**Model Optimization and Tuning Phase**

|  |  |
| --- | --- |
| Date | 6 July 2025 |
| Team ID | SWTID1749821186 |
| Project Title | Enhancing Product Reliability: Leveraging Transfer Learning for Fault Detection |
| Maximum Marks | 10 Marks |

### Hyperparameter Tuning Documentation:

|  |  |
| --- | --- |
| **Model** | **Tuned Hyperparameters** |
| VGG16 | Input shape: (224, 224, 3)  Epochs: 20  Learning rate: .0001  Optimizer: adam  Loss: binary crossentropy  Preprocessing: vgg16\_preprocess\_input  EarlyStopping: patience = 5  Additional layers: flatten and dense layer with sigmoid activation function |
| ResNet50 | Input shape: (224, 224, 3)  Epochs: 20  Learning rate: .0001  Optimizer: adam  Loss: binary crossentropy  Preprocessing: resnet50\_preprocess\_input  EarlyStopping: patience = 5  Additional layers: global average pooling 2D and dense layer with sigmoid activation function |
| InceptionV3 | Input shape: (299, 299, 3)  Epochs: 20  Learning rate: .0001  Optimizer: adam  Loss: binary crossentropy  Preprocessing: inceptionv3\_preprocess\_input  EarlyStopping: patience = 5  Additional layers: global average pooling 2D and dense layer with sigmoid activation function |

### Final Model Selection Justification:

|  |  |
| --- | --- |
| **Final Model** | **Reasoning** |
| VGG16 | VGG16 was selected as the final optimized model primarily due to its **high performance** (consistently achieving approximately 95% validation accuracy during training) combined with its **relative simplicity and ease of implementation** compared to deeper or more complex architectures like ResNet50 or InceptionV3. While ResNet50 and InceptionV3 offer superior theoretical performance on very large, diverse datasets, VGG16 demonstrated that it could achieve the required accuracy for our specific task with a more straightforward architecture. This translates to **faster iteration cycles** during development and potentially **lower inference latency** in deployment, especially if computational resources are constrained. The model's well-understood architecture and established transfer learning practices also contributed to its selection, allowing for efficient fine-tuning of its later layers to adapt to our specific dataset without requiring extensive hyperparameter searches or risking catastrophic forgetting. The balance of robust feature extraction capabilities and practical deployment considerations made VGG16 the most suitable and efficient choice for this project. |