MISHRA_PRATEET_FINAL_PROJECT

Name: PRATEET MISHRA

Github Username: PrateetMishraUSC

USC ID: 9074068892

1. Transfer Learning for Image Classification

Import Required Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import cv2
import os
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score, classification_report
from tensorflow.keras.preprocessing.image import img_to_array, load_img
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import models
from tensorflow.keras import layers
from tensorflow.keras import callbacks
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.applications import ResNet101
from tensorflow.keras.applications import EfficientNetB0
from tensorflow.keras.applications import VGG16
from tensorflow.keras.regularizers import l2
from tensorflow.keras.applications.resnet50 import preprocess_input as preprocess_resnet
from tensorflow.keras.applications.efficientnet import preprocess_input as preprocess_efficientnet
from tensorflow.keras.applications.vgg16 import preprocess_input as preprocess_vgg16
```

1. (a) After Importing libraries, build a classifier that distinguishes images of six scenes.

1. (b) Data Exploration and Pre-processing

In [2]:

```
train_dir = '../data/seg_train'
test_dir = '../data/seg_test'
```

In [3]:

```
image_size = (150, 150)
batch_size = 32
def getting_paths_and_labels(directory, img_size, model):
    file_paths = []
    labels = []
    images = []

    class_names = sorted(os.listdir(directory))
    for label, class_name in enumerate(class_names):
        class_dir = os.path.join(directory, class_name)
        if os.path.isdir(class_dir):
            for file_name in os.listdir(class_dir):
                file_path = os.path.join(class_dir, file_name)
                file_paths.append(file_path)
                labels.append(label)

                img = load_img(file_path, target_size=img_size)
                img_array = img_to_array(img)
                images.append(img_array)

    images = np.array(images)
    labels = np.array(labels)

    images = preprocessing_images(images, model)

    return images, labels, class_names
```

In [4]:
```
def preprocessing_images(images, model):
    if model in ["ResNet50", "ResNet101"]:
        return preprocess_resnet(images)
    elif model == "EfficientNetB0":
        return preprocess_efficientnet(images)
    elif model == "VGG16":
        return preprocess_vgg16(images)
    else:
        raise ValueError(f"Unknown model name: {model}")
```

In [5]:
```
model_name_50 = "ResNet50"
training_images, training_labels, class_names = getting_paths_and_labels(train_dir, image_size, model_name_
test_images, test_labels, class_names = getting_paths_and_labels(test_dir, image_size, model_name_50)
```

In [6]:
```
print(class_names)
```

```
['buildings', 'forest', 'glacier', 'mountain', 'sea', 'street']
```

In [7]:
```
print(len(training_images))
```

```
14034
```

Label Encoding

In [8]:
```
encoder = LabelEncoder()
training_labels_encoded = encoder.fit_transform(training_labels)
test_labels_encoded = encoder.fit_transform(test_labels)
```

Splitting the training dataset into 80% training and 20% validation set

In [9]:
```
xTrain, yTrain, xTest, yTest = train_test_split(
    training_images, training_labels_encoded, test_size=0.2, random_state=42, stratify=training_labels_enc
```

In [10]:
```
print(xTrain.shape)
print(yTrain.shape)
print(xTest.shape)
print(yTest.shape)
```

```
(11227, 150, 150, 3)
(2807, 150, 150, 3)
(11227,)
(2807,)
```

One Hot Encoding

In [10]:

```
training_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(xTest, num_classes=6)
validation_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(yTest, num_classes=6)
test_labels_one_hot_encoding = tf.keras.utils.to_categorical(test_labels_encoded, num_classes=6)
```

Initializing training and validation dataset

In [11]:

```
train_ds = tf.data.Dataset.from_tensor_slices((xTrain, training_labels_with_one_hot_encoding)).batch(batch_

val_ds = tf.data.Dataset.from_tensor_slices((yTrain, validation_labels_with_one_hot_encoding)).batch(batch_
```

Early Stopping

In [12]:

```
early_stopping = callbacks.EarlyStopping(
    monitor='val_loss',
    patience=10,
    restore_best_weights=True,
    verbose=1,
    start_from_epoch=50
)
```

1. (c) Transfer Learning

1. (c) (ii) To perform empirical regularization, crop, randomly zoom, rotate, flip, contrast, and translate images in your training set for image augmentation

In [13]:

```
data_augmentation = tf.keras.Sequential([
    layers.RandomRotation(0.1),
    layers.RandomTranslation(0.1, 0.1),
    layers.RandomZoom(0.1),
    layers.RandomFlip("vertical"),
    layers.RandomContrast(0.1),
],name="data_augmentation")
```

1. (c) (i) When dealing with classification of relatively small image datasets, deep networks may not perform very well because of not having enough data to train them. In such cases, one usually uses transfer learning, which uses deep learning models that are trained on very large datasets such as ImageNet as feature extractors. The idea is that such deep networks have learned to extract meaningful features from an image using their layers, and those fea- tures can be used in learning other tasks. In order to do that, usually the last layer or the last few layers of the pre-trained network are removed, and the response of the layer before the removed layers to the images in the new dataset is used as a feature vector to train one more multiple replacement lay- ers. In this project, you will use pre-trained models ResNet50, ResNet100, EfficientNetB0 and VGG16. For those pre-trained networks, you will only train the last fully connected layer, and will freeze all layers before them (i.e. we do not change their parameters during training) and use the outputs of the penultimate layer in the original pre-trained model as the features extracted from each image.

AND

1. (c) (iii) Use ReLU activation functions in the last layer and a softmax layer, along with regularization, batch normalization, and a dropout rate of 20% as well as ADAM optimizer. Use multinomial cross entropy loss

In [14]:

```
def model_with_layers(model_name):
    if model_name == "ResNet50":
        pretrained_model = ResNet50(include_top=False, pooling='avg', weights='imagenet')
    elif model_name == "ResNet101":
        pretrained_model = ResNet101(include_top=False, pooling='avg', weights='imagenet')
    elif model_name == "EfficientNetB0":
        pretrained_model = EfficientNetB0(include_top=False, pooling='avg', weights='imagenet')
    elif model_name == "VGG16":
        pretrained_model = VGG16(include_top=False, pooling='avg', weights='imagenet')
    else:
        raise ValueError(f"Unknown model name: {model_name}")
    for layer in pretrained_model.layers:
            layer.trainable=False

    model = models.Sequential([
    data_augmentation,
    pretrained_model,
    layers.Flatten(),
    layers.Dense(512, activation='relu', kernel_regularizer=l2(0.001)),
    layers.BatchNormalization(),
    layers.Dropout(0.2),
    layers.Dense(6, activation='softmax', kernel_regularizer=l2(0.001))
    ])

    model.build(input_shape=(None, 150, 150, 3))

    model.summary()

    model.compile(optimizer=Adam(learning_rate=0.0001),loss='categorical_crossentropy',metrics=['accuracy'

    return model
```

1. (c) (iv) Train the networks for at least 50 epochs (preferably 100 epochs) and perform early stopping using the validation set. Use a randomly selected subset of 20% of each class as the validation set. Keep the network parameters that have the lowest validation error. Plot the training and validation errors vs. epochs

In [15]:

```
def get_history_and_fit_model(model):
    his = model.fit(
        train_ds,
        validation_data=val_ds,
        epochs=100,
        callbacks=[early_stopping]
    )
    return his

def plotting_the_curves(his):
    training_loss = his.history['loss']
    validation_loss = his.history['val_loss']

    training_accuracy = his.history.get('accuracy')
    validation_accuracy = his.history.get('val_accuracy')

    epochs_range = range(1, len(training_loss) + 1)

    plt.figure(figsize=(8, 4))
    plt.plot(epochs_range, training_loss, label='Training Loss')
    plt.plot(epochs_range, validation_loss, label='Validation Loss', linestyle='--')
    plt.title('Training and Validation Loss vs. Epochs')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend(loc='upper right')
    plt.grid(True)
    plt.show()

    plt.figure(figsize=(8, 4))
    plt.plot(epochs_range, training_accuracy, label='Training Accuracy')
    plt.plot(epochs_range, validation_accuracy, label='Validation Accuracy', linestyle='--')
    plt.title('Training and Validation Accuracy vs. Epochs')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend(loc='upper right')
    plt.grid(True)
    plt.show()

def get_test_loss_and_accuracy_with_all_scores(model, model_name):
    test_values = model.evaluate(test_images, test_labels_one_hot_encoding, batch_size=batch_size, verbose

    test_loss = test_values[0]
    test_accuracy = test_values[1]

    print(f"Test Loss \033[1m{model_name}\033[0m: {test_loss:.4f}")
    print(f"Test Accuracy \033[1m{model_name}\033[0m: {test_accuracy:.4f}")

    test_predictions = model.predict(test_images, batch_size=batch_size, verbose=1)

    test_pred_classes = np.argmax(test_predictions, axis=1)

    true_classes = np.argmax(test_labels_one_hot_encoding, axis=1)

    precision = precision_score(true_classes, test_pred_classes, average='weighted')
    recall = recall_score(true_classes, test_pred_classes, average='weighted')
    f1 = f1_score(true_classes, test_pred_classes, average='weighted')
    auc = roc_auc_score(test_labels_one_hot_encoding, test_predictions, multi_class='ovr')
    print(f"\n----------For \033[1m{model_name}\033[0m Model----------\n")

    results = pd.DataFrame({
        "Metric": ["Precision", "Recall", "F1 Score", "AUC"],
        "Score": [precision, recall, f1, auc]
    })

    print(results)

    print("\nClassification Report:")
    print(classification_report(true_classes, test_pred_classes, target_names=class_names))

    return test_loss, test_accuracy, precision, recall, f1, auc
```

Now we will be performing all the above steps for all the 4 models.

ResNet50 Model (Already performed few steps above specifically for this model)

In [16]:
```
model_res50 = model_with_layers("ResNet50")
```

```
Model: "sequential"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ data_augmentation (Sequential)  │ (None, 150, 150, 3)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ resnet50 (Functional)           │ (None, 2048)           │    23,587,712 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten (Flatten)               │ (None, 2048)           │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense (Dense)                   │ (None, 512)            │     1,049,088 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization             │ (None, 512)            │         2,048 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout (Dropout)               │ (None, 512)            │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_1 (Dense)                 │ (None, 6)              │         3,078 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
```

```
Total params: 24,641,926 (94.00 MB)
Trainable params: 1,053,190 (4.02 MB)
Non-trainable params: 23,588,736 (89.98 MB)
```

In [17]:
```
history_res50 = get_history_and_fit_model(model_res50)
```

```
Epoch 1/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 313ms/step - accuracy: 0.6518 - auc: 0.8990 - f1_score: 0.6505 - loss: 1.
8258 - precision: 0.7018 - recall: 0.5920 - val_accuracy: 0.8500 - val_auc: 0.9799 - val_f1_score: 0.8509 -
val_loss: 1.2231 - val_precision: 0.8626 - val_recall: 0.8408
Epoch 2/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 109s 312ms/step - accuracy: 0.7916 - auc: 0.9653 - f1_score: 0.7924 - loss: 1.
3809 - precision: 0.8184 - recall: 0.7683 - val_accuracy: 0.8404 - val_auc: 0.9776 - val_f1_score: 0.8391 -
val_loss: 1.2392 - val_precision: 0.8533 - val_recall: 0.8333
Epoch 3/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 314ms/step - accuracy: 0.8190 - auc: 0.9736 - f1_score: 0.8197 - loss: 1.
2670 - precision: 0.8394 - recall: 0.7973 - val_accuracy: 0.8700 - val_auc: 0.9828 - val_f1_score: 0.8715 -
val_loss: 1.1268 - val_precision: 0.8808 - val_recall: 0.8636
Epoch 4/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 321ms/step - accuracy: 0.8278 - auc: 0.9761 - f1_score: 0.8285 - loss: 1.
2112 - precision: 0.8536 - recall: 0.8094 - val_accuracy: 0.8696 - val_auc: 0.9824 - val_f1_score: 0.8697 -
val_loss: 1.1165 - val_precision: 0.8814 - val_recall: 0.8603
Epoch 5/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.8405 - auc: 0.9787 - f1_score: 0.8410 - loss: 1.
1589 - precision: 0.8584 - recall: 0.8208 - val_accuracy: 0.8636 - val_auc: 0.9836 - val_f1_score: 0.8659 -
val_loss: 1.0784 - val_precision: 0.8712 - val_recall: 0.8557
Epoch 6/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 109s 311ms/step - accuracy: 0.8442 - auc: 0.9799 - f1_score: 0.8444 - loss: 1.
1102 - precision: 0.8618 - recall: 0.8273 - val_accuracy: 0.8764 - val_auc: 0.9855 - val_f1_score: 0.8782 -
val_loss: 1.0080 - val_precision: 0.8825 - val_recall: 0.8671
Epoch 7/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 313ms/step - accuracy: 0.8556 - auc: 0.9818 - f1_score: 0.8562 - loss: 1.
0591 - precision: 0.8725 - recall: 0.8336 - val_accuracy: 0.8721 - val_auc: 0.9863 - val_f1_score: 0.8730 -
val_loss: 0.9856 - val_precision: 0.8815 - val_recall: 0.8668
Epoch 8/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 314ms/step - accuracy: 0.8586 - auc: 0.9831 - f1_score: 0.8590 - loss: 1.
0190 - precision: 0.8764 - recall: 0.8405 - val_accuracy: 0.8846 - val_auc: 0.9859 - val_f1_score: 0.8851 -
val_loss: 0.9540 - val_precision: 0.8917 - val_recall: 0.8771
Epoch 9/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 312ms/step - accuracy: 0.8632 - auc: 0.9843 - f1_score: 0.8638 - loss: 0.
9752 - precision: 0.8797 - recall: 0.8460 - val_accuracy: 0.8821 - val_auc: 0.9860 - val_f1_score: 0.8842 -
val_loss: 0.9226 - val_precision: 0.8896 - val_recall: 0.8757
Epoch 10/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 312ms/step - accuracy: 0.8635 - auc: 0.9856 - f1_score: 0.8643 - loss: 0.
9382 - precision: 0.8798 - recall: 0.8480 - val_accuracy: 0.8739 - val_auc: 0.9846 - val_f1_score: 0.8741 -
val_loss: 0.9256 - val_precision: 0.8837 - val_recall: 0.8689
Epoch 11/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 318ms/step - accuracy: 0.8640 - auc: 0.9848 - f1_score: 0.8644 - loss: 0.
```

```
9192 - precision: 0.8798 - recall: 0.8482 - val_accuracy: 0.8824 - val_auc: 0.9870 - val_f1_score: 0.8840 -
val_loss: 0.8682 - val_precision: 0.8890 - val_recall: 0.8757
Epoch 12/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 319ms/step - accuracy: 0.8681 - auc: 0.9857 - f1_score: 0.8686 - loss: 0.
8893 - precision: 0.8864 - recall: 0.8521 - val_accuracy: 0.8878 - val_auc: 0.9872 - val_f1_score: 0.8896 -
val_loss: 0.8382 - val_precision: 0.8933 - val_recall: 0.8831
Epoch 13/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.8808 - auc: 0.9875 - f1_score: 0.8810 - loss: 0.
8370 - precision: 0.8937 - recall: 0.8670 - val_accuracy: 0.8817 - val_auc: 0.9851 - val_f1_score: 0.8830 -
val_loss: 0.8398 - val_precision: 0.8884 - val_recall: 0.8739
Epoch 14/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 315ms/step - accuracy: 0.8728 - auc: 0.9871 - f1_score: 0.8733 - loss: 0.
8240 - precision: 0.8853 - recall: 0.8543 - val_accuracy: 0.8864 - val_auc: 0.9860 - val_f1_score: 0.8877 -
val_loss: 0.8092 - val_precision: 0.8924 - val_recall: 0.8807
Epoch 15/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 315ms/step - accuracy: 0.8768 - auc: 0.9883 - f1_score: 0.8773 - loss: 0.
7910 - precision: 0.8902 - recall: 0.8583 - val_accuracy: 0.8885 - val_auc: 0.9879 - val_f1_score: 0.8906 -
val_loss: 0.7717 - val_precision: 0.8956 - val_recall: 0.8831
Epoch 16/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 315ms/step - accuracy: 0.8738 - auc: 0.9881 - f1_score: 0.8742 - loss: 0.
7738 - precision: 0.8905 - recall: 0.8569 - val_accuracy: 0.8867 - val_auc: 0.9873 - val_f1_score: 0.8894 -
val_loss: 0.7582 - val_precision: 0.8916 - val_recall: 0.8824
Epoch 17/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.8809 - auc: 0.9892 - f1_score: 0.8813 - loss: 0.
7397 - precision: 0.8936 - recall: 0.8664 - val_accuracy: 0.8921 - val_auc: 0.9863 - val_f1_score: 0.8941 -
val_loss: 0.7455 - val_precision: 0.8979 - val_recall: 0.8864
Epoch 18/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.8776 - auc: 0.9886 - f1_score: 0.8779 - loss: 0.
7302 - precision: 0.8924 - recall: 0.8618 - val_accuracy: 0.8846 - val_auc: 0.9866 - val_f1_score: 0.8860 -
val_loss: 0.7283 - val_precision: 0.8910 - val_recall: 0.8792
Epoch 19/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.8885 - auc: 0.9896 - f1_score: 0.8890 - loss: 0.
7003 - precision: 0.9024 - recall: 0.8760 - val_accuracy: 0.8814 - val_auc: 0.9861 - val_f1_score: 0.8829 -
val_loss: 0.7251 - val_precision: 0.8871 - val_recall: 0.8789
Epoch 20/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 317ms/step - accuracy: 0.8928 - auc: 0.9910 - f1_score: 0.8932 - loss: 0.
6660 - precision: 0.9063 - recall: 0.8767 - val_accuracy: 0.8835 - val_auc: 0.9866 - val_f1_score: 0.8851 -
val_loss: 0.7087 - val_precision: 0.8894 - val_recall: 0.8764
Epoch 21/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 317ms/step - accuracy: 0.8848 - auc: 0.9899 - f1_score: 0.8854 - loss: 0.
6680 - precision: 0.8990 - recall: 0.8706 - val_accuracy: 0.8931 - val_auc: 0.9888 - val_f1_score: 0.8948 -
val_loss: 0.6618 - val_precision: 0.8982 - val_recall: 0.8892
```

```
Epoch 22/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 313ms/step - accuracy: 0.8890 - auc: 0.9902 - f1_score: 0.8893 - loss: 0.
6512 - precision: 0.9021 - recall: 0.8751 - val_accuracy: 0.8910 - val_auc: 0.9860 - val_f1_score: 0.8924 -
val_loss: 0.6775 - val_precision: 0.8975 - val_recall: 0.8864
Epoch 23/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 108s 308ms/step - accuracy: 0.8886 - auc: 0.9906 - f1_score: 0.8889 - loss: 0.
6318 - precision: 0.9040 - recall: 0.8734 - val_accuracy: 0.8917 - val_auc: 0.9867 - val_f1_score: 0.8933 -
val_loss: 0.6619 - val_precision: 0.8951 - val_recall: 0.8849
Epoch 24/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 109s 311ms/step - accuracy: 0.8947 - auc: 0.9910 - f1_score: 0.8947 - loss: 0.
6137 - precision: 0.9061 - recall: 0.8806 - val_accuracy: 0.8899 - val_auc: 0.9875 - val_f1_score: 0.8918 -
val_loss: 0.6412 - val_precision: 0.8969 - val_recall: 0.8867
Epoch 25/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 315ms/step - accuracy: 0.8985 - auc: 0.9912 - f1_score: 0.8988 - loss: 0.
5963 - precision: 0.9090 - recall: 0.8868 - val_accuracy: 0.8963 - val_auc: 0.9863 - val_f1_score: 0.8980 -
val_loss: 0.6387 - val_precision: 0.9010 - val_recall: 0.8917
Epoch 26/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 109s 312ms/step - accuracy: 0.8883 - auc: 0.9907 - f1_score: 0.8891 - loss: 0.
5976 - precision: 0.9014 - recall: 0.8744 - val_accuracy: 0.8913 - val_auc: 0.9865 - val_f1_score: 0.8934 -
val_loss: 0.6279 - val_precision: 0.8965 - val_recall: 0.8885
Epoch 27/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 314ms/step - accuracy: 0.9036 - auc: 0.9922 - f1_score: 0.9041 - loss: 0.
5660 - precision: 0.9146 - recall: 0.8894 - val_accuracy: 0.8864 - val_auc: 0.9872 - val_f1_score: 0.8886 -
val_loss: 0.6297 - val_precision: 0.8954 - val_recall: 0.8817
Epoch 28/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.9010 - auc: 0.9922 - f1_score: 0.9017 - loss: 0.
5563 - precision: 0.9136 - recall: 0.8868 - val_accuracy: 0.8871 - val_auc: 0.9849 - val_f1_score: 0.8887 -
val_loss: 0.6363 - val_precision: 0.8922 - val_recall: 0.8817
Epoch 29/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 318ms/step - accuracy: 0.8969 - auc: 0.9918 - f1_score: 0.8975 - loss: 0.
5525 - precision: 0.9113 - recall: 0.8859 - val_accuracy: 0.8799 - val_auc: 0.9862 - val_f1_score: 0.8809 -
val_loss: 0.6270 - val_precision: 0.8859 - val_recall: 0.8771
Epoch 30/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 318ms/step - accuracy: 0.9008 - auc: 0.9922 - f1_score: 0.9012 - loss: 0.
5376 - precision: 0.9123 - recall: 0.8907 - val_accuracy: 0.8903 - val_auc: 0.9870 - val_f1_score: 0.8917 -
val_loss: 0.5967 - val_precision: 0.8965 - val_recall: 0.8853
Epoch 31/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 315ms/step - accuracy: 0.9052 - auc: 0.9926 - f1_score: 0.9057 - loss: 0.
5261 - precision: 0.9194 - recall: 0.8907 - val_accuracy: 0.8757 - val_auc: 0.9842 - val_f1_score: 0.8778 -
val_loss: 0.6254 - val_precision: 0.8812 - val_recall: 0.8693
Epoch 32/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 313ms/step - accuracy: 0.9040 - auc: 0.9925 - f1_score: 0.9047 - loss: 0.
```

