

Name: Parth Kulkarni

Batch: T2

PRN: 202201040007

Assignment 2

Transfer Learning

1. Transfer Learning Model Development: Fine-tune a pre-trained model on a specific dataset by modifying the top layers and optimizing hyperparameters.

Colab:- [Copy of Lab Assignment 2 Submossion .ipynb](#)

✓ Google Colab Lab Assignment -Pretrained Model

Course Name: Deep Learning (MDM)

Lab Title: Diabetic Retinopathy Detection Using VGG-16 Deep Learning Architecture

Student Name: Rohit Jagtap

Student ID: 202201040048

Date of Submission: 19/02/2025

Group Members:

Rohit Jagtap

Parth Kulkarni

Research Paper Study and Implementation

Instructions:

1. Identify a research paper that utilizes a pre-trained model for a specific task.
2. Study the methodology, dataset, and model used in the research paper.
3. Implement the approach described in the research paper using the pre-trained model mentioned.
4. Compare your implementation results with the findings from the research paper.

Objective

1. Study a research paper utilizing a pre-trained model.
2. Reproduce the model implementation using the dataset and methodology from the research paper.
3. Fine-tune the pre-trained model and optimize hyperparameters.
4. Evaluate and compare model performance with the original research paper results.

Task 1: Research Paper Selection and Dataset Preparation (2 hours)

Instructions:

1. Select a research paper that applies a pre-trained model (e.g., VGG, ResNet, EfficientNet, etc.).
2. Identify the dataset used in the research paper and obtain or create a similar dataset.
3. Perform necessary preprocessing steps:
 - Resize images to match the model input dimensions.
 - Apply data augmentation techniques if applicable.
4. Split the dataset into training, validation, and testing sets.

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

os.environ['KAGGLE_USERNAME'] = "rohitjaggy" # username from the json file
os.environ['KAGGLE_KEY'] = "e04cfe603764166128cbde139ffc704d" # key from the json file

!kaggle datasets download -d sovitrath/diabetic-retinopathy-224x224-2019-data

→ Dataset URL: https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-2019-data
License(s): CC0-1.0
diabetic-retinopathy-224x224-2019-data.zip: Skipping, found more recently modified local copy (use --force to force download)

!unzip diabetic-retinopathy-224x224-2019-data.zip

→ Archive: diabetic-retinopathy-224x224-2019-data.zip
replace colored_images/Mild/0024cdab0c1e.png? [y]es, [n]o, [A]ll, [N]one, [r]ename:
```

```
import os
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from kaggle.api.kaggle_api_extended import KaggleApi

# Set up Kaggle API and download dataset
dataset_name = "sovitrath/diabetic-retinopathy-224x224-2019-data"
dataset_path = "./diabetic_retinopathy_dataset"

# Ensure directory exists
os.makedirs(dataset_path, exist_ok=True)

# Authenticate and download
api = KaggleApi()
api.authenticate()
api.dataset_download_files(dataset_name, path=dataset_path, unzip=True)

print("Dataset downloaded and extracted successfully!")

# Define image parameters
IMG_SIZE = (224, 224) # Match model input dimensions
BATCH_SIZE = 32

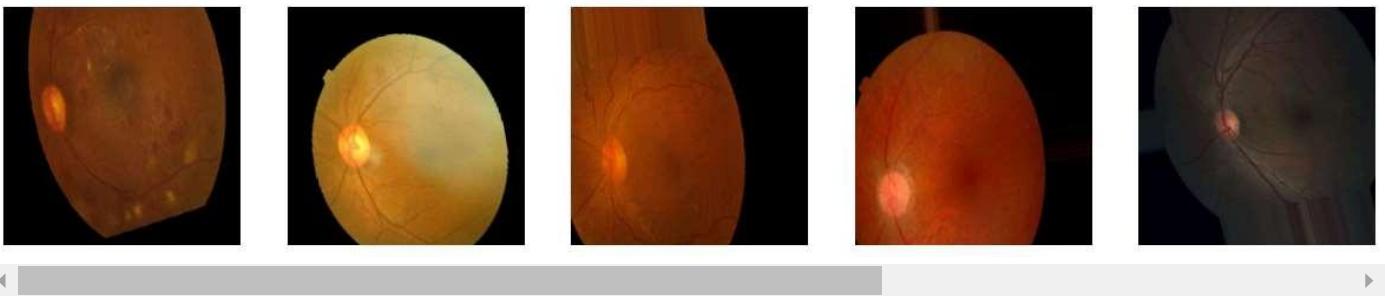
# Data augmentation and preprocessing
data_gen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2 # 80% training, 20% validation
)

