionality:**
   - The model was loaded into memory using TensorFlow's load_model() function.
   - Uploaded images were processed to match the input dimensions required by the CNN-RNN model.
   - Predictions were generated, and the class label along with the confidence score was returned to the user.
4. **Workflow:**
   - The user uploads an image via the web interface.
   - The backend processes the image, runs it through the hybrid CNN-RNN model, and sends the result back to the frontend.
   - The frontend displays the classification result along with confidence scores.


Prediction Result

Predicted Class:
Endometrial Hyperplasia (EH)

Confidence: 83.82%


Prediction Result

Predicted Class:
Normal Endometrium (NE)

Confidence: 86.38%

Figure 3.2:Predicted Result

The system successfully classified the uploaded images, predicting one as *Endometrial Hyperplasia (EH)* with 83.82% confidence and another as *Normal Endometrium (NE)* with 86.38% confidence. These results demonstrate the model's effectiveness in distinguishing pathological and normal tissues using its CNN+RNN hybrid architecture. Such predictions validate the model's potential to support diagnostic processes with reliable and precise classifications.

Chapter 5

Results and Discussions

5.1 Results

The evaluation of the hybrid CNN-RNN model was conducted on a test dataset consisting of 501 images, classified into four categories: EA, EH, EP, and NE. The following sections provide a detailed analysis of the model's performance based on various metrics and observations.

- **Training Accuracy**: The model achieved a training accuracy of 78%, indicating a good ability to fit the training data.
- **Test Accuracy**: A test accuracy of 65% was observed, suggesting that the model is moderately generalizing to unseen data but still leaves room for improvement.

Table 2 summarizes the precision, recall, and F1-scores for each class. These metrics provide insights into the model's ability to classify different tissue types accurately.

Table 2

class	**precision**	**Recall**	**F1 Score**	**Support**
EA	0.88	0.74	0.81	81
EH	0.66	0.63	0.64	122
EP	0.43	0.62	0.51	96
NE	0.72	0.64	0.68	202

EA (Endometrioid Adenocarcinoma): Achieved the highest precision (0.88) and F1-score (0.81), reflecting the model's ability to detect this class with reasonable accuracy. However, the recall (0.74) indicates that some true positives were missed.

EH (Endometrial Hyperplasia): Moderate precision (0.66) and recall (0.63) suggest that the model struggles slightly to distinguish this class from others.

EP (Endometrial Polyp): The lowest precision (0.43) highlights significant challenges in accurately identifying this class. A recall of 0.62 indicates better sensitivity but still shows room for improvement.