```
5176 - precision: 0.9150 - recall: 0.8927 - val_accuracy: 0.8778 - val_auc: 0.9828 - val_f1_score: 0.8782 -
val_loss: 0.6477 - val_precision: 0.8832 - val_recall: 0.8732
Epoch 33/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 317ms/step - accuracy: 0.9032 - auc: 0.9923 - f1_score: 0.9035 - loss: 0.
5162 - precision: 0.9141 - recall: 0.8926 - val_accuracy: 0.8867 - val_auc: 0.9866 - val_f1_score: 0.8886 -
val_loss: 0.5816 - val_precision: 0.8910 - val_recall: 0.8821
Epoch 34/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.9086 - auc: 0.9934 - f1_score: 0.9090 - loss: 0.
4902 - precision: 0.9205 - recall: 0.8979 - val_accuracy: 0.8846 - val_auc: 0.9858 - val_f1_score: 0.8867 -
val_loss: 0.5809 - val_precision: 0.8883 - val_recall: 0.8810
Epoch 35/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 318ms/step - accuracy: 0.9073 - auc: 0.9926 - f1_score: 0.9077 - loss: 0.
4937 - precision: 0.9182 - recall: 0.8961 - val_accuracy: 0.8785 - val_auc: 0.9855 - val_f1_score: 0.8800 -
val_loss: 0.5961 - val_precision: 0.8846 - val_recall: 0.8710
Epoch 36/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 318ms/step - accuracy: 0.9089 - auc: 0.9935 - f1_score: 0.9089 - loss: 0.
4775 - precision: 0.9210 - recall: 0.8980 - val_accuracy: 0.8878 - val_auc: 0.9858 - val_f1_score: 0.8895 -
val_loss: 0.5791 - val_precision: 0.8925 - val_recall: 0.8817
Epoch 37/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 317ms/step - accuracy: 0.9123 - auc: 0.9939 - f1_score: 0.9129 - loss: 0.
4674 - precision: 0.9252 - recall: 0.9017 - val_accuracy: 0.8910 - val_auc: 0.9873 - val_f1_score: 0.8931 -
val_loss: 0.5562 - val_precision: 0.8940 - val_recall: 0.8867
Epoch 38/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 315ms/step - accuracy: 0.9028 - auc: 0.9924 - f1_score: 0.9033 - loss: 0.
4816 - precision: 0.9121 - recall: 0.8943 - val_accuracy: 0.8888 - val_auc: 0.9849 - val_f1_score: 0.8896 -
val_loss: 0.5862 - val_precision: 0.8947 - val_recall: 0.8839
Epoch 39/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 314ms/step - accuracy: 0.9111 - auc: 0.9935 - f1_score: 0.9113 - loss: 0.
4633 - precision: 0.9231 - recall: 0.9000 - val_accuracy: 0.8846 - val_auc: 0.9845 - val_f1_score: 0.8856 -
val_loss: 0.5923 - val_precision: 0.8908 - val_recall: 0.8778
Epoch 40/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 315ms/step - accuracy: 0.9105 - auc: 0.9936 - f1_score: 0.9111 - loss: 0.
4517 - precision: 0.9233 - recall: 0.9000 - val_accuracy: 0.8842 - val_auc: 0.9855 - val_f1_score: 0.8857 -
val_loss: 0.5650 - val_precision: 0.8917 - val_recall: 0.8774
Epoch 41/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 314ms/step - accuracy: 0.9140 - auc: 0.9938 - f1_score: 0.9145 - loss: 0.
4475 - precision: 0.9250 - recall: 0.9040 - val_accuracy: 0.8817 - val_auc: 0.9839 - val_f1_score: 0.8833 -
val_loss: 0.5784 - val_precision: 0.8887 - val_recall: 0.8792
Epoch 42/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 313ms/step - accuracy: 0.9074 - auc: 0.9937 - f1_score: 0.9078 - loss: 0.
4474 - precision: 0.9175 - recall: 0.8988 - val_accuracy: 0.8888 - val_auc: 0.9844 - val_f1_score: 0.8898 -
val_loss: 0.5559 - val_precision: 0.8934 - val_recall: 0.8839
```

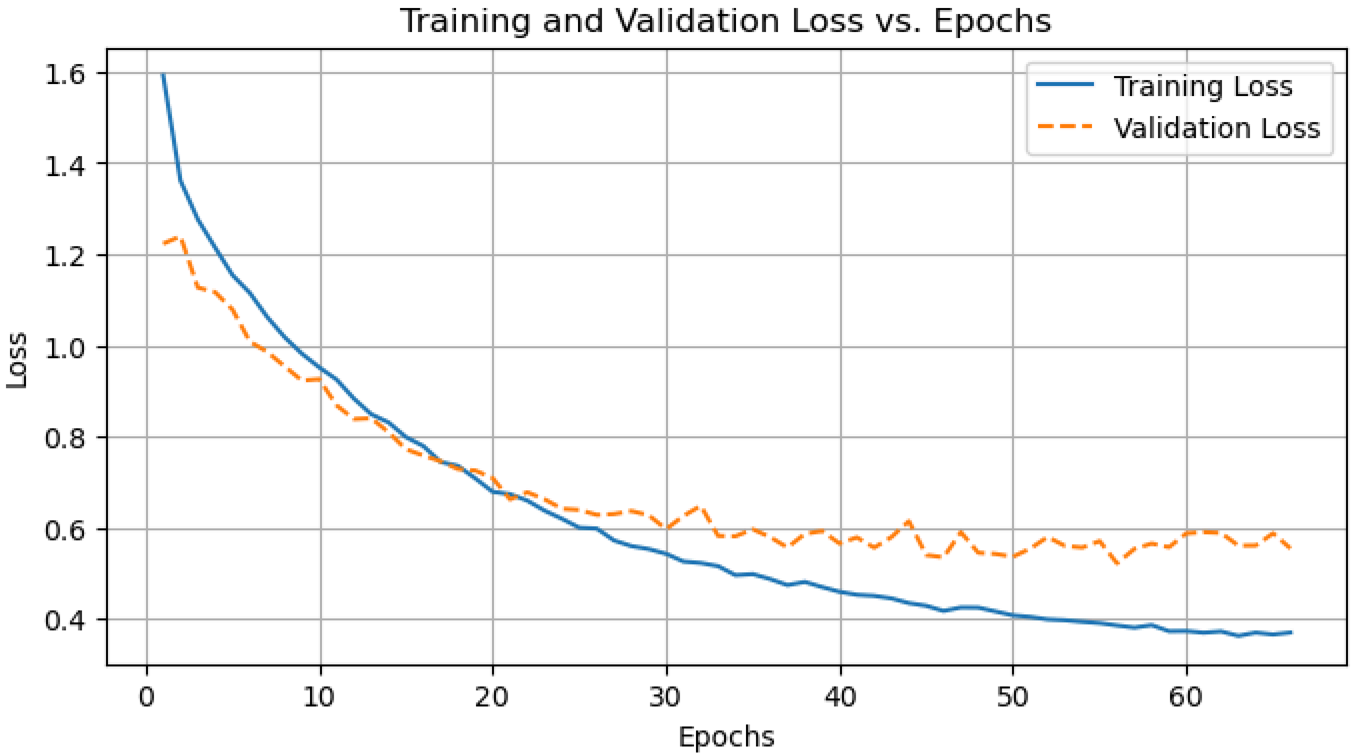
```
Epoch 43/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.9141 - auc: 0.9939 - f1_score: 0.9144 - loss: 0.
4399 - precision: 0.9232 - recall: 0.9008 - val_accuracy: 0.8778 - val_auc: 0.9832 - val_f1_score: 0.8784 -
val_loss: 0.5793 - val_precision: 0.8863 - val_recall: 0.8717
Epoch 44/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.9187 - auc: 0.9944 - f1_score: 0.9188 - loss: 0.
4295 - precision: 0.9268 - recall: 0.9066 - val_accuracy: 0.8739 - val_auc: 0.9811 - val_f1_score: 0.8744 -
val_loss: 0.6140 - val_precision: 0.8764 - val_recall: 0.8693
Epoch 45/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 314ms/step - accuracy: 0.9189 - auc: 0.9941 - f1_score: 0.9193 - loss: 0.
4268 - precision: 0.9265 - recall: 0.9081 - val_accuracy: 0.8913 - val_auc: 0.9856 - val_f1_score: 0.8927 -
val_loss: 0.5396 - val_precision: 0.8956 - val_recall: 0.8864
Epoch 46/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 315ms/step - accuracy: 0.9192 - auc: 0.9946 - f1_score: 0.9195 - loss: 0.
4147 - precision: 0.9289 - recall: 0.9085 - val_accuracy: 0.8903 - val_auc: 0.9859 - val_f1_score: 0.8919 -
val_loss: 0.5348 - val_precision: 0.8990 - val_recall: 0.8849
Epoch 47/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 318ms/step - accuracy: 0.9152 - auc: 0.9944 - f1_score: 0.9154 - loss: 0.
4153 - precision: 0.9242 - recall: 0.9076 - val_accuracy: 0.8735 - val_auc: 0.9824 - val_f1_score: 0.8751 -
val_loss: 0.5899 - val_precision: 0.8797 - val_recall: 0.8700
Epoch 48/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 322ms/step - accuracy: 0.9162 - auc: 0.9942 - f1_score: 0.9165 - loss: 0.
4164 - precision: 0.9247 - recall: 0.9071 - val_accuracy: 0.8864 - val_auc: 0.9857 - val_f1_score: 0.8880 -
val_loss: 0.5447 - val_precision: 0.8909 - val_recall: 0.8842
Epoch 49/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 114s 324ms/step - accuracy: 0.9126 - auc: 0.9936 - f1_score: 0.9128 - loss: 0.
4207 - precision: 0.9218 - recall: 0.9018 - val_accuracy: 0.8853 - val_auc: 0.9847 - val_f1_score: 0.8864 -
val_loss: 0.5421 - val_precision: 0.8923 - val_recall: 0.8821
Epoch 50/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 323ms/step - accuracy: 0.9205 - auc: 0.9950 - f1_score: 0.9209 - loss: 0.
3974 - precision: 0.9301 - recall: 0.9130 - val_accuracy: 0.8864 - val_auc: 0.9849 - val_f1_score: 0.8880 -
val_loss: 0.5361 - val_precision: 0.8909 - val_recall: 0.8814
Epoch 51/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 323ms/step - accuracy: 0.9188 - auc: 0.9945 - f1_score: 0.9194 - loss: 0.
4038 - precision: 0.9284 - recall: 0.9072 - val_accuracy: 0.8849 - val_auc: 0.9842 - val_f1_score: 0.8865 -
val_loss: 0.5538 - val_precision: 0.8907 - val_recall: 0.8824
Epoch 52/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 318ms/step - accuracy: 0.9183 - auc: 0.9947 - f1_score: 0.9184 - loss: 0.
3974 - precision: 0.9268 - recall: 0.9099 - val_accuracy: 0.8757 - val_auc: 0.9820 - val_f1_score: 0.8772 -
val_loss: 0.5794 - val_precision: 0.8814 - val_recall: 0.8710
Epoch 53/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 319ms/step - accuracy: 0.9197 - auc: 0.9951 - f1_score: 0.9201 - loss: 0.
```

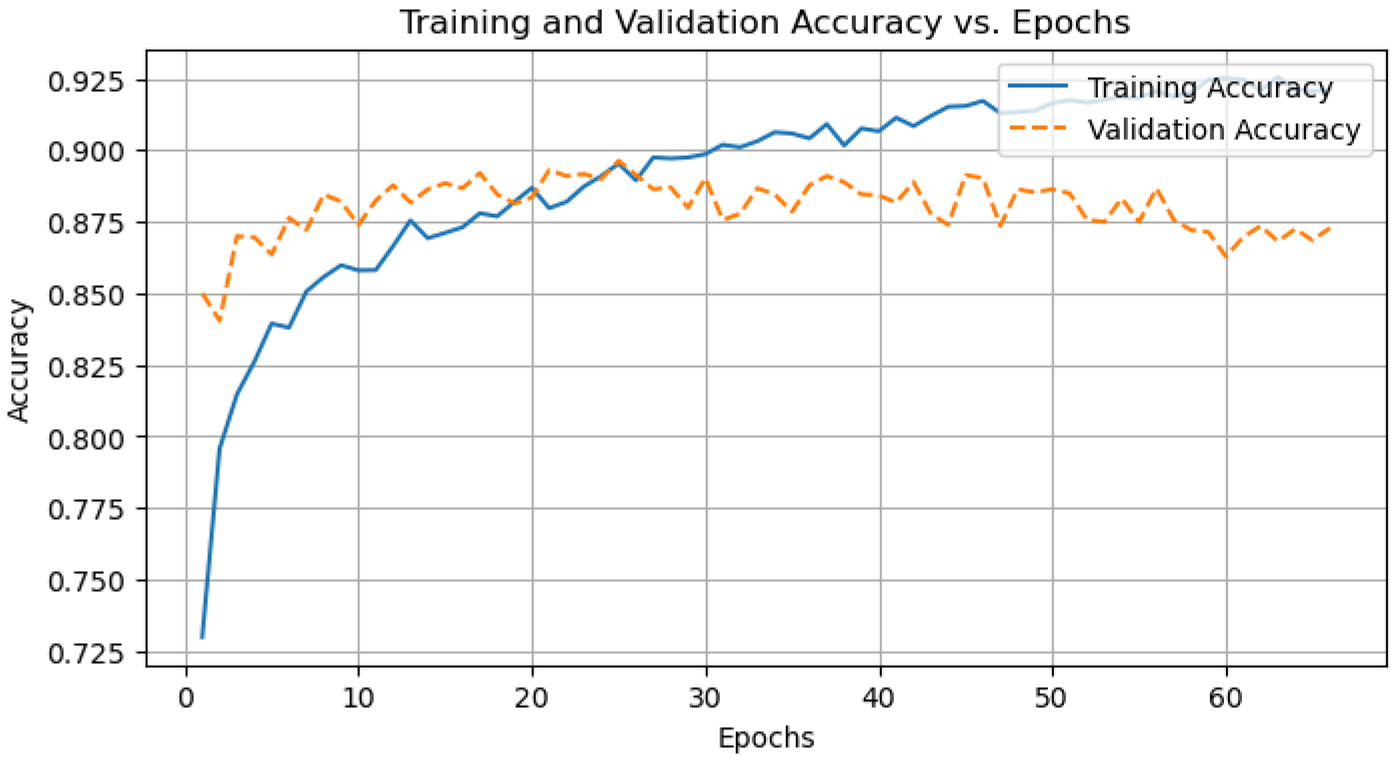
```
3865 - precision: 0.9292 - recall: 0.9121 - val_accuracy: 0.8750 - val_auc: 0.9838 - val_f1_score: 0.8760 -
val_loss: 0.5597 - val_precision: 0.8804 - val_recall: 0.8710
Epoch 54/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.9210 - auc: 0.9948 - f1_score: 0.9213 - loss: 0.
3900 - precision: 0.9317 - recall: 0.9131 - val_accuracy: 0.8831 - val_auc: 0.9829 - val_f1_score: 0.8851 -
val_loss: 0.5562 - val_precision: 0.8877 - val_recall: 0.8785
Epoch 55/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 318ms/step - accuracy: 0.9205 - auc: 0.9951 - f1_score: 0.9207 - loss: 0.
3855 - precision: 0.9294 - recall: 0.9098 - val_accuracy: 0.8750 - val_auc: 0.9830 - val_f1_score: 0.8768 -
val_loss: 0.5703 - val_precision: 0.8798 - val_recall: 0.8710
Epoch 56/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 316ms/step - accuracy: 0.9248 - auc: 0.9952 - f1_score: 0.9253 - loss: 0.
3760 - precision: 0.9355 - recall: 0.9146 - val_accuracy: 0.8867 - val_auc: 0.9857 - val_f1_score: 0.8890 -
val_loss: 0.5215 - val_precision: 0.8938 - val_recall: 0.8785
Epoch 57/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 320ms/step - accuracy: 0.9238 - auc: 0.9956 - f1_score: 0.9242 - loss: 0.
3723 - precision: 0.9328 - recall: 0.9161 - val_accuracy: 0.8757 - val_auc: 0.9837 - val_f1_score: 0.8768 -
val_loss: 0.5530 - val_precision: 0.8839 - val_recall: 0.8703
Epoch 58/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 321ms/step - accuracy: 0.9206 - auc: 0.9949 - f1_score: 0.9208 - loss: 0.
3839 - precision: 0.9302 - recall: 0.9098 - val_accuracy: 0.8721 - val_auc: 0.9830 - val_f1_score: 0.8731 -
val_loss: 0.5649 - val_precision: 0.8785 - val_recall: 0.8678
Epoch 59/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 315ms/step - accuracy: 0.9256 - auc: 0.9953 - f1_score: 0.9259 - loss: 0.
3726 - precision: 0.9357 - recall: 0.9185 - val_accuracy: 0.8714 - val_auc: 0.9825 - val_f1_score: 0.8729 -
val_loss: 0.5569 - val_precision: 0.8774 - val_recall: 0.8668
Epoch 60/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 317ms/step - accuracy: 0.9261 - auc: 0.9944 - f1_score: 0.9264 - loss: 0.
3759 - precision: 0.9336 - recall: 0.9155 - val_accuracy: 0.8628 - val_auc: 0.9816 - val_f1_score: 0.8631 -
val_loss: 0.5873 - val_precision: 0.8683 - val_recall: 0.8575
Epoch 61/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 319ms/step - accuracy: 0.9266 - auc: 0.9954 - f1_score: 0.9269 - loss: 0.
3606 - precision: 0.9345 - recall: 0.9164 - val_accuracy: 0.8696 - val_auc: 0.9797 - val_f1_score: 0.8740 -
val_loss: 0.5899 - val_precision: 0.8746 - val_recall: 0.8646
Epoch 62/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 111s 318ms/step - accuracy: 0.9234 - auc: 0.9951 - f1_score: 0.9238 - loss: 0.
3679 - precision: 0.9334 - recall: 0.9160 - val_accuracy: 0.8735 - val_auc: 0.9801 - val_f1_score: 0.8756 -
val_loss: 0.5882 - val_precision: 0.8780 - val_recall: 0.8689
Epoch 63/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 313ms/step - accuracy: 0.9275 - auc: 0.9955 - f1_score: 0.9279 - loss: 0.
3572 - precision: 0.9343 - recall: 0.9183 - val_accuracy: 0.8682 - val_auc: 0.9812 - val_f1_score: 0.8704 -
val_loss: 0.5604 - val_precision: 0.8735 - val_recall: 0.8636
```

```
Epoch 64/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 112s 320ms/step - accuracy: 0.9244 - auc: 0.9951 - f1_score: 0.9247 - loss: 0.
3647 - precision: 0.9334 - recall: 0.9167 - val_accuracy: 0.8725 - val_auc: 0.9817 - val_f1_score: 0.8744 -
val_loss: 0.5607 - val_precision: 0.8769 - val_recall: 0.8682
Epoch 65/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 113s 323ms/step - accuracy: 0.9221 - auc: 0.9952 - f1_score: 0.9225 - loss: 0.
3631 - precision: 0.9297 - recall: 0.9126 - val_accuracy: 0.8685 - val_auc: 0.9796 - val_f1_score: 0.8711 -
val_loss: 0.5873 - val_precision: 0.8725 - val_recall: 0.8653
Epoch 66/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 110s 312ms/step - accuracy: 0.9200 - auc: 0.9947 - f1_score: 0.9202 - loss: 0.
3692 - precision: 0.9283 - recall: 0.9114 - val_accuracy: 0.8728 - val_auc: 0.9820 - val_f1_score: 0.8743 -
val_loss: 0.5543 - val_precision: 0.8781 - val_recall: 0.8696
Epoch 66: early stopping
Restoring model weights from the end of the best epoch: 56.
```

In [18]:

