



Plant Disease Detection

TEAM MEMBERS:

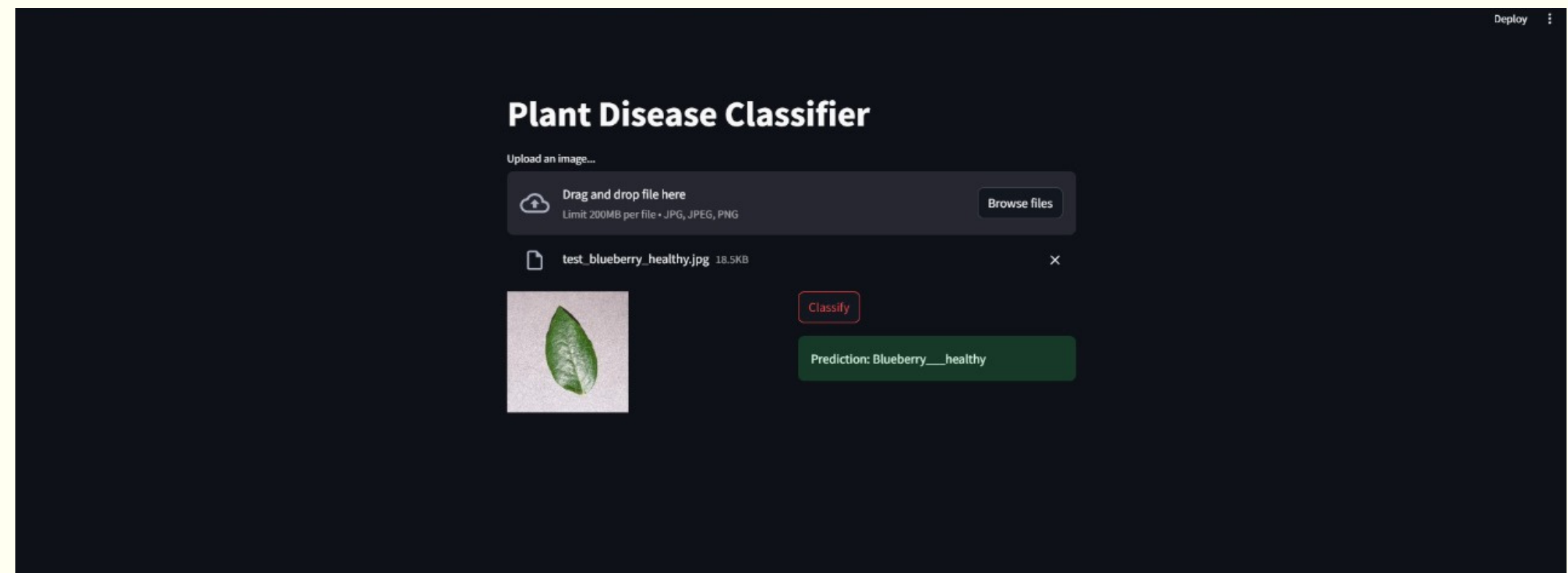
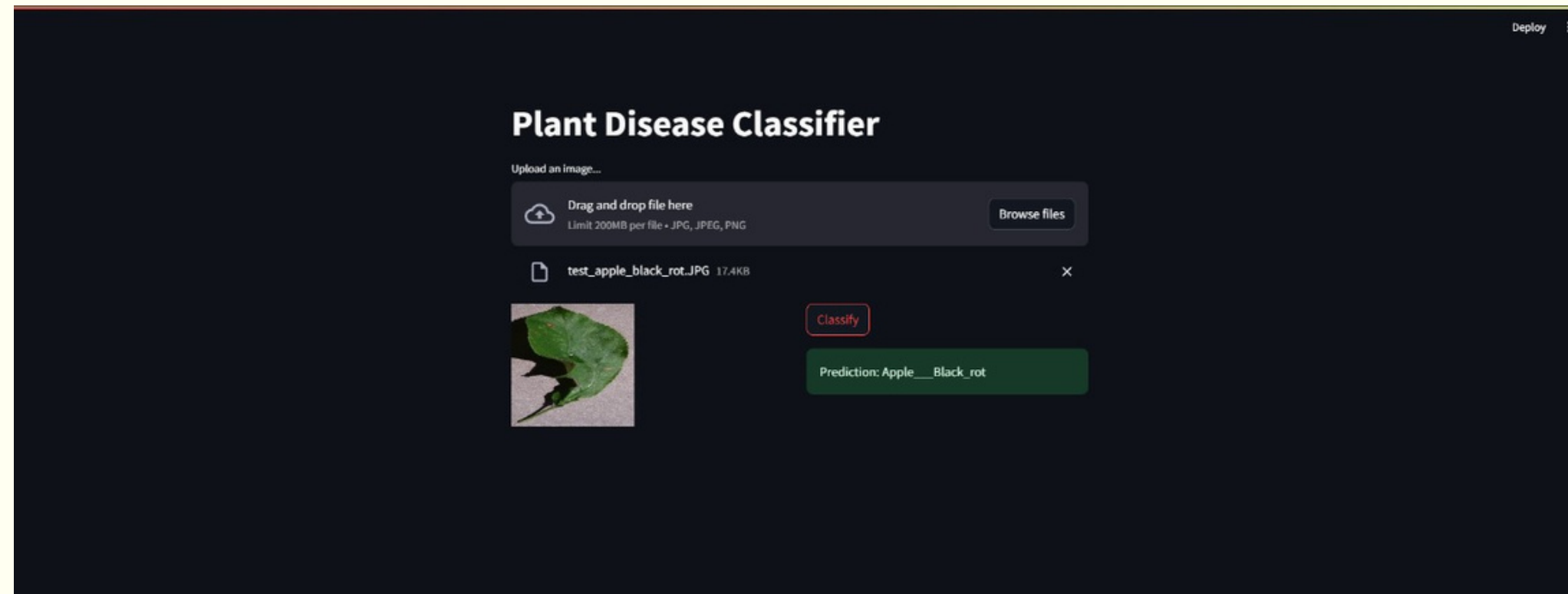
CHAVA GOVARDHANA PES2UG2ICSI41

