Model Development Phase

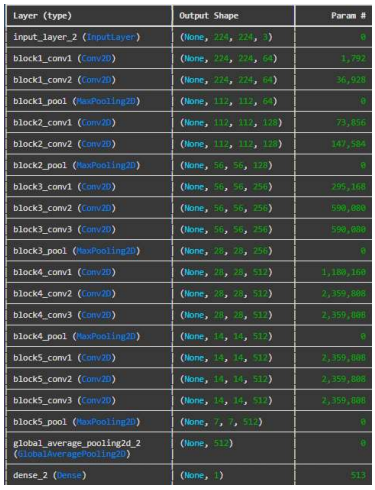
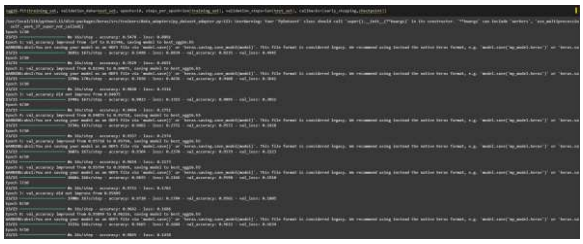
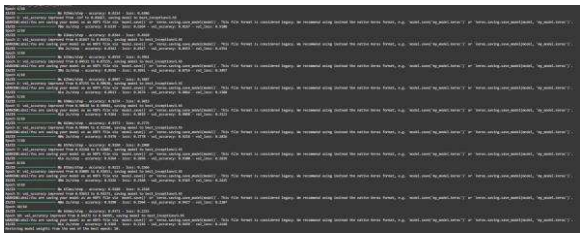
Date	6 July 2025
Team ID	SWTID1749821186
Project Title	Enhancing Product Reliability: Leveraging Transfer Learning for Fault Detection
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

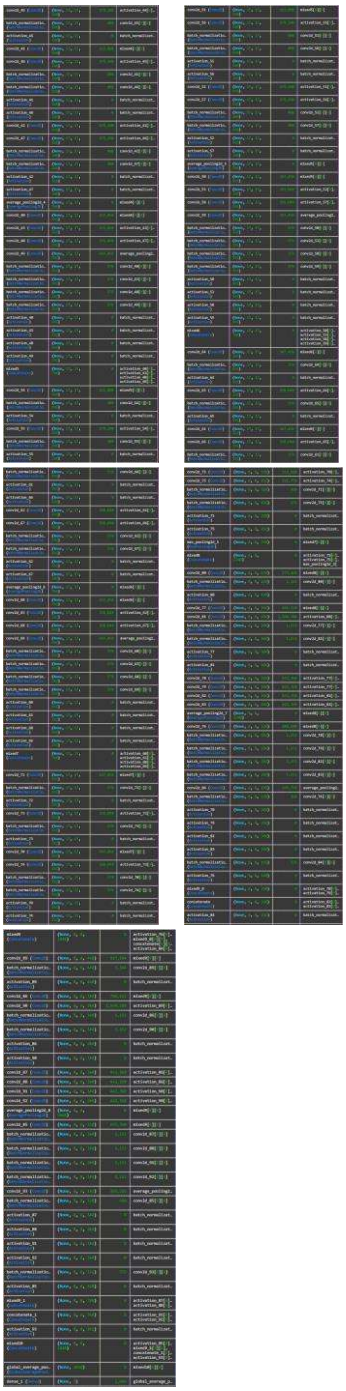
Initial Model Training Code:

```
def build_and_train_model(base_model_func, preprocess_func, model_name, input_shape=(224, 224, 3), epochs=10):
    print(f"\n--- Training {model_name} ---")

    # Load the base model
    base_model = base_model_func(weights="imagenet", include_top=False, input_shape=input_shape)

    # Freeze the layers of the base model
    for layer in base_model.layers:
        layer.trainable = False

    # Add custom classification head
    x = GlobalAveragePooling2D()(base_model.output)
    output = Dense(1, activation='sigmoid')(x)

    model = Model(inputs=base_model.input, outputs=output)

    model.summary()

    # Compile the model
    opt = Adam(learning_rate=0.001)

    model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])

    # Callbacks
    checkpoint = ModelCheckpoint(
        f'best_{model_name.lower()}.h5',
        monitor='val_accuracy',
        save_best_only=True,
        mode='max',
        verbose=1
    )
    early_stopping = EarlyStopping(monitor='val_accuracy', patience=5, restore_best_weights=True, verbose=1)

    # Create data generators for this specific model
    training_set, test_set = create_data_generators(train_directory, test_directory, preprocess_func, target_size=input_shape[:2])

    # Train the model
    history = model.fit(
        training_set,
        validation_data=test_set,
        epochs=epochs,
        steps_per_epoch=len(training_set),
        validation_steps=len(test_set),
        callbacks=[early_stopping, checkpoint]
    )

    print(f"\n--- {model_name} Training Complete ---")
    return model, history, test_set
```

Model Validation and Evaluation Report:

Model	Summary	Training and Validation Performance Metrics
VGG16		
Inceptionv3		

Layer (type)	Output Shape	Param #
input_layer_2 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513



Resnet50