```
plotting_the_curves(history_res50)
```





In [19]:
```
test_loss_res50, test_accuracy_res50, precision_res50, recall_res50, f1_res50, auc_res50 = get_test_loss_ar
```

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 23s 236ms/step - accuracy: 0.8921 - auc: 0.9851 - f1_score: 0.5373 - loss: 0.500
7 - precision: 0.8988 - recall: 0.8886
Test Loss ResNet50: 0.5403
Test Accuracy ResNet50: 0.8803
94/94 ━━━━━━━━━━━━━━━━━━━━ 23s 239ms/step

----------For ResNet50 Model----------

      Metric     Score
0  Precision  0.879674
1     Recall  0.880333
2   F1 Score  0.879559
3        AUC  0.984165

Classification Report:
              precision    recall  f1-score   support

   buildings       0.90      0.91      0.91       437
      forest       0.97      0.99      0.98       474
     glacier       0.83      0.76      0.79       553
    mountain       0.78      0.81      0.79       525
         sea       0.90      0.94      0.92       510
      street       0.92      0.91      0.91       501

    accuracy                           0.88      3000
   macro avg       0.88      0.88      0.88      3000
weighted avg       0.88      0.88      0.88      3000
```

ResNet101

In [20]:

```
model_name_50 = "ResNet101"
training_images, training_labels, class_names = getting_paths_and_labels(train_dir, image_size, model_name_
test_images, test_labels, class_names = getting_paths_and_labels(test_dir, image_size, model_name_50)
```

In [21]:

```
encoder = LabelEncoder()
training_labels_encoded = encoder.fit_transform(training_labels)
test_labels_encoded = encoder.fit_transform(test_labels)
```

In [22]:

```
xTrain, yTrain, xTest, yTest = train_test_split(
    training_images, training_labels_encoded, test_size=0.2, random_state=42, stratify=training_labels_enc
```

In [23]:

```
training_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(xTest, num_classes=6)
validation_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(yTest, num_classes=6)
test_labels_one_hot_encoding = tf.keras.utils.to_categorical(test_labels_encoded, num_classes=6)
```

In [24]:

```
train_ds = tf.data.Dataset.from_tensor_slices((xTrain, training_labels_with_one_hot_encoding)).batch(batch_

val_ds = tf.data.Dataset.from_tensor_slices((yTrain, validation_labels_with_one_hot_encoding)).batch(batch_
```

In [25]:

```
model_res101 = model_with_layers("ResNet101")
```

```
Model: "sequential_1"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape            ┃        Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━┩
│ data_augmentation (Sequential)       │ (None, 150, 150, 3)     │              0 │
├──────────────────────────────────────┼─────────────────────────┼────────────────┤
│ resnet101 (Functional)               │ (None, 2048)            │     42,658,176 │
├──────────────────────────────────────┼─────────────────────────┼────────────────┤
│ flatten_1 (Flatten)                  │ (None, 2048)            │              0 │
├──────────────────────────────────────┼─────────────────────────┼────────────────┤
│ dense_2 (Dense)                      │ (None, 512)             │      1,049,088 │
├──────────────────────────────────────┼─────────────────────────┼────────────────┤
│ batch_normalization_1                │ (None, 512)             │          2,048 │
│ (BatchNormalization)                 │                         │                │
├──────────────────────────────────────┼─────────────────────────┼────────────────┤
│ dropout_1 (Dropout)                  │ (None, 512)             │              0 │
├──────────────────────────────────────┼─────────────────────────┼────────────────┤
│ dense_3 (Dense)                      │ (None, 6)               │          3,078 │
└──────────────────────────────────────┴─────────────────────────┴────────────────┘

 Total params: 43,712,390 (166.75 MB)

 Trainable params: 1,053,190 (4.02 MB)

 Non-trainable params: 42,659,200 (162.73 MB)
```

In [26]:

```
history_res101 = get_history_and_fit_model(model_res101)
```

```
Epoch 1/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 208s 578ms/step - accuracy: 0.6509 - auc: 0.9044 - f1_score: 0.6492 - loss: 1.
8093 - precision: 0.7032 - recall: 0.5910 - val_accuracy: 0.8408 - val_auc: 0.9764 - val_f1_score: 0.8405 -
val_loss: 1.2765 - val_precision: 0.8565 - val_recall: 0.8251
Epoch 2/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 198s 564ms/step - accuracy: 0.7929 - auc: 0.9656 - f1_score: 0.7929 - loss: 1.
3775 - precision: 0.8189 - recall: 0.7646 - val_accuracy: 0.8450 - val_auc: 0.9800 - val_f1_score: 0.8456 -
val_loss: 1.2184 - val_precision: 0.8542 - val_recall: 0.8351
Epoch 3/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 198s 563ms/step - accuracy: 0.8131 - auc: 0.9704 - f1_score: 0.8140 - loss: 1.
2973 - precision: 0.8357 - recall: 0.7874 - val_accuracy: 0.8247 - val_auc: 0.9753 - val_f1_score: 0.8190 -
val_loss: 1.2601 - val_precision: 0.8372 - val_recall: 0.8169
Epoch 4/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 197s 562ms/step - accuracy: 0.8266 - auc: 0.9761 - f1_score: 0.8270 - loss: 1.
2189 - precision: 0.8476 - recall: 0.8071 - val_accuracy: 0.8564 - val_auc: 0.9817 - val_f1_score: 0.8566 -
val_loss: 1.1354 - val_precision: 0.8652 - val_recall: 0.8486
Epoch 5/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 198s 564ms/step - accuracy: 0.8370 - auc: 0.9790 - f1_score: 0.8373 - loss: 1.
1552 - precision: 0.8571 - recall: 0.8172 - val_accuracy: 0.8568 - val_auc: 0.9820 - val_f1_score: 0.8573 -
val_loss: 1.0980 - val_precision: 0.8682 - val_recall: 0.8497
Epoch 6/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 198s 565ms/step - accuracy: 0.8381 - auc: 0.9795 - f1_score: 0.8387 - loss: 1.
1190 - precision: 0.8553 - recall: 0.8207 - val_accuracy: 0.8404 - val_auc: 0.9792 - val_f1_score: 0.8383 -
val_loss: 1.1093 - val_precision: 0.8490 - val_recall: 0.8315
Epoch 7/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 200s 569ms/step - accuracy: 0.8399 - auc: 0.9808 - f1_score: 0.8403 - loss: 1.
0824 - precision: 0.8589 - recall: 0.8204 - val_accuracy: 0.8554 - val_auc: 0.9818 - val_f1_score: 0.8539 -
val_loss: 1.0408 - val_precision: 0.8649 - val_recall: 0.8465
Epoch 8/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 198s 564ms/step - accuracy: 0.8519 - auc: 0.9819 - f1_score: 0.8525 - loss: 1.
0378 - precision: 0.8693 - recall: 0.8339 - val_accuracy: 0.8568 - val_auc: 0.9818 - val_f1_score: 0.8580 -
val_loss: 1.0212 - val_precision: 0.8656 - val_recall: 0.8489
Epoch 9/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 198s 565ms/step - accuracy: 0.8541 - auc: 0.9827 - f1_score: 0.8546 - loss: 1.
0012 - precision: 0.8730 - recall: 0.8371 - val_accuracy: 0.8607 - val_auc: 0.9828 - val_f1_score: 0.8613 -
val_loss: 0.9714 - val_precision: 0.8704 - val_recall: 0.8539
Epoch 10/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 200s 571ms/step - accuracy: 0.8606 - auc: 0.9841 - f1_score: 0.8612 - loss: 0.
9610 - precision: 0.8771 - recall: 0.8414 - val_accuracy: 0.8429 - val_auc: 0.9814 - val_f1_score: 0.8405 -
val_loss: 0.9934 - val_precision: 0.8518 - val_recall: 0.8354
Epoch 11/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 575ms/step - accuracy: 0.8658 - auc: 0.9847 - f1_score: 0.8662 - loss: 0.
```

```
9286 - precision: 0.8826 - recall: 0.8483 - val_accuracy: 0.8283 - val_auc: 0.9783 - val_f1_score: 0.8223 -
val_loss: 1.0206 - val_precision: 0.8405 - val_recall: 0.8204
Epoch 12/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.8658 - auc: 0.9847 - f1_score: 0.8663 - loss: 0.
9032 - precision: 0.8818 - recall: 0.8508 - val_accuracy: 0.8440 - val_auc: 0.9793 - val_f1_score: 0.8425 -
val_loss: 0.9687 - val_precision: 0.8534 - val_recall: 0.8358
Epoch 13/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.8662 - auc: 0.9864 - f1_score: 0.8665 - loss: 0.
8655 - precision: 0.8823 - recall: 0.8527 - val_accuracy: 0.8354 - val_auc: 0.9800 - val_f1_score: 0.8337 -
val_loss: 0.9491 - val_precision: 0.8471 - val_recall: 0.8290
Epoch 14/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 574ms/step - accuracy: 0.8689 - auc: 0.9870 - f1_score: 0.8694 - loss: 0.
8336 - precision: 0.8834 - recall: 0.8541 - val_accuracy: 0.8564 - val_auc: 0.9840 - val_f1_score: 0.8572 -
val_loss: 0.8537 - val_precision: 0.8691 - val_recall: 0.8493
Epoch 15/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.8789 - auc: 0.9879 - f1_score: 0.8796 - loss: 0.
8026 - precision: 0.8917 - recall: 0.8621 - val_accuracy: 0.8596 - val_auc: 0.9829 - val_f1_score: 0.8596 -
val_loss: 0.8562 - val_precision: 0.8691 - val_recall: 0.8493
Epoch 16/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 580ms/step - accuracy: 0.8750 - auc: 0.9879 - f1_score: 0.8756 - loss: 0.
7841 - precision: 0.8949 - recall: 0.8614 - val_accuracy: 0.8400 - val_auc: 0.9814 - val_f1_score: 0.8381 -
val_loss: 0.8769 - val_precision: 0.8483 - val_recall: 0.8304
Epoch 17/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 579ms/step - accuracy: 0.8835 - auc: 0.9884 - f1_score: 0.8841 - loss: 0.
7550 - precision: 0.8969 - recall: 0.8689 - val_accuracy: 0.8465 - val_auc: 0.9813 - val_f1_score: 0.8466 -
val_loss: 0.8486 - val_precision: 0.8596 - val_recall: 0.8400
Epoch 18/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 578ms/step - accuracy: 0.8803 - auc: 0.9886 - f1_score: 0.8808 - loss: 0.
7382 - precision: 0.8936 - recall: 0.8647 - val_accuracy: 0.8571 - val_auc: 0.9832 - val_f1_score: 0.8580 -
val_loss: 0.7976 - val_precision: 0.8686 - val_recall: 0.8500
Epoch 19/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 576ms/step - accuracy: 0.8852 - auc: 0.9896 - f1_score: 0.8857 - loss: 0.
7091 - precision: 0.9002 - recall: 0.8715 - val_accuracy: 0.8457 - val_auc: 0.9818 - val_f1_score: 0.8434 -
val_loss: 0.8097 - val_precision: 0.8575 - val_recall: 0.8383
Epoch 20/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 574ms/step - accuracy: 0.8794 - auc: 0.9885 - f1_score: 0.8799 - loss: 0.
7092 - precision: 0.8947 - recall: 0.8643 - val_accuracy: 0.8546 - val_auc: 0.9826 - val_f1_score: 0.8536 -
val_loss: 0.7765 - val_precision: 0.8662 - val_recall: 0.8461
Epoch 21/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 574ms/step - accuracy: 0.8842 - auc: 0.9898 - f1_score: 0.8848 - loss: 0.
6788 - precision: 0.8977 - recall: 0.8699 - val_accuracy: 0.8579 - val_auc: 0.9831 - val_f1_score: 0.8578 -
val_loss: 0.7630 - val_precision: 0.8689 - val_recall: 0.8500
```

```
Epoch 22/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 572ms/step - accuracy: 0.8895 - auc: 0.9903 - f1_score: 0.8899 - loss: 0.
6546 - precision: 0.9028 - recall: 0.8755 - val_accuracy: 0.8589 - val_auc: 0.9829 - val_f1_score: 0.8579 -
val_loss: 0.7614 - val_precision: 0.8693 - val_recall: 0.8529
Epoch 23/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 200s 571ms/step - accuracy: 0.8933 - auc: 0.9899 - f1_score: 0.8937 - loss: 0.
6478 - precision: 0.9075 - recall: 0.8796 - val_accuracy: 0.8351 - val_auc: 0.9802 - val_f1_score: 0.8311 -
val_loss: 0.8093 - val_precision: 0.8429 - val_recall: 0.8294
Epoch 24/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.8842 - auc: 0.9896 - f1_score: 0.8846 - loss: 0.
6440 - precision: 0.8967 - recall: 0.8678 - val_accuracy: 0.8486 - val_auc: 0.9826 - val_f1_score: 0.8483 -
val_loss: 0.7390 - val_precision: 0.8594 - val_recall: 0.8408
Epoch 25/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 200s 571ms/step - accuracy: 0.8933 - auc: 0.9908 - f1_score: 0.8940 - loss: 0.
6183 - precision: 0.9057 - recall: 0.8810 - val_accuracy: 0.8500 - val_auc: 0.9832 - val_f1_score: 0.8486 -
val_loss: 0.7333 - val_precision: 0.8588 - val_recall: 0.8454
Epoch 26/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.8919 - auc: 0.9905 - f1_score: 0.8924 - loss: 0.
6081 - precision: 0.9034 - recall: 0.8774 - val_accuracy: 0.8432 - val_auc: 0.9782 - val_f1_score: 0.8433 -
val_loss: 0.7699 - val_precision: 0.8517 - val_recall: 0.8390
Epoch 27/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.8922 - auc: 0.9911 - f1_score: 0.8925 - loss: 0.
5945 - precision: 0.9066 - recall: 0.8811 - val_accuracy: 0.8564 - val_auc: 0.9832 - val_f1_score: 0.8570 -
val_loss: 0.7003 - val_precision: 0.8665 - val_recall: 0.8486
Epoch 28/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 574ms/step - accuracy: 0.8991 - auc: 0.9916 - f1_score: 0.8995 - loss: 0.
5767 - precision: 0.9124 - recall: 0.8869 - val_accuracy: 0.8678 - val_auc: 0.9845 - val_f1_score: 0.8691 -
val_loss: 0.6680 - val_precision: 0.8761 - val_recall: 0.8614
Epoch 29/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 572ms/step - accuracy: 0.9013 - auc: 0.9921 - f1_score: 0.9017 - loss: 0.
5593 - precision: 0.9101 - recall: 0.8881 - val_accuracy: 0.8571 - val_auc: 0.9836 - val_f1_score: 0.8573 -
val_loss: 0.6793 - val_precision: 0.8695 - val_recall: 0.8518
Epoch 30/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 572ms/step - accuracy: 0.8959 - auc: 0.9920 - f1_score: 0.8965 - loss: 0.
5517 - precision: 0.9080 - recall: 0.8831 - val_accuracy: 0.8500 - val_auc: 0.9826 - val_f1_score: 0.8518 -
val_loss: 0.6795 - val_precision: 0.8589 - val_recall: 0.8432
Epoch 31/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.9031 - auc: 0.9922 - f1_score: 0.9035 - loss: 0.
5419 - precision: 0.9151 - recall: 0.8912 - val_accuracy: 0.8475 - val_auc: 0.9806 - val_f1_score: 0.8481 -
val_loss: 0.7062 - val_precision: 0.8560 - val_recall: 0.8425
Epoch 32/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.8992 - auc: 0.9920 - f1_score: 0.8996 - loss: 0.
```

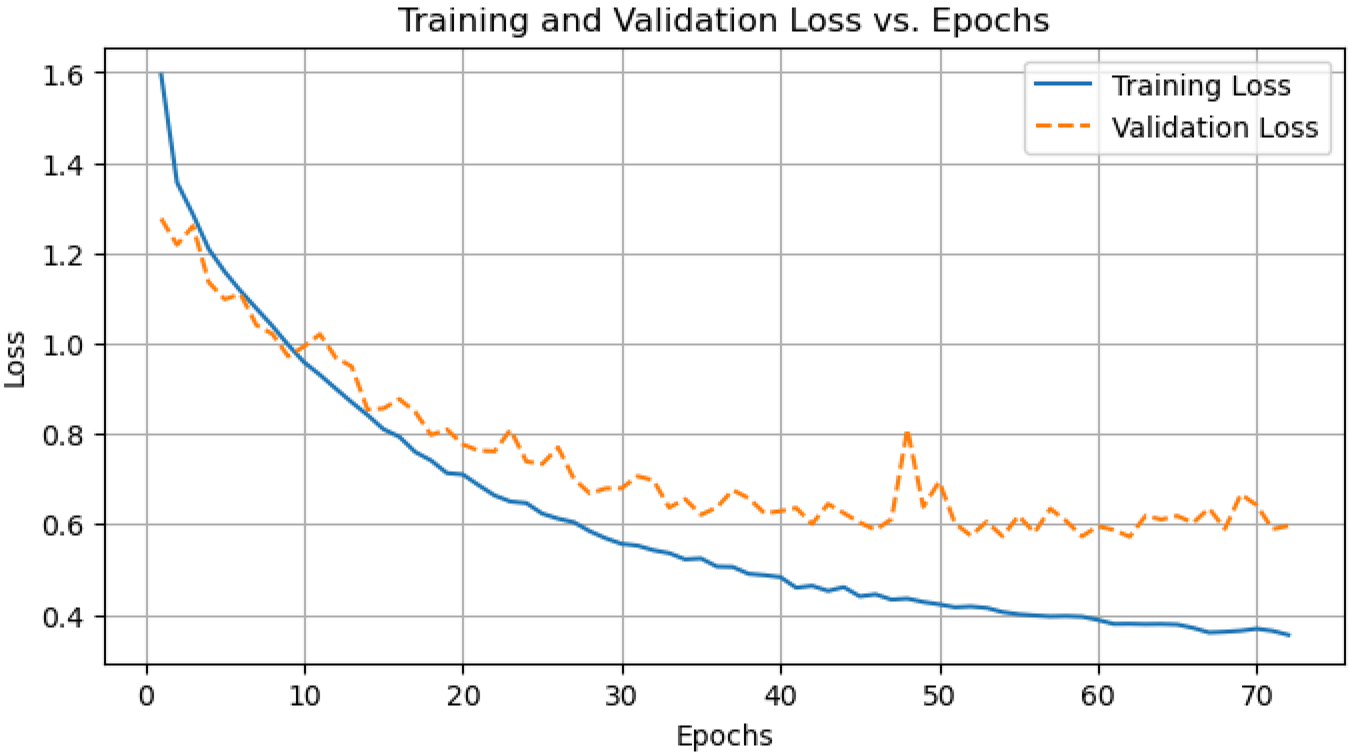
```
5400 - precision: 0.9099 - recall: 0.8880 - val_accuracy: 0.8418 - val_auc: 0.9812 - val_f1_score: 0.8407 -
val_loss: 0.6975 - val_precision: 0.8521 - val_recall: 0.8354
Epoch 33/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 573ms/step - accuracy: 0.9018 - auc: 0.9922 - f1_score: 0.9025 - loss: 0.
5283 - precision: 0.9153 - recall: 0.8911 - val_accuracy: 0.8586 - val_auc: 0.9834 - val_f1_score: 0.8615 -
val_loss: 0.6369 - val_precision: 0.8680 - val_recall: 0.8529
Epoch 34/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 574ms/step - accuracy: 0.9063 - auc: 0.9923 - f1_score: 0.9069 - loss: 0.
5143 - precision: 0.9166 - recall: 0.8929 - val_accuracy: 0.8518 - val_auc: 0.9830 - val_f1_score: 0.8532 -
val_loss: 0.6557 - val_precision: 0.8630 - val_recall: 0.8436
Epoch 35/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 574ms/step - accuracy: 0.9030 - auc: 0.9917 - f1_score: 0.9034 - loss: 0.
5197 - precision: 0.9141 - recall: 0.8897 - val_accuracy: 0.8650 - val_auc: 0.9843 - val_f1_score: 0.8666 -
val_loss: 0.6210 - val_precision: 0.8756 - val_recall: 0.8554
Epoch 36/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 575ms/step - accuracy: 0.9045 - auc: 0.9927 - f1_score: 0.9048 - loss: 0.
5024 - precision: 0.9160 - recall: 0.8924 - val_accuracy: 0.8571 - val_auc: 0.9833 - val_f1_score: 0.8598 -
val_loss: 0.6371 - val_precision: 0.8623 - val_recall: 0.8525
Epoch 37/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 201s 574ms/step - accuracy: 0.8993 - auc: 0.9918 - f1_score: 0.8998 - loss: 0.
5081 - precision: 0.9105 - recall: 0.8861 - val_accuracy: 0.8425 - val_auc: 0.9807 - val_f1_score: 0.8412 -
val_loss: 0.6756 - val_precision: 0.8509 - val_recall: 0.8379
Epoch 38/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 576ms/step - accuracy: 0.9100 - auc: 0.9930 - f1_score: 0.9105 - loss: 0.
4814 - precision: 0.9190 - recall: 0.8961 - val_accuracy: 0.8486 - val_auc: 0.9817 - val_f1_score: 0.8481 -
val_loss: 0.6579 - val_precision: 0.8581 - val_recall: 0.8443
Epoch 39/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 575ms/step - accuracy: 0.9035 - auc: 0.9928 - f1_score: 0.9039 - loss: 0.
4851 - precision: 0.9132 - recall: 0.8886 - val_accuracy: 0.8600 - val_auc: 0.9827 - val_f1_score: 0.8610 -
val_loss: 0.6244 - val_precision: 0.8689 - val_recall: 0.8550
Epoch 40/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 577ms/step - accuracy: 0.9044 - auc: 0.9929 - f1_score: 0.9052 - loss: 0.
4800 - precision: 0.9158 - recall: 0.8955 - val_accuracy: 0.8557 - val_auc: 0.9826 - val_f1_score: 0.8575 -
val_loss: 0.6290 - val_precision: 0.8630 - val_recall: 0.8507
Epoch 41/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9154 - auc: 0.9936 - f1_score: 0.9159 - loss: 0.
4622 - precision: 0.9240 - recall: 0.9045 - val_accuracy: 0.8564 - val_auc: 0.9815 - val_f1_score: 0.8568 -
val_loss: 0.6359 - val_precision: 0.8640 - val_recall: 0.8507
Epoch 42/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 580ms/step - accuracy: 0.9106 - auc: 0.9928 - f1_score: 0.9108 - loss: 0.
4689 - precision: 0.9206 - recall: 0.8983 - val_accuracy: 0.8564 - val_auc: 0.9835 - val_f1_score: 0.8591 -
val_loss: 0.6006 - val_precision: 0.8647 - val_recall: 0.8518
```

