Model Summary

The objective of this task was to develop a deep learning model to classify oral diseases into two classes: **Caries** and **Gingivitis**. The dataset provided had the following structure:

- **Training Data**:
  - Caries: 744 images
  - Gingivitis: 742 images
- **Testing Data**:
  - Caries: 204 images
  - Gingivitis: 204 images

The model used was based on **Transfer Learning** with a pre-trained **EfficientNetB0** architecture. This helped utilise the knowledge of a pre-trained model for better results on the given dataset.

Training Results

The model was trained for 20 epochs with a learning rate that was dynamically adjusted using the **ReduceLROnPlateau** scheduler. Below are the observations from the training process:

1. The training and validation accuracy steadily improved, reaching a final validation accuracy of **92.65%** at epoch 20.
2. The loss values consistently decreased for both training and validation, indicating the model learned effectively.

Test Results

After training, the model was evaluated on the test dataset. Here are the results:

- **Test Accuracy**: **91.42%**
- **Test Loss**: **0.2724**

Confusion Matrix

	Predicted: Caries	Predicted: Gingivitis
Actual: Caries	184	20
Actual: Gingivitis	15	189

This shows that:

- 184 Caries images were correctly classified, and 20 were misclassified as Gingivitis.
- 189 Gingivitis images were correctly classified, and 15 were misclassified as Caries.

Classification Report

Class	Precision	Recall	F1-Score
Caries	0.92	0.90	0.91
Gingivitis	0.90	0.93	0.92

Overall Metrics:

- **Precision**: How many of the predicted positive cases were correct? **91%**
- **Recall**: How many of the actual positive cases were identified? **91%**
- **F1-Score**: The harmonic mean of precision and recall. **91%**

Insights from Plots

1. **Accuracy Plot**: The training and validation accuracy curves converge after a few epochs, indicating the model is generalizing well.
2. **Loss Plot**: Both training and validation loss decrease steadily without overfitting, which shows the model is learning effectively.

Model Limitations

Some misclassifications (20 Caries as Gingivitis and 15 Gingivitis as Caries) indicate that the model might struggle with borderline or unclear images.

Variations in lighting, orientation, or image quality could still pose challenges despite preprocessing.

Possible Improvements

1. **Data Augmentation**: We can introduce more variations in the training data (e.g., flips, rotations, zooms) to help the model generalize further.
2. **Fine-Tuning**: Further fine-tune the model to extract more meaningful features.
3. **Use More Data**: If possible, we can include additional labeled data to improve performance further.

How approach was changed to get better results

Initial Approach and Issues

1. **Learning Rate Issues**:
   - We used a constant learning rate at the start. This caused the model to either learn too fast and miss important details or learn too slowly and get stuck.
   - Result: The model didn't improve much during training, and validation accuracy fluctuated.
2. **Model Architecture**:
   - Initially, the model architecture was simpler, and it wasn't extracting enough useful features from the images.
   - Result: The model couldn't handle variations in the data, like different lighting conditions or orientations.
3. **Overfitting Risk**:
   - The model performed well on training data but not on validation data, which showed signs of **overfitting** (memorizing training data instead of generalizing to unseen data).

Earlier with below settings, we get results as follows:

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_HEIGHT, IMG_WIDTH, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(1, activation='sigmoid')
])
model.compile(optimizer=Adam(learning_rate=0.001),
```

```
        loss='binary_crossentropy',
        metrics=['accuracy'])
EPOCHS = 10
history = model.fit(
  train_generator,
  epochs=EPOCHS,
  validation_data=test_generator
)
```

Results:

```
Epoch 1/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 28s 536ms/step - accuracy: 0.6354 - loss: 1.0740 - val_accuracy: 0.7426 - val_loss: 0.4941
Epoch 2/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 39s 496ms/step - accuracy: 0.8760 - loss: 0.2855 - val_accuracy: 0.8015 - val_loss: 0.4681
Epoch 3/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 25s 521ms/step - accuracy: 0.9433 - loss: 0.1779 - val_accuracy: 0.8946 - val_loss: 0.2951
Epoch 4/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 25s 523ms/step - accuracy: 0.9575 - loss: 0.1208 - val_accuracy: 0.8652 - val_loss: 0.3559
Epoch 5/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 24s 494ms/step - accuracy: 0.9571 - loss: 0.1244 - val_accuracy: 0.9240 - val_loss: 0.2156
Epoch 6/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 25s 507ms/step - accuracy: 0.9618 - loss: 0.0953 - val_accuracy: 0.9216 - val_loss: 0.2582
Epoch 7/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 24s 503ms/step - accuracy: 0.9822 - loss: 0.0685 - val_accuracy: 0.8676 - val_loss: 0.3421
Epoch 8/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 25s 521ms/step - accuracy: 0.9749 - loss: 0.0792 - val_accuracy: 0.8725 - val_loss: 0.3271
Epoch 9/10
47/47 ━━━━━━━━━━━━━━━━━━━━ 24s 501ms/step - accuracy: 0.9770 - loss: 0.0655 - val_accuracy: 0.9167 - val_loss: 0.2837
```

Epoch 10/10
47/47 ━━━━━━━━━━━━━━━━━━━━ **24s** 497ms/step - accuracy: 0.9797 - loss: 0.0580 - val_accuracy: 0.9118 - val_loss: 0.2940

Steps We Took to Improve

1. **Using Transfer Learning**:
   - We switched to a **pre-trained EfficientNetB0** model. This model had already learned important image features from a large dataset, which we fine-tuned for our specific task.
   - Result: The model started identifying disease features much better, leading to improved accuracy.
2. **Dynamic Learning Rate**:
   - We implemented a **learning rate scheduler** (ReduceLROnPlateau) that reduced the learning rate when the model stopped improving.
   - Result: The model learned more efficiently and avoided "stagnating" during training.
3. **Early Stopping**:
   - To prevent overfitting, we monitored the validation performance and stopped training when the validation accuracy stopped improving.
   - Result: The model generalized better to unseen data.
4. **More Epochs**:
   - Initially, we trained the model for fewer epochs (iterations over the data), which didn't allow the model to learn enough. Increasing the training epochs helped the model capture more features.
   - Result: The accuracy improved significantly after more training.
5. **Fine-Tuning and Adjustments**:
   - We fine-tuned the EfficientNetB0 model's deeper layers for better performance.

- Result: The model became better at distinguishing between Caries and Gingivitis.