CHOCKALINGAM A PES2UG2ICSI47

DARSHAN JS PES2UG2ICSI52



DARSHAN V PES2UG2ICSI54

OUTPUT SCREENSHOTS





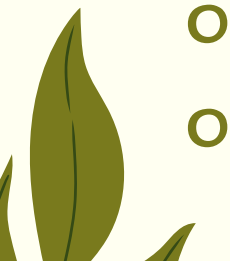
METHODOLOGY

- **Dataset:** The project likely uses the PlantVillage dataset, which contains images of healthy and diseased plants along with labels indicating the type of disease. The dataset is structured into directories, where each directory corresponds to a different class (e.g., different plant diseases).
 - **Preprocessing:** The images are loaded using the ImageDataGenerator from TensorFlow/Keras, which also allows for data augmentation and splitting into training and validation sets. In this case, images are rescaled to have pixel values in the range $[0, 1]$.
 - **Model Architecture:** The model architecture consists of a series of convolutional (Conv2D) and max-pooling (MaxPooling2D) layers followed by fully connected (Dense) layers. Convolutional layers extract features from the input images, and max-pooling layers downsample the feature maps. The final layer uses softmax activation to output probabilities for each class.
- 
- 



METHODOLOGY



- **Training:** The model is compiled with an optimizer (in this case, Adam), a loss function (categorical cross-entropy, suitable for multi-class classification), and evaluation metrics (accuracy). It is then trained using the training data, with validation data used for monitoring performance during training.
 - **Evaluation:** After training, the model's performance is evaluated using the validation set. This involves computing the loss and accuracy on the validation data.
 - **Visualization:** The training history (accuracy and loss over epochs) is plotted using Matplotlib to assess model performance and potential overfitting.
 - **Prediction:** A function is defined to load, preprocess, and predict the class of an unseen image. This function utilizes the trained model to predict the class of a given image.
 - **Example Usage:** An example image is selected (`image_path`), and the model predicts the class of the image using the `predict_image_class` function. The predicted class name is then outputted.
- 
- 