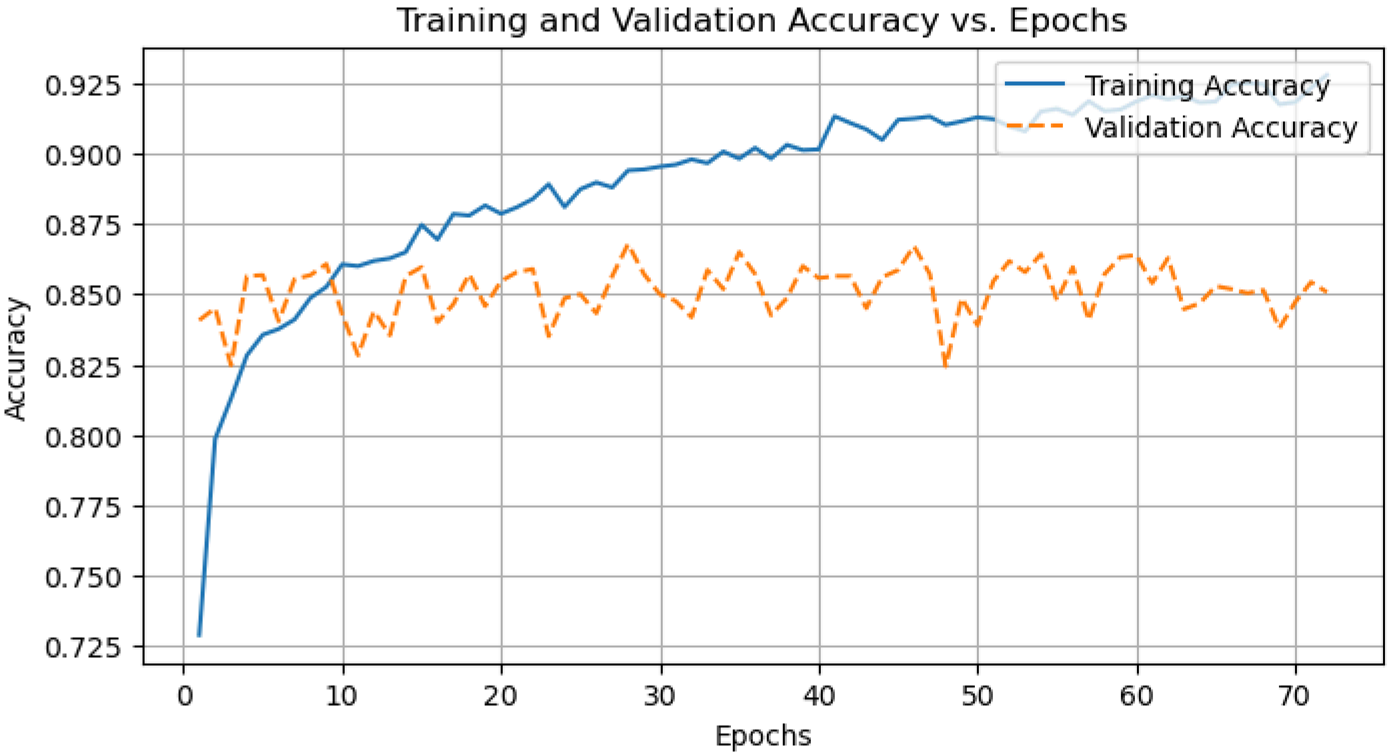
```
Epoch 43/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9115 - auc: 0.9943 - f1_score: 0.9120 - loss: 0.
4434 - precision: 0.9213 - recall: 0.9022 - val_accuracy: 0.8450 - val_auc: 0.9804 - val_f1_score: 0.8455 -
val_loss: 0.6449 - val_precision: 0.8519 - val_recall: 0.8400
Epoch 44/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 578ms/step - accuracy: 0.9078 - auc: 0.9928 - f1_score: 0.9083 - loss: 0.
4563 - precision: 0.9192 - recall: 0.8974 - val_accuracy: 0.8561 - val_auc: 0.9821 - val_f1_score: 0.8569 -
val_loss: 0.6246 - val_precision: 0.8625 - val_recall: 0.8514
Epoch 45/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 577ms/step - accuracy: 0.9139 - auc: 0.9944 - f1_score: 0.9146 - loss: 0.
4297 - precision: 0.9218 - recall: 0.9039 - val_accuracy: 0.8586 - val_auc: 0.9827 - val_f1_score: 0.8602 -
val_loss: 0.6037 - val_precision: 0.8662 - val_recall: 0.8532
Epoch 46/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 205s 584ms/step - accuracy: 0.9140 - auc: 0.9935 - f1_score: 0.9143 - loss: 0.
4381 - precision: 0.9209 - recall: 0.9056 - val_accuracy: 0.8671 - val_auc: 0.9829 - val_f1_score: 0.8687 -
val_loss: 0.5875 - val_precision: 0.8757 - val_recall: 0.8611
Epoch 47/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 580ms/step - accuracy: 0.9141 - auc: 0.9945 - f1_score: 0.9145 - loss: 0.
4257 - precision: 0.9233 - recall: 0.9031 - val_accuracy: 0.8571 - val_auc: 0.9824 - val_f1_score: 0.8589 -
val_loss: 0.6114 - val_precision: 0.8637 - val_recall: 0.8507
Epoch 48/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 206s 586ms/step - accuracy: 0.9137 - auc: 0.9939 - f1_score: 0.9141 - loss: 0.
4291 - precision: 0.9251 - recall: 0.9011 - val_accuracy: 0.8244 - val_auc: 0.9671 - val_f1_score: 0.8190 -
val_loss: 0.8108 - val_precision: 0.8304 - val_recall: 0.8183
Epoch 49/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 205s 585ms/step - accuracy: 0.9114 - auc: 0.9938 - f1_score: 0.9120 - loss: 0.
4296 - precision: 0.9222 - recall: 0.9004 - val_accuracy: 0.8486 - val_auc: 0.9804 - val_f1_score: 0.8490 -
val_loss: 0.6384 - val_precision: 0.8571 - val_recall: 0.8440
Epoch 50/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 583ms/step - accuracy: 0.9192 - auc: 0.9940 - f1_score: 0.9194 - loss: 0.
4173 - precision: 0.9290 - recall: 0.9078 - val_accuracy: 0.8390 - val_auc: 0.9752 - val_f1_score: 0.8399 -
val_loss: 0.6945 - val_precision: 0.8469 - val_recall: 0.8354
Epoch 51/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 207s 589ms/step - accuracy: 0.9146 - auc: 0.9945 - f1_score: 0.9151 - loss: 0.
4095 - precision: 0.9242 - recall: 0.9064 - val_accuracy: 0.8546 - val_auc: 0.9825 - val_f1_score: 0.8557 -
val_loss: 0.6016 - val_precision: 0.8628 - val_recall: 0.8472
Epoch 52/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 207s 591ms/step - accuracy: 0.9136 - auc: 0.9945 - f1_score: 0.9139 - loss: 0.
4101 - precision: 0.9224 - recall: 0.9031 - val_accuracy: 0.8618 - val_auc: 0.9835 - val_f1_score: 0.8645 -
val_loss: 0.5748 - val_precision: 0.8714 - val_recall: 0.8568
Epoch 53/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 205s 585ms/step - accuracy: 0.9093 - auc: 0.9944 - f1_score: 0.9097 - loss: 0.
```

```
4084 - precision: 0.9227 - recall: 0.9000 - val_accuracy: 0.8579 - val_auc: 0.9805 - val_f1_score: 0.8585 -
val_loss: 0.6056 - val_precision: 0.8647 - val_recall: 0.8539
Epoch 54/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 205s 584ms/step - accuracy: 0.9213 - auc: 0.9949 - f1_score: 0.9217 - loss: 0.
3942 - precision: 0.9312 - recall: 0.9104 - val_accuracy: 0.8643 - val_auc: 0.9826 - val_f1_score: 0.8664 -
val_loss: 0.5731 - val_precision: 0.8710 - val_recall: 0.8586
Epoch 55/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 578ms/step - accuracy: 0.9150 - auc: 0.9947 - f1_score: 0.9155 - loss: 0.
3978 - precision: 0.9259 - recall: 0.9055 - val_accuracy: 0.8479 - val_auc: 0.9799 - val_f1_score: 0.8477 -
val_loss: 0.6189 - val_precision: 0.8565 - val_recall: 0.8443
Epoch 56/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 578ms/step - accuracy: 0.9191 - auc: 0.9950 - f1_score: 0.9194 - loss: 0.
3875 - precision: 0.9285 - recall: 0.9083 - val_accuracy: 0.8596 - val_auc: 0.9813 - val_f1_score: 0.8608 -
val_loss: 0.5810 - val_precision: 0.8678 - val_recall: 0.8539
Epoch 57/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 578ms/step - accuracy: 0.9223 - auc: 0.9950 - f1_score: 0.9227 - loss: 0.
3836 - precision: 0.9312 - recall: 0.9130 - val_accuracy: 0.8408 - val_auc: 0.9787 - val_f1_score: 0.8411 -
val_loss: 0.6344 - val_precision: 0.8477 - val_recall: 0.8365
Epoch 58/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 577ms/step - accuracy: 0.9206 - auc: 0.9947 - f1_score: 0.9212 - loss: 0.
3851 - precision: 0.9301 - recall: 0.9116 - val_accuracy: 0.8571 - val_auc: 0.9810 - val_f1_score: 0.8572 -
val_loss: 0.6073 - val_precision: 0.8639 - val_recall: 0.8500
Epoch 59/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 579ms/step - accuracy: 0.9171 - auc: 0.9943 - f1_score: 0.9175 - loss: 0.
3939 - precision: 0.9272 - recall: 0.9070 - val_accuracy: 0.8632 - val_auc: 0.9832 - val_f1_score: 0.8648 -
val_loss: 0.5730 - val_precision: 0.8731 - val_recall: 0.8579
Epoch 60/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 580ms/step - accuracy: 0.9192 - auc: 0.9945 - f1_score: 0.9195 - loss: 0.
3881 - precision: 0.9279 - recall: 0.9105 - val_accuracy: 0.8639 - val_auc: 0.9805 - val_f1_score: 0.8651 -
val_loss: 0.5957 - val_precision: 0.8703 - val_recall: 0.8579
Epoch 61/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 580ms/step - accuracy: 0.9241 - auc: 0.9953 - f1_score: 0.9244 - loss: 0.
3729 - precision: 0.9315 - recall: 0.9141 - val_accuracy: 0.8539 - val_auc: 0.9804 - val_f1_score: 0.8548 -
val_loss: 0.5873 - val_precision: 0.8616 - val_recall: 0.8450
Epoch 62/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9208 - auc: 0.9948 - f1_score: 0.9211 - loss: 0.
3782 - precision: 0.9277 - recall: 0.9136 - val_accuracy: 0.8628 - val_auc: 0.9815 - val_f1_score: 0.8640 -
val_loss: 0.5725 - val_precision: 0.8703 - val_recall: 0.8561
Epoch 63/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 582ms/step - accuracy: 0.9235 - auc: 0.9953 - f1_score: 0.9238 - loss: 0.
3713 - precision: 0.9312 - recall: 0.9146 - val_accuracy: 0.8447 - val_auc: 0.9790 - val_f1_score: 0.8462 -
val_loss: 0.6185 - val_precision: 0.8523 - val_recall: 0.8386
```

```
Epoch 64/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 580ms/step - accuracy: 0.9190 - auc: 0.9952 - f1_score: 0.9195 - loss: 0.
3726 - precision: 0.9262 - recall: 0.9093 - val_accuracy: 0.8468 - val_auc: 0.9793 - val_f1_score: 0.8478 -
val_loss: 0.6105 - val_precision: 0.8546 - val_recall: 0.8436
Epoch 65/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 202s 577ms/step - accuracy: 0.9179 - auc: 0.9946 - f1_score: 0.9184 - loss: 0.
3749 - precision: 0.9257 - recall: 0.9088 - val_accuracy: 0.8529 - val_auc: 0.9788 - val_f1_score: 0.8525 -
val_loss: 0.6184 - val_precision: 0.8617 - val_recall: 0.8432
Epoch 66/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 579ms/step - accuracy: 0.9302 - auc: 0.9954 - f1_score: 0.9305 - loss: 0.
3641 - precision: 0.9367 - recall: 0.9192 - val_accuracy: 0.8518 - val_auc: 0.9783 - val_f1_score: 0.8528 -
val_loss: 0.6032 - val_precision: 0.8584 - val_recall: 0.8443
Epoch 67/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9243 - auc: 0.9955 - f1_score: 0.9247 - loss: 0.
3617 - precision: 0.9317 - recall: 0.9156 - val_accuracy: 0.8504 - val_auc: 0.9781 - val_f1_score: 0.8494 -
val_loss: 0.6352 - val_precision: 0.8580 - val_recall: 0.8436
Epoch 68/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 203s 579ms/step - accuracy: 0.9248 - auc: 0.9952 - f1_score: 0.9250 - loss: 0.
3639 - precision: 0.9317 - recall: 0.9162 - val_accuracy: 0.8514 - val_auc: 0.9792 - val_f1_score: 0.8534 -
val_loss: 0.5898 - val_precision: 0.8612 - val_recall: 0.8468
Epoch 69/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9172 - auc: 0.9952 - f1_score: 0.9177 - loss: 0.
3627 - precision: 0.9259 - recall: 0.9096 - val_accuracy: 0.8379 - val_auc: 0.9750 - val_f1_score: 0.8385 -
val_loss: 0.6666 - val_precision: 0.8441 - val_recall: 0.8354
Epoch 70/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9230 - auc: 0.9950 - f1_score: 0.9234 - loss: 0.
3622 - precision: 0.9291 - recall: 0.9158 - val_accuracy: 0.8472 - val_auc: 0.9763 - val_f1_score: 0.8480 -
val_loss: 0.6422 - val_precision: 0.8521 - val_recall: 0.8415
Epoch 71/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9284 - auc: 0.9954 - f1_score: 0.9288 - loss: 0.
3507 - precision: 0.9354 - recall: 0.9200 - val_accuracy: 0.8543 - val_auc: 0.9801 - val_f1_score: 0.8567 -
val_loss: 0.5893 - val_precision: 0.8581 - val_recall: 0.8489
Epoch 72/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 204s 581ms/step - accuracy: 0.9292 - auc: 0.9956 - f1_score: 0.9294 - loss: 0.
3510 - precision: 0.9342 - recall: 0.9206 - val_accuracy: 0.8507 - val_auc: 0.9789 - val_f1_score: 0.8538 -
val_loss: 0.5965 - val_precision: 0.8562 - val_recall: 0.8440
Epoch 72: early stopping
Restoring model weights from the end of the best epoch: 62.
```

```
In [27]: plotting_the_curves(history_res101)
```


Training and Validation Loss vs. Epochs
Training Loss
Validation Loss
Loss
1.6
1.4
1.2
1.0
0.8
0.6
0.4
Epochs
0
10
20
30
40
50
60
70



In [28]:

```
test_loss_res101, test_accuracy_res101, precision_res101, recall_res101, f1_res101, auc_res101 = get_test_
```

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 43s 439ms/step - accuracy: 0.8770 - auc: 0.9847 - f1_score: 0.5287 - loss: 0.512
5 - precision: 0.8837 - recall: 0.8701
Test Loss ResNet101: 0.5677
Test Accuracy ResNet101: 0.8637
94/94 ━━━━━━━━━━━━━━━━━━━━ 43s 450ms/step

----------For ResNet101 Model----------

      Metric     Score
0  Precision  0.863195
1     Recall  0.863667
2   F1 Score  0.861547
3        AUC  0.982191

Classification Report:
              precision    recall  f1-score   support

   buildings       0.89      0.89      0.89       437
      forest       0.98      0.98      0.98       474
     glacier       0.75      0.79      0.77       553
    mountain       0.82      0.66      0.73       525
         sea       0.88      0.97      0.92       510
      street       0.88      0.91      0.90       501

    accuracy                           0.86      3000
   macro avg       0.87      0.87      0.87      3000
weighted avg       0.86      0.86      0.86      3000
```

EfficientNetB0

In [97]:

```python
model_name_50 = "EfficientNetB0"
training_images, training_labels, class_names = getting_paths_and_labels(train_dir, image_size, model_name_
test_images, test_labels, class_names = getting_paths_and_labels(test_dir, image_size, model_name_50)
```

In [99]:

```python
encoder = LabelEncoder()
training_labels_encoded = encoder.fit_transform(training_labels)
test_labels_encoded = encoder.fit_transform(test_labels)
```

In [101…

```python
xTrain, yTrain, xTest, yTest = train_test_split(
    training_images, training_labels_encoded, test_size=0.2, random_state=42, stratify=training_labels_enc
```

In [102…

```
training_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(xTest, num_classes=6)
validation_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(yTest, num_classes=6)
test_labels_one_hot_encoding = tf.keras.utils.to_categorical(test_labels_encoded, num_classes=6)
```

In [103…

```
train_ds = tf.data.Dataset.from_tensor_slices((xTrain, training_labels_with_one_hot_encoding)).batch(batch_

val_ds = tf.data.Dataset.from_tensor_slices((yTrain, validation_labels_with_one_hot_encoding)).batch(batch_
```

In [106…

```
model_EnB0 = model_with_layers("EfficientNetB0")
```

```
Model: "sequential_4"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ data_augmentation (Sequential)  │ (None, 150, 150, 3)    │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ efficientnetb0 (Functional)     │ (None, 1280)           │     4,049,571 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten_4 (Flatten)             │ (None, 1280)           │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_8 (Dense)                 │ (None, 512)            │       655,872 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_4           │ (None, 512)            │         2,048 │
│ (BatchNormalization)            │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_4 (Dropout)             │ (None, 512)            │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_9 (Dense)                 │ (None, 6)              │         3,078 │
└─────────────────────────────────┴────────────────────────┴───────────────┘

 Total params: 4,710,569 (17.97 MB)

 Trainable params: 659,974 (2.52 MB)

