FaultFindy Project Report

1. Introduction:

The FaultFindy project aims to develop an intelligent system using deep learning to predict faulty tyres in the manufacturing process. This predictive capability will help manufacturers optimize processes, reduce waste, and improve overall production efficiency.

2. Design Choices:

2.1 Data Collection:

Historical manufacturing data of good and faulty tyre images were gathered from a specified directory.

2.2 Data Preprocessing:

Various transformations were applied to augment the dataset, including:

- **Resizing**: Standardizing the size of images.
- **Horizontal flipping**: To simulate variation in the data.
- **Rotation**: To ensure the model is invariant to orientation.
- **Gray scaling**: To reduce computational complexity and focus on texture.
- **Gaussian blurring**: To reduce noise. The images were then normalized to ensure consistent input to the model.

2.3 Feature Engineering:

The primary features are the images themselves, with preprocessing transformations applied to enhance the model's ability to generalize.

2.4 Model Selection:

- **ResNet18**: Pretrained on the ImageNet dataset, effective for image classification tasks due to its depth and ability to learn hierarchical features.
- **DenseNet**: Known for its connectivity pattern, facilitating gradient flow and encouraging feature reuse.
- **EfficientNet**: Chosen for its balance between accuracy and computational efficiency.

2.5 Model Training:

The dataset was split into training, validation, and test sets. A batch size of 32 was used for training and evaluation. Fine-tuned only the final layer of each model, freezing the pretrained weights for earlier layers to leverage learned features.

2.6 Model Evaluation:

- **Loss Function**: Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss) was used.
- **Validation Monitoring**: Ensured the model generalizes well to unseen data, avoiding overfitting.

2.7 Hyperparameter Tuning:

Used the Adam optimizer with default parameters for fine-tuning. Due to computational constraints, limited hyperparameter tuning was performed. Future iterations could employ grid search or random search for optimization.

3. Performance Evaluation:

The models were trained over 5 epochs, and the following performance metrics were observed:

- **Training Accuracy**: Increased consistently, indicating the models' learning capability.
- **Validation Accuracy**: Monitored to ensure generalization to unseen data.
- **Test Accuracy**: Assessed the final model performance on the test set, which was withheld during training.

4. Deployment:

A Flask app was created to deploy the trained model, enabling real-time fault detection in a production environment.

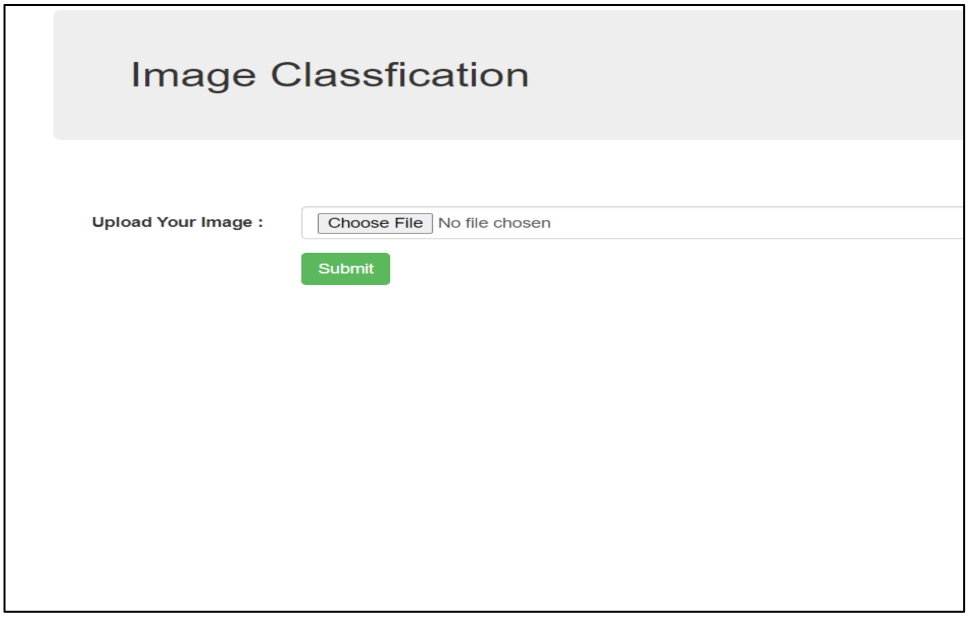
Image Classfication
Upload Your Image :
Choose File
No file chosen
Submit