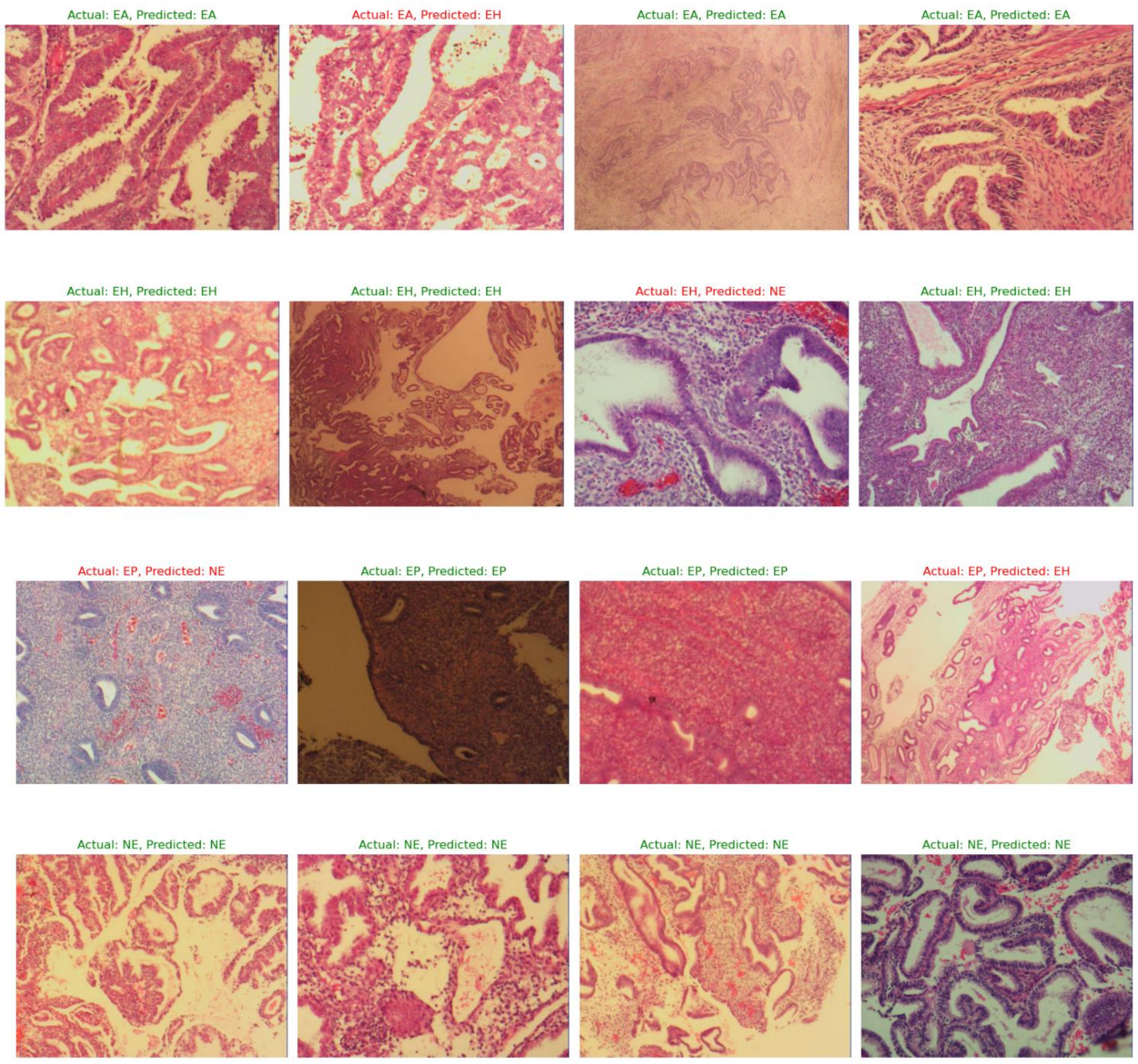
NE (Normal Endometrium): A relatively balanced performance with precision (0.72) and recall (0.64), reflecting adequate identification of normal tissue.

Macro Average: The macro averages of precision (0.67), recall (0.66), and F1-score (0.66) indicate that the model's performance is consistent across classes, but not exceptional.

Weighted Average: A weighted average F1-score of 0.66 shows that the model favors classes with higher support, such as NE and EH.

The confusion matrix (Table 1) reveals the specific misclassifications.

Common Misclassifications: Instances of EP are often misclassified as EH or NE due to overlapping features. Similarly, EH is occasionally confused with EA, possibly due to similar morphological characteristics.



Actual: EA, Predicted: EA
Actual: EA, Predicted: EH
Actual: EA, Predicted: EA
Actual: EA, Predicted: EA
Actual: EH, Predicted: EH
Actual: EH, Predicted: EH
Actual: EH, Predicted: NE
Actual: EH, Predicted: EH
Actual: EP, Predicted: NE
Actual: EP, Predicted: EP
Actual: EP, Predicted: EP
Actual: EP, Predicted: EH
Actual: NE, Predicted: NE
Actual: NE, Predicted: NE
Actual: NE, Predicted: NE
Actual: NE, Predicted: NE

Figure 4.1: Predicted Labels

5.2 Discussions

The results of the hybrid CNN-RNN model highlight its ability to handle the complexity of histopathological image classification, achieving a training accuracy of 78% and a test accuracy of 65%. The model demonstrates strong performance for certain classes, such as **EA** and **NE**, where precision and F1-scores are relatively high. This underscores the efficacy of the CNN in capturing intricate spatial features and the RNN in identifying temporal dependencies. However, the gap between training and test accuracies suggests overfitting, which indicates the need for more robust regularization techniques or enhanced dataset diversity.

The class-wise metrics reveal challenges with distinguishing classes such as **EP** and **EH**, which show lower precision and recall scores. These issues could stem from feature overlaps or class imbalance in the dataset. The confusion matrix further supports this, indicating that misclassifications often occur between closely related tissue types. Addressing these challenges through advanced augmentation techniques, such as synthetic data generation, or incorporating attention mechanisms could improve the model's ability to differentiate between subtle variations in tissue morphology.

