We start by organizing the data into folders, each representing a different class (e.g., Non-PCOS and PCOS).

We load images from these folders, resize them if necessary, and store their file paths along with their corresponding labels in a Data Frame.

Images are preprocessed and normalized before feeding them into the model.

Preprocessing steps may include resizing, normalization (scaling pixel values to a range [0, 1]), and converting images into arrays.

We define and train a Convolutional Neural Network (CNN) model for image classification.

The CNN model typically consists of convolutional layers followed by max-pooling layers to extract features from images.

The model is trained on the labeled dataset with a specified number of epochs and batch size.

After training the CNN model, we integrate transformer layers on top of the pre-trained CNN.

The transformer model is defined separately and added to the trained CNN model to enhance its capabilities.

This step involves combining the feature extraction capabilities of the CNN with the attention mechanism of the transformer.

The combined model (CNN + Transformer) is compiled and trained on our classification task.

It uses the preprocessed image data as input and is trained to predict the correct class labels.

Features are extracted from the last convolutional layer of the trained model.

These features represent high-level abstract representations of the input images and can be used for further analysis or visualization.

we use the trained model to make predictions on new data.

Predictions are obtained by passing the new data through the trained model and interpreting the output probabilities.

we evaluate the performance of the trained model using various metrics such as accuracy, precision, recall, and F1-score.

Classification reports and confusion matrices are generated to assess the model's performance on different classes.

Trained models are saved in HDF5 format (.h5) for future use.

We can load the saved models later to make predictions on new data without retraining.

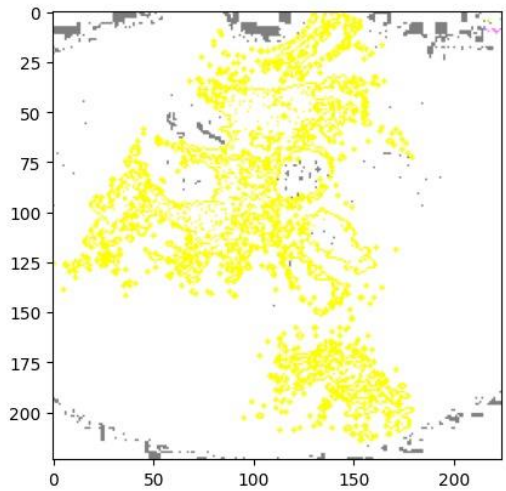
Results:

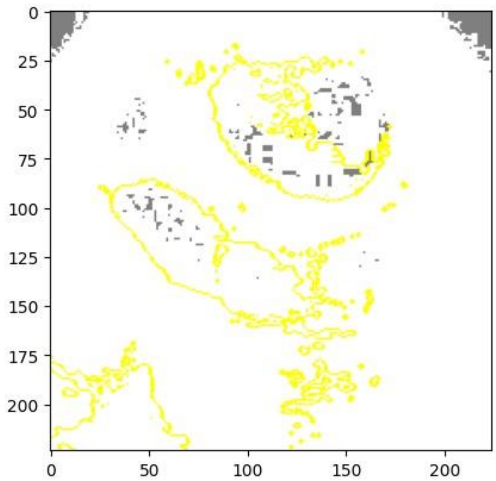
Class	Accuracy	Precision	Recall	F1-score
0(NON-PCOS)	95.14%	98%	95%	96%
1(PCOS)	97.92%	95%	98%	97%

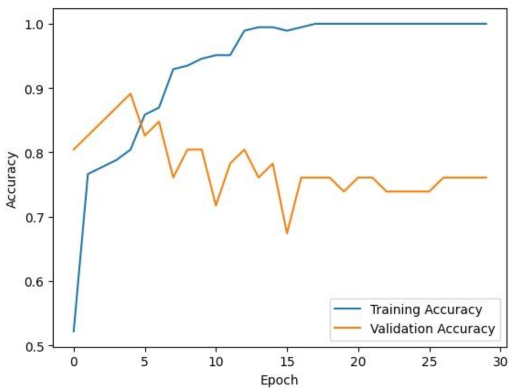
Output images:

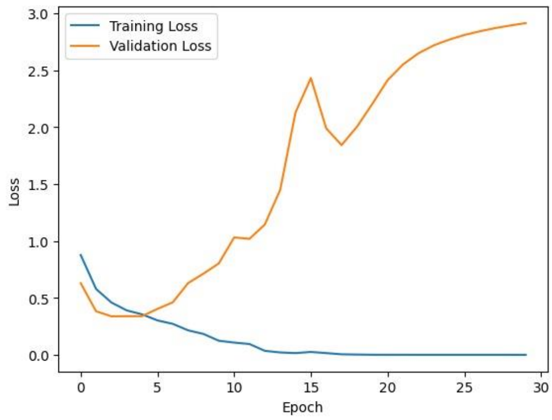
Graphs:

1)Lime Plot









Our Experimentations and Experience:

Firstly we attempted to do on Attention model and then with ensembling models(voting, Bagging, Boosting, Stacking),but the accuracy and training results are not satisfying .Then we created this model where (transfer learning)/Sequential models are the part of Transformers ,
Then we tried to augmenting the dataset and the specification are
Horizontal Flip (Fliplr):

This augmentation horizontally flips 50% of the images.

Applied with a probability of 0.5.

Affine Transformation (Affine):

Rotates images by an angle randomly chosen from the range [-10, 10] degrees.

The rotation angle is randomly selected from a uniform distribution between -10 and 10 degrees.

Gaussian Blur:

Blurs images with a sigma value randomly chosen from the range [0, 1.0].

Sigma represents the standard deviation of the Gaussian kernel.

The sigma value is randomly selected from a uniform distribution between 0 and 1.0.