 Non-trainable params: 4,050,595 (15.45 MB)
```

In [109…

```
history_EnB0 = get_history_and_fit_model(model_EnB0)
```

```
Epoch 1/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 47s 121ms/step - accuracy: 0.6880 - auc: 0.9149 - f1_score: 0.6876 - loss: 1.6
412 - precision: 0.7420 - recall: 0.6294 - val_accuracy: 0.8860 - val_auc: 0.9885 - val_f1_score: 0.8869 -
val_loss: 1.0467 - val_precision: 0.9003 - val_recall: 0.8685
Epoch 2/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.8345 - auc: 0.9781 - f1_score: 0.8350 - loss: 1.1
763 - precision: 0.8478 - recall: 0.8172 - val_accuracy: 0.8874 - val_auc: 0.9896 - val_f1_score: 0.8880 -
val_loss: 1.0081 - val_precision: 0.8957 - val_recall: 0.8810
Epoch 3/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.8640 - auc: 0.9837 - f1_score: 0.8645 - loss: 1.0
779 - precision: 0.8782 - recall: 0.8522 - val_accuracy: 0.8945 - val_auc: 0.9907 - val_f1_score: 0.8956 -
val_loss: 0.9675 - val_precision: 0.9053 - val_recall: 0.8892
Epoch 4/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.8681 - auc: 0.9854 - f1_score: 0.8685 - loss: 1.0
353 - precision: 0.8804 - recall: 0.8581 - val_accuracy: 0.8963 - val_auc: 0.9907 - val_f1_score: 0.8978 -
val_loss: 0.9428 - val_precision: 0.9034 - val_recall: 0.8892
Epoch 5/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.8720 - auc: 0.9858 - f1_score: 0.8727 - loss: 1.0
041 - precision: 0.8840 - recall: 0.8629 - val_accuracy: 0.8953 - val_auc: 0.9905 - val_f1_score: 0.8966 -
val_loss: 0.9197 - val_precision: 0.9025 - val_recall: 0.8903
Epoch 6/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.8828 - auc: 0.9879 - f1_score: 0.8837 - loss: 0.9
634 - precision: 0.8930 - recall: 0.8751 - val_accuracy: 0.9020 - val_auc: 0.9910 - val_f1_score: 0.9031 -
val_loss: 0.8939 - val_precision: 0.9088 - val_recall: 0.8942
Epoch 7/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.8797 - auc: 0.9879 - f1_score: 0.8800 - loss: 0.9
382 - precision: 0.8937 - recall: 0.8706 - val_accuracy: 0.8928 - val_auc: 0.9908 - val_f1_score: 0.8944 -
val_loss: 0.8729 - val_precision: 0.9011 - val_recall: 0.8860
Epoch 8/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.8893 - auc: 0.9901 - f1_score: 0.8900 - loss: 0.8
835 - precision: 0.9003 - recall: 0.8792 - val_accuracy: 0.8960 - val_auc: 0.9903 - val_f1_score: 0.8973 -
val_loss: 0.8631 - val_precision: 0.8988 - val_recall: 0.8888
Epoch 9/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.8923 - auc: 0.9905 - f1_score: 0.8930 - loss: 0.8
553 - precision: 0.9016 - recall: 0.8817 - val_accuracy: 0.9002 - val_auc: 0.9907 - val_f1_score: 0.9019 -
val_loss: 0.8250 - val_precision: 0.9054 - val_recall: 0.8935
Epoch 10/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 118ms/step - accuracy: 0.8985 - auc: 0.9909 - f1_score: 0.8990 - loss: 0.8
300 - precision: 0.9094 - recall: 0.8887 - val_accuracy: 0.9052 - val_auc: 0.9909 - val_f1_score: 0.9066 -
val_loss: 0.8037 - val_precision: 0.9098 - val_recall: 0.8981
Epoch 11/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 41s 116ms/step - accuracy: 0.8977 - auc: 0.9916 - f1_score: 0.8982 - loss: 0.7
```

```
995 - precision: 0.9076 - recall: 0.8897 - val_accuracy: 0.9027 - val_auc: 0.9908 - val_f1_score: 0.9038 -
val_loss: 0.7832 - val_precision: 0.9083 - val_recall: 0.8995
Epoch 12/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 41s 118ms/step - accuracy: 0.9090 - auc: 0.9928 - f1_score: 0.9097 - loss: 0.7
560 - precision: 0.9157 - recall: 0.9025 - val_accuracy: 0.9013 - val_auc: 0.9909 - val_f1_score: 0.9025 -
val_loss: 0.7636 - val_precision: 0.9065 - val_recall: 0.8978
Epoch 13/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 41s 118ms/step - accuracy: 0.9088 - auc: 0.9924 - f1_score: 0.9098 - loss: 0.7
427 - precision: 0.9188 - recall: 0.8991 - val_accuracy: 0.9006 - val_auc: 0.9913 - val_f1_score: 0.9021 -
val_loss: 0.7440 - val_precision: 0.9077 - val_recall: 0.8967
Epoch 14/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9085 - auc: 0.9928 - f1_score: 0.9093 - loss: 0.7
185 - precision: 0.9196 - recall: 0.9003 - val_accuracy: 0.9020 - val_auc: 0.9905 - val_f1_score: 0.9038 -
val_loss: 0.7355 - val_precision: 0.9076 - val_recall: 0.8995
Epoch 15/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9099 - auc: 0.9929 - f1_score: 0.9109 - loss: 0.7
007 - precision: 0.9183 - recall: 0.9033 - val_accuracy: 0.9049 - val_auc: 0.9913 - val_f1_score: 0.9063 -
val_loss: 0.7086 - val_precision: 0.9114 - val_recall: 0.9013
Epoch 16/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 41s 118ms/step - accuracy: 0.9120 - auc: 0.9936 - f1_score: 0.9127 - loss: 0.6
758 - precision: 0.9195 - recall: 0.9062 - val_accuracy: 0.9059 - val_auc: 0.9909 - val_f1_score: 0.9074 -
val_loss: 0.6963 - val_precision: 0.9114 - val_recall: 0.9013
Epoch 17/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9149 - auc: 0.9940 - f1_score: 0.9157 - loss: 0.6
514 - precision: 0.9228 - recall: 0.9058 - val_accuracy: 0.9070 - val_auc: 0.9912 - val_f1_score: 0.9083 -
val_loss: 0.6822 - val_precision: 0.9118 - val_recall: 0.9020
Epoch 18/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9178 - auc: 0.9940 - f1_score: 0.9184 - loss: 0.6
338 - precision: 0.9246 - recall: 0.9121 - val_accuracy: 0.9027 - val_auc: 0.9909 - val_f1_score: 0.9042 -
val_loss: 0.6647 - val_precision: 0.9083 - val_recall: 0.9002
Epoch 19/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9165 - auc: 0.9946 - f1_score: 0.9174 - loss: 0.6
087 - precision: 0.9224 - recall: 0.9093 - val_accuracy: 0.9038 - val_auc: 0.9912 - val_f1_score: 0.9052 -
val_loss: 0.6469 - val_precision: 0.9090 - val_recall: 0.8999
Epoch 20/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 41s 118ms/step - accuracy: 0.9188 - auc: 0.9947 - f1_score: 0.9193 - loss: 0.5
894 - precision: 0.9255 - recall: 0.9149 - val_accuracy: 0.9074 - val_auc: 0.9913 - val_f1_score: 0.9085 -
val_loss: 0.6311 - val_precision: 0.9127 - val_recall: 0.9049
Epoch 21/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9170 - auc: 0.9942 - f1_score: 0.9178 - loss: 0.5
926 - precision: 0.9239 - recall: 0.9087 - val_accuracy: 0.9067 - val_auc: 0.9911 - val_f1_score: 0.9077 -
val_loss: 0.6208 - val_precision: 0.9118 - val_recall: 0.9024
```

```
Epoch 22/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9240 - auc: 0.9950 - f1_score: 0.9248 - loss: 0.5
672 - precision: 0.9307 - recall: 0.9167 - val_accuracy: 0.9099 - val_auc: 0.9915 - val_f1_score: 0.9112 -
val_loss: 0.6014 - val_precision: 0.9168 - val_recall: 0.9070
Epoch 23/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9274 - auc: 0.9950 - f1_score: 0.9283 - loss: 0.5
484 - precision: 0.9328 - recall: 0.9212 - val_accuracy: 0.9088 - val_auc: 0.9914 - val_f1_score: 0.9100 -
val_loss: 0.5951 - val_precision: 0.9134 - val_recall: 0.9059
Epoch 24/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9258 - auc: 0.9955 - f1_score: 0.9266 - loss: 0.5
326 - precision: 0.9316 - recall: 0.9189 - val_accuracy: 0.9056 - val_auc: 0.9905 - val_f1_score: 0.9069 -
val_loss: 0.5982 - val_precision: 0.9108 - val_recall: 0.9024
Epoch 25/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9266 - auc: 0.9954 - f1_score: 0.9272 - loss: 0.5
232 - precision: 0.9347 - recall: 0.9203 - val_accuracy: 0.9116 - val_auc: 0.9913 - val_f1_score: 0.9127 -
val_loss: 0.5705 - val_precision: 0.9189 - val_recall: 0.9081
Epoch 26/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9280 - auc: 0.9954 - f1_score: 0.9286 - loss: 0.5
099 - precision: 0.9326 - recall: 0.9202 - val_accuracy: 0.9095 - val_auc: 0.9914 - val_f1_score: 0.9108 -
val_loss: 0.5611 - val_precision: 0.9155 - val_recall: 0.9067
Epoch 27/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9286 - auc: 0.9957 - f1_score: 0.9292 - loss: 0.4
946 - precision: 0.9348 - recall: 0.9220 - val_accuracy: 0.9109 - val_auc: 0.9917 - val_f1_score: 0.9126 -
val_loss: 0.5549 - val_precision: 0.9162 - val_recall: 0.9074
Epoch 28/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9292 - auc: 0.9960 - f1_score: 0.9299 - loss: 0.4
787 - precision: 0.9346 - recall: 0.9224 - val_accuracy: 0.9070 - val_auc: 0.9916 - val_f1_score: 0.9083 -
val_loss: 0.5447 - val_precision: 0.9120 - val_recall: 0.9042
Epoch 29/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9261 - auc: 0.9958 - f1_score: 0.9267 - loss: 0.4
771 - precision: 0.9320 - recall: 0.9200 - val_accuracy: 0.9059 - val_auc: 0.9899 - val_f1_score: 0.9071 -
val_loss: 0.5593 - val_precision: 0.9106 - val_recall: 0.9038
Epoch 30/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9272 - auc: 0.9958 - f1_score: 0.9276 - loss: 0.4
677 - precision: 0.9326 - recall: 0.9213 - val_accuracy: 0.9088 - val_auc: 0.9908 - val_f1_score: 0.9102 -
val_loss: 0.5475 - val_precision: 0.9130 - val_recall: 0.9045
Epoch 31/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9318 - auc: 0.9957 - f1_score: 0.9322 - loss: 0.4
601 - precision: 0.9360 - recall: 0.9250 - val_accuracy: 0.9031 - val_auc: 0.9902 - val_f1_score: 0.9045 -
val_loss: 0.5468 - val_precision: 0.9079 - val_recall: 0.8988
Epoch 32/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9320 - auc: 0.9959 - f1_score: 0.9326 - loss: 0.4
```

```
528 - precision: 0.9376 - recall: 0.9254 - val_accuracy: 0.9084 - val_auc: 0.9917 - val_f1_score: 0.9096 -
val_loss: 0.5160 - val_precision: 0.9130 - val_recall: 0.9052
Epoch 33/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9396 - auc: 0.9966 - f1_score: 0.9404 - loss: 0.4
273 - precision: 0.9452 - recall: 0.9344 - val_accuracy: 0.9052 - val_auc: 0.9916 - val_f1_score: 0.9067 -
val_loss: 0.5101 - val_precision: 0.9109 - val_recall: 0.8995
Epoch 34/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9336 - auc: 0.9963 - f1_score: 0.9342 - loss: 0.4
306 - precision: 0.9385 - recall: 0.9282 - val_accuracy: 0.9109 - val_auc: 0.9914 - val_f1_score: 0.9126 -
val_loss: 0.5026 - val_precision: 0.9176 - val_recall: 0.9084
Epoch 35/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9357 - auc: 0.9967 - f1_score: 0.9360 - loss: 0.4
144 - precision: 0.9423 - recall: 0.9312 - val_accuracy: 0.9092 - val_auc: 0.9912 - val_f1_score: 0.9109 -
val_loss: 0.5007 - val_precision: 0.9143 - val_recall: 0.9049
Epoch 36/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 41s 118ms/step - accuracy: 0.9326 - auc: 0.9965 - f1_score: 0.9330 - loss: 0.4
141 - precision: 0.9370 - recall: 0.9279 - val_accuracy: 0.9088 - val_auc: 0.9902 - val_f1_score: 0.9099 -
val_loss: 0.5028 - val_precision: 0.9140 - val_recall: 0.9052
Epoch 37/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9367 - auc: 0.9961 - f1_score: 0.9375 - loss: 0.4
093 - precision: 0.9412 - recall: 0.9317 - val_accuracy: 0.9088 - val_auc: 0.9909 - val_f1_score: 0.9103 -
val_loss: 0.4918 - val_precision: 0.9137 - val_recall: 0.9049
Epoch 38/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9408 - auc: 0.9970 - f1_score: 0.9415 - loss: 0.3
869 - precision: 0.9449 - recall: 0.9363 - val_accuracy: 0.9010 - val_auc: 0.9907 - val_f1_score: 0.9023 -
val_loss: 0.4980 - val_precision: 0.9063 - val_recall: 0.8995
Epoch 39/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9374 - auc: 0.9967 - f1_score: 0.9381 - loss: 0.3
892 - precision: 0.9414 - recall: 0.9310 - val_accuracy: 0.9088 - val_auc: 0.9904 - val_f1_score: 0.9103 -
val_loss: 0.4852 - val_precision: 0.9126 - val_recall: 0.9074
Epoch 40/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9404 - auc: 0.9970 - f1_score: 0.9409 - loss: 0.3
781 - precision: 0.9457 - recall: 0.9368 - val_accuracy: 0.9067 - val_auc: 0.9904 - val_f1_score: 0.9078 -
val_loss: 0.4918 - val_precision: 0.9116 - val_recall: 0.9042
Epoch 41/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9398 - auc: 0.9973 - f1_score: 0.9404 - loss: 0.3
657 - precision: 0.9450 - recall: 0.9363 - val_accuracy: 0.9081 - val_auc: 0.9900 - val_f1_score: 0.9098 -
val_loss: 0.4825 - val_precision: 0.9113 - val_recall: 0.9045
Epoch 42/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9414 - auc: 0.9968 - f1_score: 0.9420 - loss: 0.3
665 - precision: 0.9462 - recall: 0.9375 - val_accuracy: 0.9116 - val_auc: 0.9907 - val_f1_score: 0.9130 -
val_loss: 0.4724 - val_precision: 0.9147 - val_recall: 0.9095
```

```
Epoch 43/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9385 - auc: 0.9970 - f1_score: 0.9389 - loss: 0.3
662 - precision: 0.9434 - recall: 0.9338 - val_accuracy: 0.9042 - val_auc: 0.9910 - val_f1_score: 0.9056 -
val_loss: 0.4710 - val_precision: 0.9069 - val_recall: 0.9027
Epoch 44/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9460 - auc: 0.9974 - f1_score: 0.9466 - loss: 0.3
465 - precision: 0.9505 - recall: 0.9422 - val_accuracy: 0.9084 - val_auc: 0.9901 - val_f1_score: 0.9096 -
val_loss: 0.4696 - val_precision: 0.9116 - val_recall: 0.9070
Epoch 45/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9483 - auc: 0.9971 - f1_score: 0.9488 - loss: 0.3
470 - precision: 0.9520 - recall: 0.9431 - val_accuracy: 0.8999 - val_auc: 0.9897 - val_f1_score: 0.9017 -
val_loss: 0.4758 - val_precision: 0.9045 - val_recall: 0.8974
Epoch 46/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9457 - auc: 0.9971 - f1_score: 0.9462 - loss: 0.3
454 - precision: 0.9499 - recall: 0.9421 - val_accuracy: 0.9059 - val_auc: 0.9899 - val_f1_score: 0.9073 -
val_loss: 0.4666 - val_precision: 0.9106 - val_recall: 0.9031
Epoch 47/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9409 - auc: 0.9973 - f1_score: 0.9416 - loss: 0.3
436 - precision: 0.9451 - recall: 0.9365 - val_accuracy: 0.9084 - val_auc: 0.9899 - val_f1_score: 0.9092 -
val_loss: 0.4608 - val_precision: 0.9131 - val_recall: 0.9056
Epoch 48/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9428 - auc: 0.9972 - f1_score: 0.9432 - loss: 0.3
404 - precision: 0.9460 - recall: 0.9388 - val_accuracy: 0.9088 - val_auc: 0.9905 - val_f1_score: 0.9102 -
val_loss: 0.4503 - val_precision: 0.9130 - val_recall: 0.9052
Epoch 49/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9432 - auc: 0.9975 - f1_score: 0.9436 - loss: 0.3
337 - precision: 0.9459 - recall: 0.9388 - val_accuracy: 0.9070 - val_auc: 0.9903 - val_f1_score: 0.9085 -
val_loss: 0.4503 - val_precision: 0.9116 - val_recall: 0.9035
Epoch 50/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9506 - auc: 0.9979 - f1_score: 0.9510 - loss: 0.3
194 - precision: 0.9544 - recall: 0.9468 - val_accuracy: 0.9045 - val_auc: 0.9911 - val_f1_score: 0.9061 -
val_loss: 0.4505 - val_precision: 0.9086 - val_recall: 0.8995
Epoch 51/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9485 - auc: 0.9975 - f1_score: 0.9493 - loss: 0.3
215 - precision: 0.9522 - recall: 0.9449 - val_accuracy: 0.8970 - val_auc: 0.9904 - val_f1_score: 0.8987 -
val_loss: 0.4654 - val_precision: 0.9019 - val_recall: 0.8942
Epoch 52/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9508 - auc: 0.9976 - f1_score: 0.9513 - loss: 0.3
151 - precision: 0.9535 - recall: 0.9464 - val_accuracy: 0.9059 - val_auc: 0.9897 - val_f1_score: 0.9076 -
val_loss: 0.4566 - val_precision: 0.9090 - val_recall: 0.9038
Epoch 53/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9440 - auc: 0.9975 - f1_score: 0.9446 - loss: 0.3
```

```
180 - precision: 0.9495 - recall: 0.9399 - val_accuracy: 0.9024 - val_auc: 0.9901 - val_f1_score: 0.9044 -
val_loss: 0.4565 - val_precision: 0.9061 - val_recall: 0.8974
Epoch 54/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 121ms/step - accuracy: 0.9431 - auc: 0.9975 - f1_score: 0.9434 - loss: 0.3
161 - precision: 0.9477 - recall: 0.9393 - val_accuracy: 0.9056 - val_auc: 0.9903 - val_f1_score: 0.9072 -
val_loss: 0.4471 - val_precision: 0.9083 - val_recall: 0.9035
Epoch 55/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 122ms/step - accuracy: 0.9497 - auc: 0.9975 - f1_score: 0.9503 - loss: 0.3
059 - precision: 0.9531 - recall: 0.9460 - val_accuracy: 0.9027 - val_auc: 0.9893 - val_f1_score: 0.9047 -
val_loss: 0.4593 - val_precision: 0.9077 - val_recall: 0.9006
Epoch 56/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 121ms/step - accuracy: 0.9518 - auc: 0.9978 - f1_score: 0.9522 - loss: 0.3
000 - precision: 0.9556 - recall: 0.9500 - val_accuracy: 0.9045 - val_auc: 0.9897 - val_f1_score: 0.9061 -
val_loss: 0.4462 - val_precision: 0.9069 - val_recall: 0.9024
Epoch 57/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 121ms/step - accuracy: 0.9475 - auc: 0.9974 - f1_score: 0.9479 - loss: 0.3
101 - precision: 0.9518 - recall: 0.9426 - val_accuracy: 0.9059 - val_auc: 0.9898 - val_f1_score: 0.9076 -
val_loss: 0.4392 - val_precision: 0.9099 - val_recall: 0.9031
Epoch 58/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9500 - auc: 0.9979 - f1_score: 0.9505 - loss: 0.2
945 - precision: 0.9554 - recall: 0.9455 - val_accuracy: 0.9092 - val_auc: 0.9897 - val_f1_score: 0.9108 -
val_loss: 0.4377 - val_precision: 0.9133 - val_recall: 0.9077
Epoch 59/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 122ms/step - accuracy: 0.9481 - auc: 0.9978 - f1_score: 0.9486 - loss: 0.2
989 - precision: 0.9520 - recall: 0.9448 - val_accuracy: 0.9109 - val_auc: 0.9879 - val_f1_score: 0.9124 -
val_loss: 0.4438 - val_precision: 0.9140 - val_recall: 0.9084
Epoch 60/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9517 - auc: 0.9977 - f1_score: 0.9521 - loss: 0.2
904 - precision: 0.9541 - recall: 0.9477 - val_accuracy: 0.9052 - val_auc: 0.9886 - val_f1_score: 0.9068 -
val_loss: 0.4438 - val_precision: 0.9086 - val_recall: 0.9035
Epoch 61/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9493 - auc: 0.9978 - f1_score: 0.9499 - loss: 0.2
896 - precision: 0.9536 - recall: 0.9457 - val_accuracy: 0.9102 - val_auc: 0.9892 - val_f1_score: 0.9116 -
val_loss: 0.4324 - val_precision: 0.9117 - val_recall: 0.9081
Epoch 62/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9550 - auc: 0.9980 - f1_score: 0.9554 - loss: 0.2
807 - precision: 0.9574 - recall: 0.9512 - val_accuracy: 0.9095 - val_auc: 0.9893 - val_f1_score: 0.9112 -
val_loss: 0.4338 - val_precision: 0.9107 - val_recall: 0.9077
Epoch 63/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9512 - auc: 0.9980 - f1_score: 0.9517 - loss: 0.2
808 - precision: 0.9557 - recall: 0.9475 - val_accuracy: 0.9106 - val_auc: 0.9894 - val_f1_score: 0.9120 -
val_loss: 0.4282 - val_precision: 0.9131 - val_recall: 0.9056
```