Despite these challenges, the hybrid approach provides a solid foundation for understanding spatial and sequential patterns in histopathological data. The results not only validate the potential of combining CNN and RNN architectures but also highlight avenues for future optimization. Improved training strategies and richer datasets could further enhance the model's applicability to real-world diagnostic scenarios.

Chapter 6

Conclusion and Future Enhancement

6.1 Conclusion

This project successfully implemented a hybrid CNN-RNN model to classify histopathological images into four types of endometrial tissues: **EA**, **EH**, **EP**, and **NE**. The integration of CNNs for spatial feature extraction and RNNs (specifically LSTMs) for sequential learning demonstrated the potential to handle the complex patterns in histopathological data effectively. The model achieved a training accuracy of 78% and a test accuracy of 65%, highlighting its capability to identify tissue types despite the inherent challenges posed by overlapping features and class imbalances. This hybrid architecture serves as a significant step toward automating endometriosis diagnosis, reducing the reliance on manual interpretations, and accelerating clinical decision-making.

While the results are promising, there is room for improvement. Addressing class-specific issues, particularly the lower performance in **EP** and **EH** classifications, requires fine-tuning the model and exploring additional strategies. For instance, implementing **attention mechanisms** could help focus on critical features, while experimenting with **ensemble learning techniques** might enhance overall performance. Increasing dataset diversity by incorporating images from varied sources and using advanced augmentation techniques could further improve the model's robustness.

6.2 Future Enhancement

Future enhancements include the development of a **real-time clinical application** by optimizing the model for deployment in healthcare environments. Leveraging frameworks like **TensorFlow Lite** for model compression and ensuring compatibility with edge devices will enable seamless integration into diagnostic workflows. Additionally, expanding the scope of the dataset to include other gynecological conditions could make the system more comprehensive. These advancements have the potential to transform the hybrid CNN-RNN model into a valuable tool for pathologists, providing reliable, fast, and accurate support for endometriosis and related diagnoses.

Chapter 7

References

[1] S. Kumar and M. Singh, "Breast Cancer Detection Based on Feature Selection Using Enhanced Grey Wolf Optimizer and Support Vector Machine Algorithms", Vietnam Journal of Computer Science, vol. 08, no. 02, pp. 177-197, 2020.

[2] H. Chen, Z. Zhang, W. Yin, C. Zhao, F. Wang and Y. Li, "A study on depth classification of defects by machine learning based on hyper-parameter search", Measurement, vol. 189, p. 110660, 2022.,https://doi.org/10.1016/j.measurement.2021.110660.

[3] K. Skorupskaite and H. Bhandari, "Endometriosis and fertility", Obstetrics, Gynaecology & Reproductive Medicine, vol. 31, no. 5, pp. 131-136, 2021., https://doi.org/10.1016/j.ogrm.2021.03.003.

[4] B. Shi et al., "Early Recognition and Discrimination of COVID-19 Severity Using Slime Mould Support Vector Machine for Medical Decision-Making," in IEEE Access, vol. 9, pp. 121996-122015, 2021.

[5] C. Chang, Y. Lu, W. Ting, T. Shen and W. Peng, "An artificial immune system with bootstrap sampling for the diagnosis of recurrent endometrial cancers", Open Medicine, vol. 16, no. 1, pp. 237-245, 2021.

[6] S. Guerriero, M. Pascual, S. Ajossa, M. Neri, E. Musa, B. Graupera, I. Rodriguez and J. Alcazar, "Artificial intelligence (AI) in the detection of rectosigmoid deep endometriosis", European Journal of Obstetrics & Gynecology and Reproductive Biology, vol. 261, pp. 29-33, 2021.

[7] X. Zhu, J. Ying, H. Yang, L. Fu, B. Li and B. Jiang, "Detection of deep myometrial invasion in endometrial cancer MR imaging based on multi-feature fusion and probabilistic support vector machine ensemble", Computers in Biology and Medicine, vol. 134, p. 104487, 2021.

[8] Cyril, C Pretty Diana, J Rene Beulah, Neelakandan Subramani, Prakash Mohan, A Harshavardhan, and D Sivabalaselvamani. “AnAutomated Learning Model for Sentiment Analysis andDataClassification of Twitter Data Using Balanced CA-SVM.”Concurrent Engineering 29, no. 4 (December 2021): 386–95. https://doi.org/10.1177/1063293X211031485.