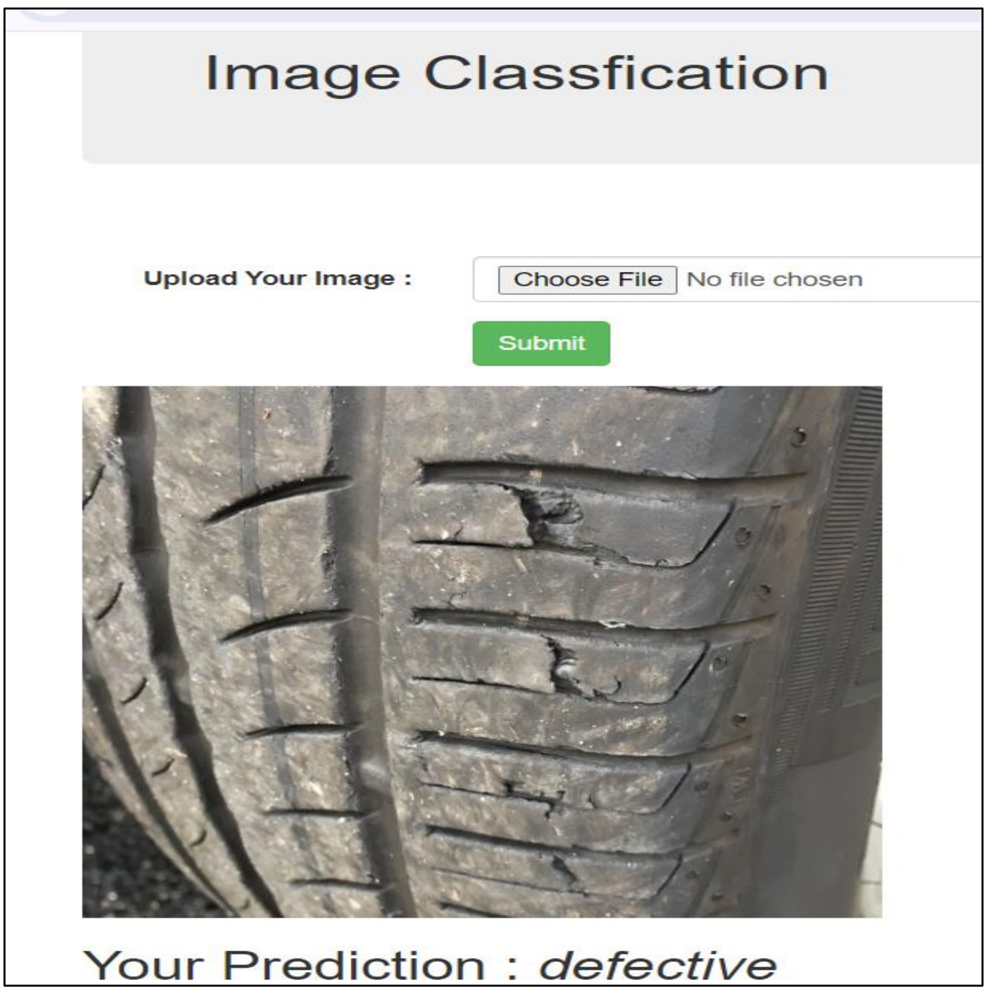
Image Classfication
Upload Your Image :
Choose File
No file chosen
Submit
Your Prediction : defective

5. Future Work:

- **Data Augmentation**: Implement additional augmentation techniques to further enhance the dataset.
- **Hyperparameter Optimization**: Use more advanced techniques to find optimal hyperparameters for better performance.
- **Model Architecture**: Experiment with different architectures like EfficientNet or deeper versions of ResNet.
- **Explainability**: Implement model interpretability techniques to understand which features contribute most to predictions.
- **Real-time Deployment**: Further develop the pipeline for real-time fault detection in a production environment.

This report outlines the steps and methodologies employed in the FaultFindy project, aiming to provide an efficient and accurate predictive system for tyre manufacturing faults. The use of multiple deep learning models and a deployment-ready Flask app underscores the project's practical applications and future potential enhancements.