```
Epoch 64/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9489 - auc: 0.9980 - f1_score: 0.9493 - loss: 0.2
810 - precision: 0.9546 - recall: 0.9466 - val_accuracy: 0.9059 - val_auc: 0.9885 - val_f1_score: 0.9074 -
val_loss: 0.4316 - val_precision: 0.9087 - val_recall: 0.9045
Epoch 65/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9520 - auc: 0.9979 - f1_score: 0.9525 - loss: 0.2
818 - precision: 0.9547 - recall: 0.9482 - val_accuracy: 0.8974 - val_auc: 0.9893 - val_f1_score: 0.8991 -
val_loss: 0.4330 - val_precision: 0.9024 - val_recall: 0.8960
Epoch 66/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9561 - auc: 0.9982 - f1_score: 0.9568 - loss: 0.2
702 - precision: 0.9586 - recall: 0.9519 - val_accuracy: 0.9002 - val_auc: 0.9886 - val_f1_score: 0.9022 -
val_loss: 0.4405 - val_precision: 0.9028 - val_recall: 0.8995
Epoch 67/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9509 - auc: 0.9981 - f1_score: 0.9513 - loss: 0.2
704 - precision: 0.9541 - recall: 0.9477 - val_accuracy: 0.9109 - val_auc: 0.9895 - val_f1_score: 0.9123 -
val_loss: 0.4224 - val_precision: 0.9139 - val_recall: 0.9074
Epoch 68/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9547 - auc: 0.9981 - f1_score: 0.9551 - loss: 0.2
724 - precision: 0.9568 - recall: 0.9518 - val_accuracy: 0.9024 - val_auc: 0.9887 - val_f1_score: 0.9038 -
val_loss: 0.4383 - val_precision: 0.9070 - val_recall: 0.8995
Epoch 69/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9575 - auc: 0.9983 - f1_score: 0.9580 - loss: 0.2
637 - precision: 0.9606 - recall: 0.9551 - val_accuracy: 0.9063 - val_auc: 0.9889 - val_f1_score: 0.9079 -
val_loss: 0.4250 - val_precision: 0.9116 - val_recall: 0.9038
Epoch 70/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9534 - auc: 0.9980 - f1_score: 0.9540 - loss: 0.2
680 - precision: 0.9564 - recall: 0.9502 - val_accuracy: 0.9052 - val_auc: 0.9893 - val_f1_score: 0.9068 -
val_loss: 0.4258 - val_precision: 0.9079 - val_recall: 0.9027
Epoch 71/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9556 - auc: 0.9979 - f1_score: 0.9562 - loss: 0.2
629 - precision: 0.9585 - recall: 0.9525 - val_accuracy: 0.9042 - val_auc: 0.9886 - val_f1_score: 0.9059 -
val_loss: 0.4238 - val_precision: 0.9065 - val_recall: 0.9017
Epoch 72/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 121ms/step - accuracy: 0.9579 - auc: 0.9981 - f1_score: 0.9584 - loss: 0.2
610 - precision: 0.9603 - recall: 0.9538 - val_accuracy: 0.9109 - val_auc: 0.9895 - val_f1_score: 0.9121 -
val_loss: 0.4182 - val_precision: 0.9161 - val_recall: 0.9106
Epoch 73/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9571 - auc: 0.9985 - f1_score: 0.9576 - loss: 0.2
545 - precision: 0.9599 - recall: 0.9541 - val_accuracy: 0.9081 - val_auc: 0.9881 - val_f1_score: 0.9093 -
val_loss: 0.4202 - val_precision: 0.9102 - val_recall: 0.9059
Epoch 74/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 119ms/step - accuracy: 0.9568 - auc: 0.9984 - f1_score: 0.9572 - loss: 0.2
```

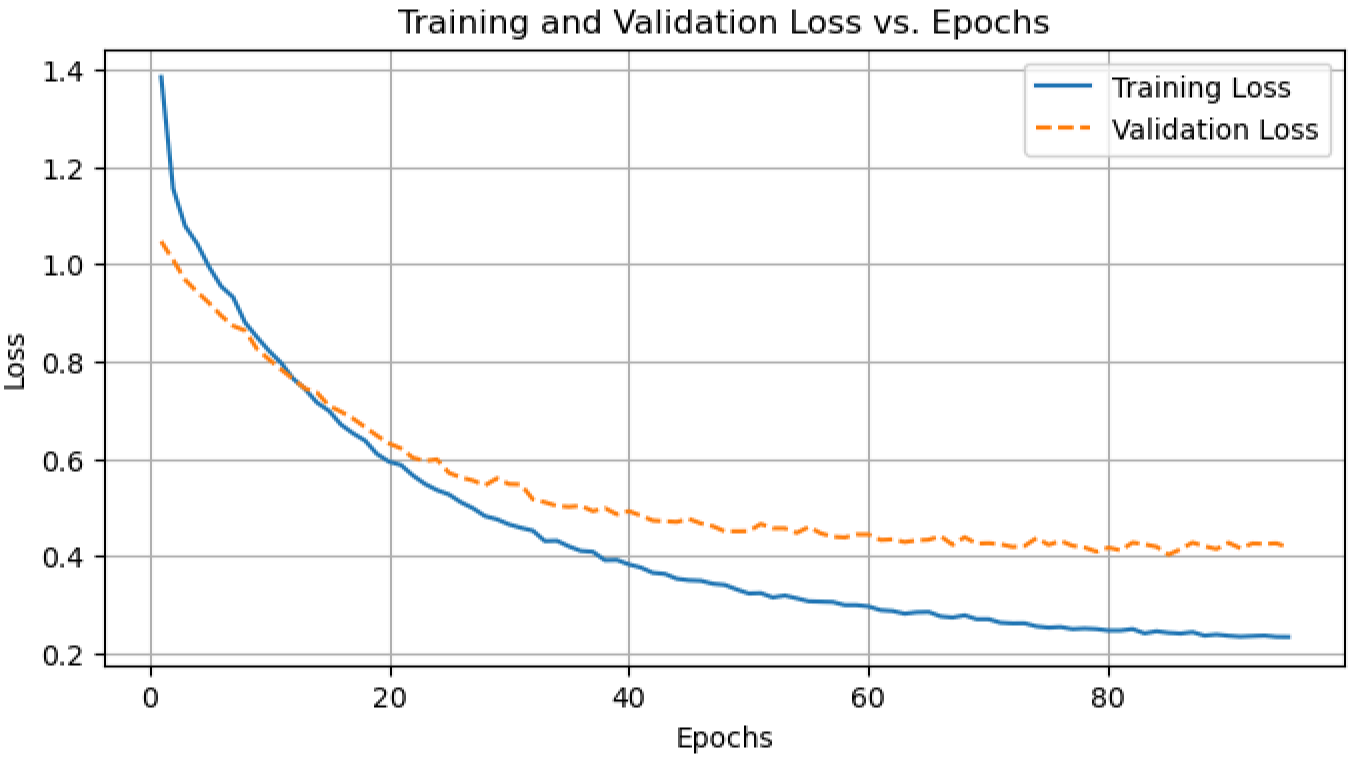
```
524 - precision: 0.9593 - recall: 0.9540 - val_accuracy: 0.9024 - val_auc: 0.9880 - val_f1_score: 0.9043 -
val_loss: 0.4365 - val_precision: 0.9044 - val_recall: 0.8995
Epoch 75/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9599 - auc: 0.9985 - f1_score: 0.9603 - loss: 0.2
469 - precision: 0.9625 - recall: 0.9578 - val_accuracy: 0.9006 - val_auc: 0.9893 - val_f1_score: 0.9021 -
val_loss: 0.4225 - val_precision: 0.9022 - val_recall: 0.8970
Epoch 76/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9567 - auc: 0.9985 - f1_score: 0.9572 - loss: 0.2
501 - precision: 0.9600 - recall: 0.9540 - val_accuracy: 0.9017 - val_auc: 0.9882 - val_f1_score: 0.9031 -
val_loss: 0.4306 - val_precision: 0.9043 - val_recall: 0.8988
Epoch 77/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9559 - auc: 0.9984 - f1_score: 0.9564 - loss: 0.2
527 - precision: 0.9583 - recall: 0.9537 - val_accuracy: 0.9045 - val_auc: 0.9895 - val_f1_score: 0.9064 -
val_loss: 0.4206 - val_precision: 0.9069 - val_recall: 0.9024
Epoch 78/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9595 - auc: 0.9984 - f1_score: 0.9601 - loss: 0.2
473 - precision: 0.9613 - recall: 0.9572 - val_accuracy: 0.9013 - val_auc: 0.9891 - val_f1_score: 0.9028 -
val_loss: 0.4170 - val_precision: 0.9026 - val_recall: 0.8978
Epoch 79/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9561 - auc: 0.9984 - f1_score: 0.9564 - loss: 0.2
456 - precision: 0.9582 - recall: 0.9531 - val_accuracy: 0.9088 - val_auc: 0.9893 - val_f1_score: 0.9104 -
val_loss: 0.4085 - val_precision: 0.9120 - val_recall: 0.9077
Epoch 80/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9599 - auc: 0.9985 - f1_score: 0.9603 - loss: 0.2
432 - precision: 0.9623 - recall: 0.9559 - val_accuracy: 0.9074 - val_auc: 0.9885 - val_f1_score: 0.9091 -
val_loss: 0.4169 - val_precision: 0.9106 - val_recall: 0.9067
Epoch 81/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9568 - auc: 0.9983 - f1_score: 0.9574 - loss: 0.2
480 - precision: 0.9597 - recall: 0.9536 - val_accuracy: 0.9127 - val_auc: 0.9893 - val_f1_score: 0.9141 -
val_loss: 0.4111 - val_precision: 0.9163 - val_recall: 0.9088
Epoch 82/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9553 - auc: 0.9983 - f1_score: 0.9558 - loss: 0.2
449 - precision: 0.9582 - recall: 0.9521 - val_accuracy: 0.9031 - val_auc: 0.9879 - val_f1_score: 0.9046 -
val_loss: 0.4270 - val_precision: 0.9078 - val_recall: 0.9017
Epoch 83/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9600 - auc: 0.9984 - f1_score: 0.9605 - loss: 0.2
414 - precision: 0.9619 - recall: 0.9586 - val_accuracy: 0.9063 - val_auc: 0.9871 - val_f1_score: 0.9083 -
val_loss: 0.4236 - val_precision: 0.9100 - val_recall: 0.9045
Epoch 84/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9585 - auc: 0.9984 - f1_score: 0.9589 - loss: 0.2
448 - precision: 0.9615 - recall: 0.9558 - val_accuracy: 0.9074 - val_auc: 0.9882 - val_f1_score: 0.9089 -
val_loss: 0.4186 - val_precision: 0.9094 - val_recall: 0.9045
```

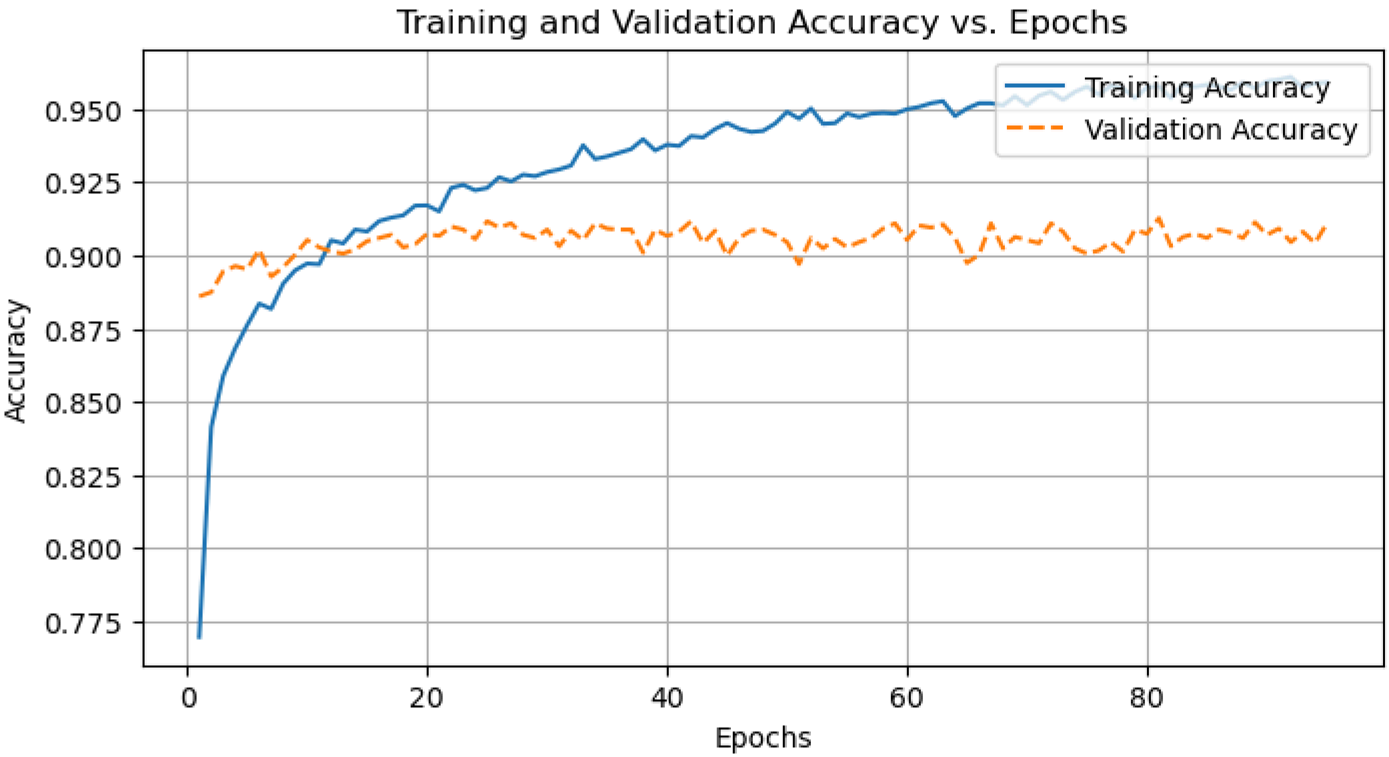
```
Epoch 85/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9610 - auc: 0.9985 - f1_score: 0.9614 - loss: 0.2
397 - precision: 0.9629 - recall: 0.9581 - val_accuracy: 0.9059 - val_auc: 0.9900 - val_f1_score: 0.9073 -
val_loss: 0.4022 - val_precision: 0.9102 - val_recall: 0.9031
Epoch 86/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 122ms/step - accuracy: 0.9590 - auc: 0.9986 - f1_score: 0.9595 - loss: 0.2
362 - precision: 0.9621 - recall: 0.9563 - val_accuracy: 0.9088 - val_auc: 0.9896 - val_f1_score: 0.9103 -
val_loss: 0.4134 - val_precision: 0.9125 - val_recall: 0.9070
Epoch 87/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 122ms/step - accuracy: 0.9578 - auc: 0.9982 - f1_score: 0.9583 - loss: 0.2
406 - precision: 0.9608 - recall: 0.9542 - val_accuracy: 0.9077 - val_auc: 0.9876 - val_f1_score: 0.9088 -
val_loss: 0.4271 - val_precision: 0.9088 - val_recall: 0.9056
Epoch 88/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9602 - auc: 0.9986 - f1_score: 0.9606 - loss: 0.2
322 - precision: 0.9633 - recall: 0.9577 - val_accuracy: 0.9059 - val_auc: 0.9887 - val_f1_score: 0.9074 -
val_loss: 0.4197 - val_precision: 0.9086 - val_recall: 0.9027
Epoch 89/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 92s 264ms/step - accuracy: 0.9601 - auc: 0.9985 - f1_score: 0.9605 - loss: 0.2
329 - precision: 0.9637 - recall: 0.9573 - val_accuracy: 0.9113 - val_auc: 0.9891 - val_f1_score: 0.9129 -
val_loss: 0.4139 - val_precision: 0.9134 - val_recall: 0.9092
Epoch 90/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 49s 140ms/step - accuracy: 0.9574 - auc: 0.9985 - f1_score: 0.9579 - loss: 0.2
366 - precision: 0.9604 - recall: 0.9552 - val_accuracy: 0.9070 - val_auc: 0.9881 - val_f1_score: 0.9080 -
val_loss: 0.4269 - val_precision: 0.9103 - val_recall: 0.9042
Epoch 91/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 45s 129ms/step - accuracy: 0.9609 - auc: 0.9983 - f1_score: 0.9613 - loss: 0.2
347 - precision: 0.9627 - recall: 0.9583 - val_accuracy: 0.9092 - val_auc: 0.9885 - val_f1_score: 0.9108 -
val_loss: 0.4155 - val_precision: 0.9113 - val_recall: 0.9074
Epoch 92/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9630 - auc: 0.9984 - f1_score: 0.9634 - loss: 0.2
302 - precision: 0.9644 - recall: 0.9612 - val_accuracy: 0.9045 - val_auc: 0.9881 - val_f1_score: 0.9065 -
val_loss: 0.4253 - val_precision: 0.9071 - val_recall: 0.9010
Epoch 93/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 43s 121ms/step - accuracy: 0.9565 - auc: 0.9984 - f1_score: 0.9568 - loss: 0.2
331 - precision: 0.9590 - recall: 0.9547 - val_accuracy: 0.9081 - val_auc: 0.9876 - val_f1_score: 0.9099 -
val_loss: 0.4245 - val_precision: 0.9120 - val_recall: 0.9049
Epoch 94/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 120ms/step - accuracy: 0.9581 - auc: 0.9982 - f1_score: 0.9584 - loss: 0.2
361 - precision: 0.9607 - recall: 0.9559 - val_accuracy: 0.9042 - val_auc: 0.9873 - val_f1_score: 0.9061 -
val_loss: 0.4255 - val_precision: 0.9076 - val_recall: 0.9024
Epoch 95/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 42s 121ms/step - accuracy: 0.9595 - auc: 0.9983 - f1_score: 0.9599 - loss: 0.2
```

```
341 - precision: 0.9622 - recall: 0.9551 - val_accuracy: 0.9109 - val_auc: 0.9882 - val_f1_score: 0.9126 -
val_loss: 0.4173 - val_precision: 0.9147 - val_recall: 0.9092
Epoch 95: early stopping
Restoring model weights from the end of the best epoch: 85.
```

In [111…

```
plotting_the_curves(history_EnB0)
```





In [113…

```
test_loss_EnB0, test_accuracy_EnB0, precision_EnB0, recall_EnB0, f1_EnB0, auc_EnB0 = get_test_loss_and_accu
```

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 9s 86ms/step - accuracy: 0.9094 - auc: 0.9885 - f1_score: 0.5498 - loss: 0.4094
- precision: 0.9112 - recall: 0.9087
Test Loss EfficientNetB0: 0.4397
Test Accuracy EfficientNetB0: 0.9023
94/94 ━━━━━━━━━━━━━━━━━━━━ 9s 91ms/step

----------For EfficientNetB0 Model----------

      Metric     Score
0  Precision  0.901998
1     Recall  0.902333
2   F1 Score  0.901972
3        AUC  0.989722

Classification Report:
              precision    recall  f1-score   support

   buildings       0.90      0.91      0.90       437
      forest       0.99      0.99      0.99       474
     glacier       0.84      0.83      0.83       553
    mountain       0.87      0.83      0.85       525
         sea       0.91      0.95      0.93       510
      street       0.92      0.91      0.91       501

    accuracy                           0.90      3000
   macro avg       0.90      0.90      0.90      3000
weighted avg       0.90      0.90      0.90      3000
```

VGG16

In [38]:

```python
model_name_50 = "VGG16"
training_images, training_labels, class_names = getting_paths_and_labels(train_dir, image_size, model_name_
test_images, test_labels, class_names = getting_paths_and_labels(test_dir, image_size, model_name_50)
```

In [39]:

```python
encoder = LabelEncoder()
training_labels_encoded = encoder.fit_transform(training_labels)
test_labels_encoded = encoder.fit_transform(test_labels)
```

In [40]:

```python
xTrain, yTrain, xTest, yTest = train_test_split(
    training_images, training_labels_encoded, test_size=0.2, random_state=42, stratify=training_labels_enc
```

In [41]:

```
training_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(xTest, num_classes=6)
validation_labels_with_one_hot_encoding = tf.keras.utils.to_categorical(yTest, num_classes=6)
test_labels_one_hot_encoding = tf.keras.utils.to_categorical(test_labels_encoded, num_classes=6)
```

In [42]:

```
train_ds = tf.data.Dataset.from_tensor_slices((xTrain, training_labels_with_one_hot_encoding)).batch(batch_

val_ds = tf.data.Dataset.from_tensor_slices((yTrain, validation_labels_with_one_hot_encoding)).batch(batch_
```

In [43]:

```
model_VGG16 = model_with_layers("VGG16")
```

```
Model: "sequential_3"

┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                        ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ data_augmentation (Sequential)      │ (None, 150, 150, 3)    │             0 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ vgg16 (Functional)                  │ (None, 512)            │    14,714,688 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten_3 (Flatten)                 │ (None, 512)            │             0 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_6 (Dense)                     │ (None, 512)            │       262,656 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_3               │ (None, 512)            │         2,048 │
│ (BatchNormalization)                │                        │               │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_3 (Dropout)                 │ (None, 512)            │             0 │
├─────────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_7 (Dense)                     │ (None, 6)              │         3,078 │
└─────────────────────────────────────┴────────────────────────┴───────────────┘

 Total params: 14,982,470 (57.15 MB)

 Trainable params: 266,758 (1.02 MB)

 Non-trainable params: 14,715,712 (56.14 MB)
```

In [44]:

```
history_VGG16 = get_history_and_fit_model(model_VGG16)
```

```
Epoch 1/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 287s 810ms/step - accuracy: 0.4744 - auc: 0.7918 - f1_score: 0.4704 - loss: 2.
0172 - precision: 0.5409 - recall: 0.3766 - val_accuracy: 0.7239 - val_auc: 0.9251 - val_f1_score: 0.7273 -
val_loss: 1.6303 - val_precision: 0.7364 - val_recall: 0.7175
Epoch 2/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.7227 - auc: 0.9396 - f1_score: 0.7221 - loss: 1.
2894 - precision: 0.7588 - recall: 0.6709 - val_accuracy: 0.7955 - val_auc: 0.9501 - val_f1_score: 0.7961 -
val_loss: 1.3444 - val_precision: 0.8007 - val_recall: 0.7898
Epoch 3/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.7482 - auc: 0.9513 - f1_score: 0.7480 - loss: 1.
1857 - precision: 0.7839 - recall: 0.7071 - val_accuracy: 0.8048 - val_auc: 0.9566 - val_f1_score: 0.8061 -
val_loss: 1.2678 - val_precision: 0.8086 - val_recall: 0.8009
Epoch 4/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.7595 - auc: 0.9563 - f1_score: 0.7593 - loss: 1.
1385 - precision: 0.7953 - recall: 0.7216 - val_accuracy: 0.8201 - val_auc: 0.9623 - val_f1_score: 0.8211 -
val_loss: 1.1741 - val_precision: 0.8246 - val_recall: 0.8172
Epoch 5/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.7666 - auc: 0.9598 - f1_score: 0.7660 - loss: 1.
0959 - precision: 0.8022 - recall: 0.7323 - val_accuracy: 0.8283 - val_auc: 0.9629 - val_f1_score: 0.8295 -
val_loss: 1.1605 - val_precision: 0.8319 - val_recall: 0.8269
Epoch 6/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.7842 - auc: 0.9621 - f1_score: 0.7843 - loss: 1.
0599 - precision: 0.8144 - recall: 0.7488 - val_accuracy: 0.8425 - val_auc: 0.9666 - val_f1_score: 0.8438 -
val_loss: 1.0931 - val_precision: 0.8457 - val_recall: 0.8418
Epoch 7/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 802ms/step - accuracy: 0.7822 - auc: 0.9640 - f1_score: 0.7819 - loss: 1.
0319 - precision: 0.8119 - recall: 0.7497 - val_accuracy: 0.8301 - val_auc: 0.9629 - val_f1_score: 0.8302 -
val_loss: 1.1545 - val_precision: 0.8329 - val_recall: 0.8276
Epoch 8/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.7905 - auc: 0.9660 - f1_score: 0.7906 - loss: 1.
0011 - precision: 0.8249 - recall: 0.7547 - val_accuracy: 0.8461 - val_auc: 0.9675 - val_f1_score: 0.8478 -
val_loss: 1.0467 - val_precision: 0.8491 - val_recall: 0.8440
Epoch 9/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 800ms/step - accuracy: 0.7978 - auc: 0.9670 - f1_score: 0.7978 - loss: 0.
9814 - precision: 0.8272 - recall: 0.7661 - val_accuracy: 0.8468 - val_auc: 0.9698 - val_f1_score: 0.8488 -
val_loss: 1.0201 - val_precision: 0.8513 - val_recall: 0.8443
Epoch 10/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 804ms/step - accuracy: 0.8007 - auc: 0.9678 - f1_score: 0.8013 - loss: 0.
9616 - precision: 0.8296 - recall: 0.7715 - val_accuracy: 0.8418 - val_auc: 0.9675 - val_f1_score: 0.8433 -
val_loss: 1.0579 - val_precision: 0.8453 - val_recall: 0.8390
Epoch 11/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 802ms/step - accuracy: 0.8044 - auc: 0.9694 - f1_score: 0.8049 - loss: 0.
```

```
9334 - precision: 0.8335 - recall: 0.7726 - val_accuracy: 0.8461 - val_auc: 0.9682 - val_f1_score: 0.8482 -
val_loss: 1.0249 - val_precision: 0.8481 - val_recall: 0.8436
Epoch 12/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 802ms/step - accuracy: 0.8069 - auc: 0.9701 - f1_score: 0.8073 - loss: 0.
9161 - precision: 0.8371 - recall: 0.7758 - val_accuracy: 0.8554 - val_auc: 0.9693 - val_f1_score: 0.8571 -
val_loss: 0.9996 - val_precision: 0.8565 - val_recall: 0.8525
Epoch 13/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 802ms/step - accuracy: 0.8052 - auc: 0.9703 - f1_score: 0.8055 - loss: 0.
9072 - precision: 0.8340 - recall: 0.7744 - val_accuracy: 0.8500 - val_auc: 0.9714 - val_f1_score: 0.8519 -
val_loss: 0.9533 - val_precision: 0.8520 - val_recall: 0.8493
Epoch 14/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 802ms/step - accuracy: 0.8150 - auc: 0.9721 - f1_score: 0.8154 - loss: 0.
8775 - precision: 0.8406 - recall: 0.7819 - val_accuracy: 0.8550 - val_auc: 0.9704 - val_f1_score: 0.8567 -
val_loss: 0.9721 - val_precision: 0.8571 - val_recall: 0.8525
Epoch 15/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 803ms/step - accuracy: 0.8207 - auc: 0.9752 - f1_score: 0.8212 - loss: 0.
8402 - precision: 0.8478 - recall: 0.7895 - val_accuracy: 0.8596 - val_auc: 0.9717 - val_f1_score: 0.8616 -
val_loss: 0.9343 - val_precision: 0.8598 - val_recall: 0.8564
Epoch 16/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 804ms/step - accuracy: 0.8180 - auc: 0.9724 - f1_score: 0.8183 - loss: 0.
8556 - precision: 0.8424 - recall: 0.7838 - val_accuracy: 0.8518 - val_auc: 0.9691 - val_f1_score: 0.8534 -
val_loss: 0.9582 - val_precision: 0.8539 - val_recall: 0.8497
Epoch 17/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 808ms/step - accuracy: 0.8211 - auc: 0.9744 - f1_score: 0.8214 - loss: 0.
8288 - precision: 0.8472 - recall: 0.7918 - val_accuracy: 0.8525 - val_auc: 0.9702 - val_f1_score: 0.8541 -
val_loss: 0.9352 - val_precision: 0.8540 - val_recall: 0.8504
Epoch 18/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8260 - auc: 0.9751 - f1_score: 0.8264 - loss: 0.
8087 - precision: 0.8519 - recall: 0.7944 - val_accuracy: 0.8586 - val_auc: 0.9700 - val_f1_score: 0.8602 -
val_loss: 0.9319 - val_precision: 0.8592 - val_recall: 0.8568
Epoch 19/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 804ms/step - accuracy: 0.8205 - auc: 0.9760 - f1_score: 0.8211 - loss: 0.
7934 - precision: 0.8456 - recall: 0.7942 - val_accuracy: 0.8514 - val_auc: 0.9675 - val_f1_score: 0.8528 -
val_loss: 0.9739 - val_precision: 0.8533 - val_recall: 0.8493
Epoch 20/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 809ms/step - accuracy: 0.8205 - auc: 0.9754 - f1_score: 0.8205 - loss: 0.
7902 - precision: 0.8483 - recall: 0.7906 - val_accuracy: 0.8482 - val_auc: 0.9705 - val_f1_score: 0.8500 -
val_loss: 0.9203 - val_precision: 0.8494 - val_recall: 0.8461
Epoch 21/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8337 - auc: 0.9775 - f1_score: 0.8341 - loss: 0.
7634 - precision: 0.8579 - recall: 0.8024 - val_accuracy: 0.8561 - val_auc: 0.9713 - val_f1_score: 0.8577 -
val_loss: 0.8816 - val_precision: 0.8579 - val_recall: 0.8536
```

```
Epoch 22/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 805ms/step - accuracy: 0.8220 - auc: 0.9758 - f1_score: 0.8223 - loss: 0.
7711 - precision: 0.8497 - recall: 0.7944 - val_accuracy: 0.8571 - val_auc: 0.9712 - val_f1_score: 0.8588 -
val_loss: 0.8833 - val_precision: 0.8586 - val_recall: 0.8564
Epoch 23/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.8355 - auc: 0.9776 - f1_score: 0.8360 - loss: 0.
7439 - precision: 0.8607 - recall: 0.8081 - val_accuracy: 0.8621 - val_auc: 0.9712 - val_f1_score: 0.8639 -
val_loss: 0.8506 - val_precision: 0.8621 - val_recall: 0.8596
Epoch 24/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 804ms/step - accuracy: 0.8324 - auc: 0.9782 - f1_score: 0.8330 - loss: 0.
7318 - precision: 0.8544 - recall: 0.8019 - val_accuracy: 0.8596 - val_auc: 0.9723 - val_f1_score: 0.8612 -
val_loss: 0.8534 - val_precision: 0.8598 - val_recall: 0.8568
Epoch 25/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 805ms/step - accuracy: 0.8326 - auc: 0.9779 - f1_score: 0.8330 - loss: 0.
7278 - precision: 0.8588 - recall: 0.8083 - val_accuracy: 0.8625 - val_auc: 0.9724 - val_f1_score: 0.8642 -
val_loss: 0.8294 - val_precision: 0.8629 - val_recall: 0.8611
Epoch 26/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 804ms/step - accuracy: 0.8325 - auc: 0.9787 - f1_score: 0.8327 - loss: 0.
7124 - precision: 0.8579 - recall: 0.8048 - val_accuracy: 0.8653 - val_auc: 0.9729 - val_f1_score: 0.8670 -
val_loss: 0.8201 - val_precision: 0.8657 - val_recall: 0.8636
Epoch 27/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 802ms/step - accuracy: 0.8392 - auc: 0.9799 - f1_score: 0.8397 - loss: 0.
6952 - precision: 0.8644 - recall: 0.8127 - val_accuracy: 0.8657 - val_auc: 0.9712 - val_f1_score: 0.8673 -
val_loss: 0.8369 - val_precision: 0.8662 - val_recall: 0.8625
Epoch 28/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8390 - auc: 0.9803 - f1_score: 0.8396 - loss: 0.
6850 - precision: 0.8650 - recall: 0.8127 - val_accuracy: 0.8671 - val_auc: 0.9729 - val_f1_score: 0.8690 -
val_loss: 0.8043 - val_precision: 0.8680 - val_recall: 0.8643
Epoch 29/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8416 - auc: 0.9806 - f1_score: 0.8421 - loss: 0.
6780 - precision: 0.8670 - recall: 0.8123 - val_accuracy: 0.8671 - val_auc: 0.9710 - val_f1_score: 0.8687 -
val_loss: 0.8031 - val_precision: 0.8683 - val_recall: 0.8664
Epoch 30/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8437 - auc: 0.9804 - f1_score: 0.8444 - loss: 0.
6706 - precision: 0.8686 - recall: 0.8159 - val_accuracy: 0.8611 - val_auc: 0.9722 - val_f1_score: 0.8625 -
val_loss: 0.8094 - val_precision: 0.8617 - val_recall: 0.8589
Epoch 31/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8425 - auc: 0.9808 - f1_score: 0.8427 - loss: 0.
6634 - precision: 0.8700 - recall: 0.8160 - val_accuracy: 0.8657 - val_auc: 0.9719 - val_f1_score: 0.8669 -
val_loss: 0.8089 - val_precision: 0.8670 - val_recall: 0.8643
Epoch 32/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 808ms/step - accuracy: 0.8424 - auc: 0.9809 - f1_score: 0.8428 - loss: 0.
```

```
6567 - precision: 0.8648 - recall: 0.8166 - val_accuracy: 0.8632 - val_auc: 0.9740 - val_f1_score: 0.8652 -
val_loss: 0.7716 - val_precision: 0.8632 - val_recall: 0.8611
Epoch 33/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 809ms/step - accuracy: 0.8401 - auc: 0.9814 - f1_score: 0.8409 - loss: 0.
6475 - precision: 0.8665 - recall: 0.8129 - val_accuracy: 0.8703 - val_auc: 0.9736 - val_f1_score: 0.8719 -
val_loss: 0.7674 - val_precision: 0.8708 - val_recall: 0.8689
Epoch 34/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8431 - auc: 0.9817 - f1_score: 0.8434 - loss: 0.
6360 - precision: 0.8658 - recall: 0.8194 - val_accuracy: 0.8657 - val_auc: 0.9734 - val_f1_score: 0.8674 -
val_loss: 0.7688 - val_precision: 0.8670 - val_recall: 0.8639
Epoch 35/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8444 - auc: 0.9815 - f1_score: 0.8450 - loss: 0.
6373 - precision: 0.8664 - recall: 0.8166 - val_accuracy: 0.8675 - val_auc: 0.9725 - val_f1_score: 0.8692 -
val_loss: 0.7857 - val_precision: 0.8679 - val_recall: 0.8639
Epoch 36/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8412 - auc: 0.9817 - f1_score: 0.8414 - loss: 0.
6284 - precision: 0.8647 - recall: 0.8191 - val_accuracy: 0.8710 - val_auc: 0.9722 - val_f1_score: 0.8724 -
val_loss: 0.7575 - val_precision: 0.8724 - val_recall: 0.8696
Epoch 37/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 805ms/step - accuracy: 0.8556 - auc: 0.9837 - f1_score: 0.8560 - loss: 0.
6024 - precision: 0.8763 - recall: 0.8287 - val_accuracy: 0.8664 - val_auc: 0.9705 - val_f1_score: 0.8678 -
val_loss: 0.7959 - val_precision: 0.8685 - val_recall: 0.8657
Epoch 38/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 805ms/step - accuracy: 0.8571 - auc: 0.9831 - f1_score: 0.8581 - loss: 0.
6025 - precision: 0.8802 - recall: 0.8323 - val_accuracy: 0.8653 - val_auc: 0.9710 - val_f1_score: 0.8671 -
val_loss: 0.7896 - val_precision: 0.8676 - val_recall: 0.8636
Epoch 39/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 811ms/step - accuracy: 0.8538 - auc: 0.9837 - f1_score: 0.8541 - loss: 0.
5949 - precision: 0.8746 - recall: 0.8328 - val_accuracy: 0.8664 - val_auc: 0.9730 - val_f1_score: 0.8680 -
val_loss: 0.7644 - val_precision: 0.8676 - val_recall: 0.8660
Epoch 40/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 802ms/step - accuracy: 0.8550 - auc: 0.9837 - f1_score: 0.8555 - loss: 0.
5911 - precision: 0.8758 - recall: 0.8309 - val_accuracy: 0.8668 - val_auc: 0.9721 - val_f1_score: 0.8687 -
val_loss: 0.7763 - val_precision: 0.8678 - val_recall: 0.8653
Epoch 41/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8548 - auc: 0.9835 - f1_score: 0.8548 - loss: 0.
5883 - precision: 0.8743 - recall: 0.8352 - val_accuracy: 0.8643 - val_auc: 0.9707 - val_f1_score: 0.8660 -
val_loss: 0.7756 - val_precision: 0.8663 - val_recall: 0.8636
Epoch 42/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 805ms/step - accuracy: 0.8527 - auc: 0.9838 - f1_score: 0.8535 - loss: 0.
5826 - precision: 0.8728 - recall: 0.8300 - val_accuracy: 0.8678 - val_auc: 0.9716 - val_f1_score: 0.8694 -
val_loss: 0.7566 - val_precision: 0.8689 - val_recall: 0.8668
```

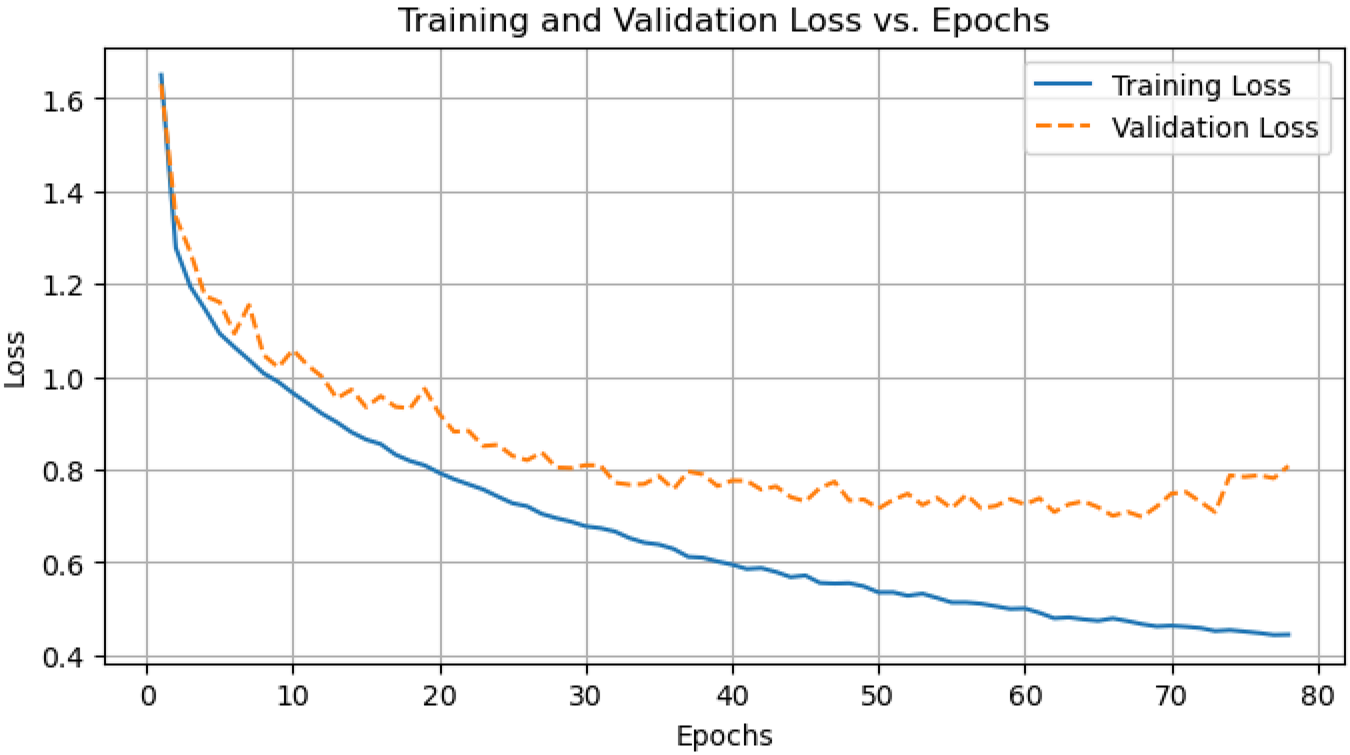
```
Epoch 43/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 805ms/step - accuracy: 0.8584 - auc: 0.9837 - f1_score: 0.8592 - loss: 0.
5778 - precision: 0.8775 - recall: 0.8341 - val_accuracy: 0.8671 - val_auc: 0.9710 - val_f1_score: 0.8689 -
val_loss: 0.7636 - val_precision: 0.8680 - val_recall: 0.8646
Epoch 44/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 805ms/step - accuracy: 0.8580 - auc: 0.9846 - f1_score: 0.8587 - loss: 0.
5681 - precision: 0.8789 - recall: 0.8366 - val_accuracy: 0.8671 - val_auc: 0.9739 - val_f1_score: 0.8688 -
val_loss: 0.7407 - val_precision: 0.8689 - val_recall: 0.8664
Epoch 45/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8591 - auc: 0.9851 - f1_score: 0.8595 - loss: 0.
5593 - precision: 0.8819 - recall: 0.8352 - val_accuracy: 0.8710 - val_auc: 0.9728 - val_f1_score: 0.8726 -
val_loss: 0.7313 - val_precision: 0.8723 - val_recall: 0.8685
Epoch 46/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 808ms/step - accuracy: 0.8587 - auc: 0.9850 - f1_score: 0.8594 - loss: 0.
5566 - precision: 0.8794 - recall: 0.8369 - val_accuracy: 0.8675 - val_auc: 0.9708 - val_f1_score: 0.8692 -
val_loss: 0.7604 - val_precision: 0.8684 - val_recall: 0.8653
Epoch 47/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8654 - auc: 0.9853 - f1_score: 0.8659 - loss: 0.
5493 - precision: 0.8846 - recall: 0.8465 - val_accuracy: 0.8675 - val_auc: 0.9710 - val_f1_score: 0.8694 -
val_loss: 0.7737 - val_precision: 0.8675 - val_recall: 0.8657
Epoch 48/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8631 - auc: 0.9853 - f1_score: 0.8635 - loss: 0.
5485 - precision: 0.8843 - recall: 0.8433 - val_accuracy: 0.8700 - val_auc: 0.9740 - val_f1_score: 0.8719 -
val_loss: 0.7334 - val_precision: 0.8710 - val_recall: 0.8682
Epoch 49/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8628 - auc: 0.9860 - f1_score: 0.8630 - loss: 0.
5355 - precision: 0.8837 - recall: 0.8409 - val_accuracy: 0.8710 - val_auc: 0.9737 - val_f1_score: 0.8729 -
val_loss: 0.7353 - val_precision: 0.8724 - val_recall: 0.8696
Epoch 50/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8689 - auc: 0.9862 - f1_score: 0.8691 - loss: 0.
5315 - precision: 0.8879 - recall: 0.8457 - val_accuracy: 0.8742 - val_auc: 0.9740 - val_f1_score: 0.8757 -
val_loss: 0.7159 - val_precision: 0.8754 - val_recall: 0.8710
Epoch 51/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8695 - auc: 0.9864 - f1_score: 0.8700 - loss: 0.
5259 - precision: 0.8898 - recall: 0.8488 - val_accuracy: 0.8643 - val_auc: 0.9713 - val_f1_score: 0.8662 -
val_loss: 0.7348 - val_precision: 0.8662 - val_recall: 0.8628
Epoch 52/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 808ms/step - accuracy: 0.8722 - auc: 0.9868 - f1_score: 0.8726 - loss: 0.
5186 - precision: 0.8919 - recall: 0.8530 - val_accuracy: 0.8703 - val_auc: 0.9709 - val_f1_score: 0.8719 -
val_loss: 0.7477 - val_precision: 0.8708 - val_recall: 0.8689
Epoch 53/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8676 - auc: 0.9863 - f1_score: 0.8684 - loss: 0.
```

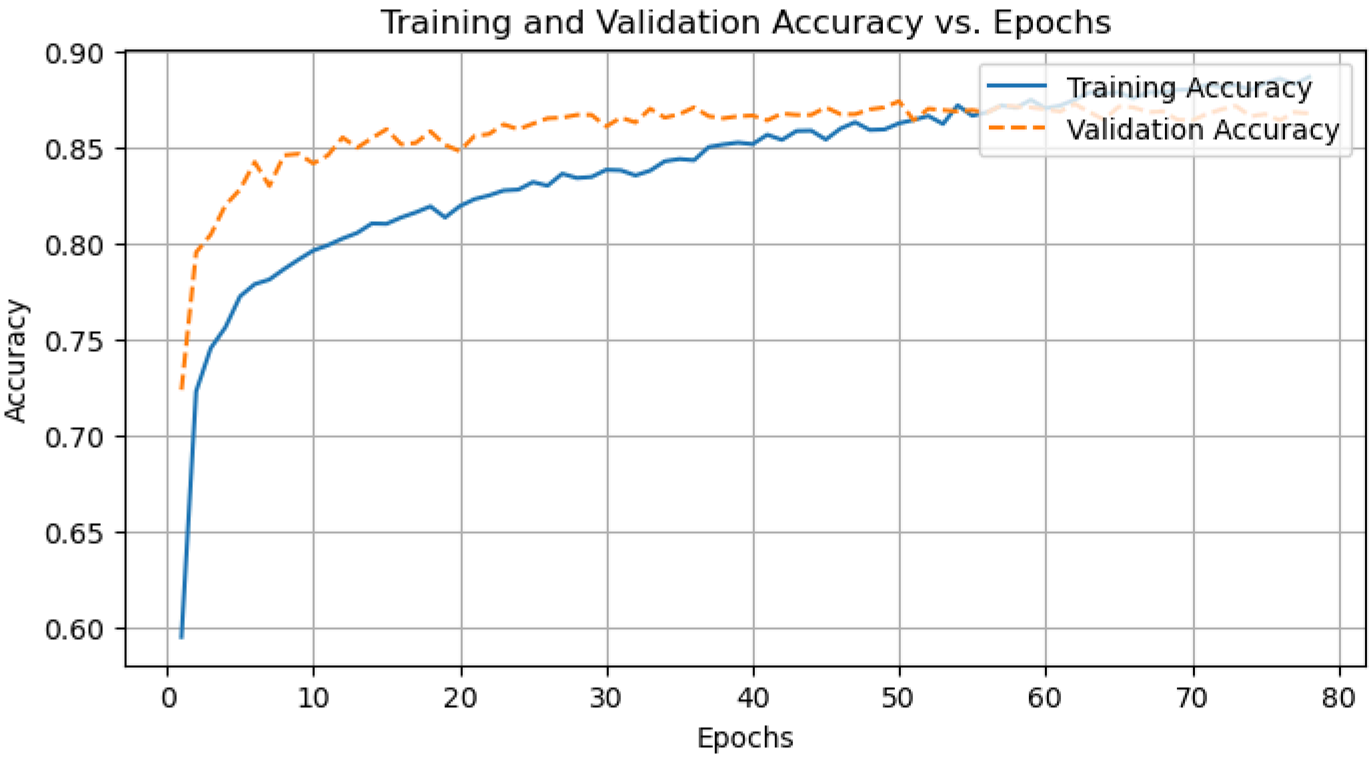
```
5243 - precision: 0.8868 - recall: 0.8486 - val_accuracy: 0.8696 - val_auc: 0.9724 - val_f1_score: 0.8714 -
val_loss: 0.7236 - val_precision: 0.8713 - val_recall: 0.8682
Epoch 54/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8754 - auc: 0.9865 - f1_score: 0.8760 - loss: 0.
5176 - precision: 0.8952 - recall: 0.8528 - val_accuracy: 0.8689 - val_auc: 0.9714 - val_f1_score: 0.8708 -
val_loss: 0.7389 - val_precision: 0.8702 - val_recall: 0.8668
Epoch 55/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8709 - auc: 0.9872 - f1_score: 0.8711 - loss: 0.
5053 - precision: 0.8912 - recall: 0.8520 - val_accuracy: 0.8696 - val_auc: 0.9741 - val_f1_score: 0.8714 -
val_loss: 0.7164 - val_precision: 0.8712 - val_recall: 0.8678
Epoch 56/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 808ms/step - accuracy: 0.8697 - auc: 0.9865 - f1_score: 0.8698 - loss: 0.
5126 - precision: 0.8886 - recall: 0.8513 - val_accuracy: 0.8682 - val_auc: 0.9701 - val_f1_score: 0.8705 -
val_loss: 0.7458 - val_precision: 0.8691 - val_recall: 0.8660
Epoch 57/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8782 - auc: 0.9879 - f1_score: 0.8790 - loss: 0.
4935 - precision: 0.8947 - recall: 0.8571 - val_accuracy: 0.8721 - val_auc: 0.9711 - val_f1_score: 0.8741 -
val_loss: 0.7166 - val_precision: 0.8740 - val_recall: 0.8721
Epoch 58/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 809ms/step - accuracy: 0.8735 - auc: 0.9877 - f1_score: 0.8738 - loss: 0.
4990 - precision: 0.8912 - recall: 0.8526 - val_accuracy: 0.8714 - val_auc: 0.9709 - val_f1_score: 0.8735 -
val_loss: 0.7215 - val_precision: 0.8718 - val_recall: 0.8696
Epoch 59/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 282s 805ms/step - accuracy: 0.8740 - auc: 0.9871 - f1_score: 0.8747 - loss: 0.
4995 - precision: 0.8918 - recall: 0.8568 - val_accuracy: 0.8710 - val_auc: 0.9710 - val_f1_score: 0.8731 -
val_loss: 0.7364 - val_precision: 0.8727 - val_recall: 0.8696
Epoch 60/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 808ms/step - accuracy: 0.8736 - auc: 0.9878 - f1_score: 0.8740 - loss: 0.
4925 - precision: 0.8905 - recall: 0.8535 - val_accuracy: 0.8700 - val_auc: 0.9696 - val_f1_score: 0.8725 -
val_loss: 0.7249 - val_precision: 0.8720 - val_recall: 0.8689
Epoch 61/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 808ms/step - accuracy: 0.8757 - auc: 0.9885 - f1_score: 0.8763 - loss: 0.
4808 - precision: 0.8951 - recall: 0.8593 - val_accuracy: 0.8689 - val_auc: 0.9697 - val_f1_score: 0.8708 -
val_loss: 0.7380 - val_precision: 0.8710 - val_recall: 0.8682
Epoch 62/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 808ms/step - accuracy: 0.8760 - auc: 0.9888 - f1_score: 0.8766 - loss: 0.
4720 - precision: 0.8928 - recall: 0.8584 - val_accuracy: 0.8728 - val_auc: 0.9727 - val_f1_score: 0.8744 -
val_loss: 0.7085 - val_precision: 0.8742 - val_recall: 0.8714
Epoch 63/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8793 - auc: 0.9884 - f1_score: 0.8798 - loss: 0.
4763 - precision: 0.8969 - recall: 0.8596 - val_accuracy: 0.8685 - val_auc: 0.9725 - val_f1_score: 0.8703 -
val_loss: 0.7250 - val_precision: 0.8700 - val_recall: 0.8675
```

```
Epoch 64/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 800ms/step - accuracy: 0.8832 - auc: 0.9891 - f1_score: 0.8839 - loss: 0.
4642 - precision: 0.9019 - recall: 0.8670 - val_accuracy: 0.8646 - val_auc: 0.9705 - val_f1_score: 0.8663 -
val_loss: 0.7315 - val_precision: 0.8660 - val_recall: 0.8632
Epoch 65/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 806ms/step - accuracy: 0.8825 - auc: 0.9893 - f1_score: 0.8833 - loss: 0.
4628 - precision: 0.9012 - recall: 0.8626 - val_accuracy: 0.8714 - val_auc: 0.9717 - val_f1_score: 0.8738 -
val_loss: 0.7187 - val_precision: 0.8728 - val_recall: 0.8700
Epoch 66/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 808ms/step - accuracy: 0.8773 - auc: 0.9885 - f1_score: 0.8780 - loss: 0.
4709 - precision: 0.8928 - recall: 0.8600 - val_accuracy: 0.8710 - val_auc: 0.9726 - val_f1_score: 0.8730 -
val_loss: 0.7003 - val_precision: 0.8726 - val_recall: 0.8689
Epoch 67/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 811ms/step - accuracy: 0.8835 - auc: 0.9886 - f1_score: 0.8836 - loss: 0.
4676 - precision: 0.8996 - recall: 0.8661 - val_accuracy: 0.8685 - val_auc: 0.9720 - val_f1_score: 0.8707 -
val_loss: 0.7088 - val_precision: 0.8691 - val_recall: 0.8678
Epoch 68/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 285s 813ms/step - accuracy: 0.8863 - auc: 0.9902 - f1_score: 0.8869 - loss: 0.
4481 - precision: 0.9019 - recall: 0.8693 - val_accuracy: 0.8689 - val_auc: 0.9728 - val_f1_score: 0.8709 -
val_loss: 0.6980 - val_precision: 0.8707 - val_recall: 0.8685
Epoch 69/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 285s 813ms/step - accuracy: 0.8859 - auc: 0.9894 - f1_score: 0.8863 - loss: 0.
4528 - precision: 0.9006 - recall: 0.8698 - val_accuracy: 0.8646 - val_auc: 0.9724 - val_f1_score: 0.8663 -
val_loss: 0.7202 - val_precision: 0.8651 - val_recall: 0.8636
Epoch 70/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 808ms/step - accuracy: 0.8834 - auc: 0.9893 - f1_score: 0.8840 - loss: 0.
4573 - precision: 0.8987 - recall: 0.8643 - val_accuracy: 0.8643 - val_auc: 0.9708 - val_f1_score: 0.8659 -
val_loss: 0.7480 - val_precision: 0.8654 - val_recall: 0.8636
Epoch 71/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 811ms/step - accuracy: 0.8850 - auc: 0.9897 - f1_score: 0.8854 - loss: 0.
4516 - precision: 0.9032 - recall: 0.8652 - val_accuracy: 0.8675 - val_auc: 0.9684 - val_f1_score: 0.8689 -
val_loss: 0.7515 - val_precision: 0.8676 - val_recall: 0.8660
Epoch 72/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 809ms/step - accuracy: 0.8826 - auc: 0.9893 - f1_score: 0.8830 - loss: 0.
4525 - precision: 0.8971 - recall: 0.8638 - val_accuracy: 0.8700 - val_auc: 0.9717 - val_f1_score: 0.8719 -
val_loss: 0.7301 - val_precision: 0.8710 - val_recall: 0.8685
Epoch 73/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 810ms/step - accuracy: 0.8878 - auc: 0.9894 - f1_score: 0.8883 - loss: 0.
4463 - precision: 0.9073 - recall: 0.8699 - val_accuracy: 0.8717 - val_auc: 0.9724 - val_f1_score: 0.8739 -
val_loss: 0.7082 - val_precision: 0.8727 - val_recall: 0.8696
Epoch 74/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 809ms/step - accuracy: 0.8831 - auc: 0.9899 - f1_score: 0.8837 - loss: 0.
```

```
4433 - precision: 0.8984 - recall: 0.8655 - val_accuracy: 0.8664 - val_auc: 0.9664 - val_f1_score: 0.8674 -
val_loss: 0.7872 - val_precision: 0.8672 - val_recall: 0.8657
Epoch 75/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 281s 801ms/step - accuracy: 0.8870 - auc: 0.9894 - f1_score: 0.8878 - loss: 0.
4486 - precision: 0.9022 - recall: 0.8675 - val_accuracy: 0.8675 - val_auc: 0.9662 - val_f1_score: 0.8697 -
val_loss: 0.7843 - val_precision: 0.8698 - val_recall: 0.8660
Epoch 76/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8853 - auc: 0.9899 - f1_score: 0.8861 - loss: 0.
4408 - precision: 0.8992 - recall: 0.8698 - val_accuracy: 0.8643 - val_auc: 0.9671 - val_f1_score: 0.8663 -
val_loss: 0.7876 - val_precision: 0.8665 - val_recall: 0.8621
Epoch 77/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 284s 810ms/step - accuracy: 0.8887 - auc: 0.9902 - f1_score: 0.8891 - loss: 0.
4377 - precision: 0.9015 - recall: 0.8700 - val_accuracy: 0.8685 - val_auc: 0.9669 - val_f1_score: 0.8708 -
val_loss: 0.7813 - val_precision: 0.8688 - val_recall: 0.8660
Epoch 78/100
351/351 ━━━━━━━━━━━━━━━━━━━━ 283s 807ms/step - accuracy: 0.8893 - auc: 0.9896 - f1_score: 0.8900 - loss: 0.
4423 - precision: 0.9047 - recall: 0.8684 - val_accuracy: 0.8678 - val_auc: 0.9668 - val_f1_score: 0.8692 -
val_loss: 0.8076 - val_precision: 0.8683 - val_recall: 0.8668
Epoch 78: early stopping
Restoring model weights from the end of the best epoch: 68.
```

In [45]:
```
plotting_the_curves(history_VGG16)
```


Training and Validation Loss vs. Epochs
Training Loss
Validation Loss
Loss
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
10
20
30
40
50
60
70
80
Epochs



In [46]:

```
test_loss_VGG16, test_accuracy_VGG16, precision_VGG16, recall_VGG16, f1_VGG16, auc_VGG16 = get_test_loss_ar
```

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 60s 634ms/step - accuracy: 0.8764 - auc: 0.9765 - f1_score: 0.5299 - loss: 0.661
2 - precision: 0.8805 - recall: 0.8760
Test Loss VGG16: 0.7023
Test Accuracy VGG16: 0.8643
94/94 ━━━━━━━━━━━━━━━━━━━━ 59s 631ms/step

----------For VGG16 Model----------

      Metric     Score
0  Precision  0.863516
1     Recall  0.864333
2   F1 Score  0.863692
3        AUC  0.982484

Classification Report:
              precision    recall  f1-score   support

   buildings       0.87      0.87      0.87       437
      forest       0.97      0.98      0.98       474
     glacier       0.78      0.79      0.79       553
    mountain       0.80      0.74      0.77       525
         sea       0.89      0.92      0.91       510
      street       0.89      0.89      0.89       501

    accuracy                           0.86      3000
   macro avg       0.87      0.87      0.87      3000
weighted avg       0.86      0.86      0.86      3000
```

Comparision Table of all models (For Test)

In [158…

```
data = {
    "Model": ["VGG16", "EfficientNetB0", "ResNet101", "ResNet50"],
    "Test Loss": [test_loss_VGG16, test_loss_EnB0, test_loss_res101, test_loss_res50],
    "Test Accuracy": [test_accuracy_VGG16, test_accuracy_EnB0, test_accuracy_res101, test_accuracy_res50],
    "Precision": [precision_VGG16, precision_EnB0, precision_res101, precision_res50],
    "Recall": [recall_VGG16, recall_EnB0, recall_res101, recall_res50],
    "F1 Score": [f1_VGG16, f1_EnB0, f1_res101, f1_res50],
    "AUC": [auc_VGG16, auc_EnB0, auc_res101, auc_res50]
}
```

```
results_df = pd.DataFrame(data).set_index("Model")
results_df
```

Out[158…

Model	Test Loss	Test Accuracy	Precision	Recall	F1 Score	AUC
VGG16	0.702267	0.864333	0.863516	0.864333	0.863692	0.982484
EfficientNetB0	0.439714	0.902333	0.901998	0.902333	0.901972	0.989722
ResNet101	0.567710	0.863667	0.863195	0.863667	0.861547	0.982191
ResNet50	0.540267	0.880333	0.879674	0.880333	0.879559	0.984165

From the above table we can say that **EfficientNetB0 model outperforms other models** by having a low test loss and high test accuracy.

References

Links:

1. https://keras.io/api/
2. https://www.tensorflow.org/guide/keras
3. https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.classification_report.html
4. https://medium.com/@silvershine1st/transfer-learning-for-cnn-7eed1d8a5305
5. https://stackoverflow.com/questions/60896416/tensorflow-keras-model-how-to-get-the-best-score-from-a-history-object
6. https://stackoverflow.com/questions/56897864/how-do-i-use-input-shape-and-input-tensor-in-transfer-learning-in-keras

GPT Prompts:

1. how to bold in print statement ? **Response - In Python, you can make text bold in the terminal using ANSI escape codes. The specific escape code for bold text is \033[1m. To reset the text to normal formatting, use \033[0m. print("\033[1mThis is bold text\033[0m") print("This is normal text")**

2. train_ds = tf.data.Dataset.from_tensor_slices((xTrain, training_labels_with_one_hot_encoding)).batch(batch_size) how can i add prefetch in this ? **Response - To add prefetching to your tf.data pipeline, you can use the prefetch method. Prefetching allows the data pipeline to prepare the next batch of data while the current batch is being processed by the model, reducing idle time and speeding up training. Here's how you can modify your train_ds pipeline to include prefetching: train_ds = tf.data.Dataset.from_tensor_slices((xTrain, training_labels_with_one_hot_encoding)) .batch(batch_size).prefetch(tf.data.AUTOTUNE) tf.data.AUTOTUNE: Dynamically determines the optimal number of elements to prefetch based on available system resources. prefetch: Ensures that data loading and preprocessing for the next batch happen in parallel with model training.**

```
In [ ]:
```