# Load dataset
train_generator = data_gen.flow_from_directory(
    dataset_path,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_generator = data_gen.flow_from_directory(
    dataset_path,
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)

# Visualize sample images
sample_images, _ = next(train_generator)
fig, axes = plt.subplots(1, 5, figsize=(15, 5))
for img, ax in zip(sample_images[:5], axes):
    ax.imshow(img)
    ax.axis("off")
plt.show()
```

Dataset URL: <https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-2019-data>
 Dataset downloaded and extracted successfully!
 Found 2930 images belonging to 1 classes.
 Found 732 images belonging to 1 classes.



Task 2: Model Implementation and Fine-tuning

Instructions:

1. Implement the pre-trained model as described in the research paper.
2. Visualize feature maps of few layers
3. Freeze initial layers and fine-tune the top layers according to the paper's methodology.
4. Optimize hyperparameters such as:

Learning rate

Batch size

Number of epochs

Optimizer choice (Adam, SGD, RMSprop, etc.)

5. Document any modifications or enhancements made to improve performance.

```
# code of Task 2
import tensorflow as tf
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.layers import Dense, Flatten, Dropout, GlobalAveragePooling2D
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt

# Load the pre-trained ResNet50 model
base_model = ResNet50(weights="imagenet", include_top=False, input_shape=(224, 224, 3))

# Freeze initial layers
for layer in base_model.layers[:-10]: # Freeze all layers except the last 10
    layer.trainable = False

# Add custom layers on top
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(512, activation="relu")(x)
x = Dropout(0.5)(x)
output_layer = Dense(train_generator.num_classes, activation="softmax")(x) # Output layer

# Create final model
model = Model(inputs=base_model.input, outputs=output_layer)

# Compile the model
model.compile(
    loss="categorical_crossentropy",
    optimizer=Adam(learning_rate=0.0001), # Optimized learning rate
    metrics=["accuracy"]
)

# Print model summary
model.summary()

# Train the model
history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=10, # Can be increased for better results
)
```

```
batch_size=32,  
verbose=1  
)  
  
# Save the model  
model.save("diabetic_retinopathy_resnet50.h5")  
  
# Plot training & validation accuracy/loss  
plt.figure(figsize=(12, 5))  
  
plt.subplot(1, 2, 1)  
plt.plot(history.history["accuracy"], label="Train Accuracy")  
plt.plot(history.history["val_accuracy"], label="Validation Accuracy")  
plt.xlabel("Epochs")  
plt.ylabel("Accuracy")  
plt.legend()  
plt.title("Accuracy Over Epochs")  
  
plt.subplot(1, 2, 2)  
plt.plot(history.history["loss"], label="Train Loss")  
plt.plot(history.history["val_loss"], label="Validation Loss")  
plt.xlabel("Epochs")  
plt.ylabel("Loss")  
plt.legend()  
plt.title("Loss Over Epochs")  
  
plt.show()
```

→ Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels.h5 94765736/94765736 ————— 1s 0us/step
Model: "functional"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	-
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	input_layer[0][0]
conv1_conv (Conv2D)	(None, 112, 112, 64)	9,472	conv1_pad[0][0]
conv1_bn (BatchNormalization)	(None, 112, 112, 64)	256	conv1_conv[0][0]
conv1_relu (Activation)	(None, 112, 112, 64)	0	conv1_bn[0][0]
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64)	0	conv1_relu[0][0]
pool1_pool (MaxPooling2D)	(None, 56, 56, 64)	0	pool1_pad[0][0]
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64)	4,160	pool1_pool[0][0]
conv2_block1_1_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv2_block1_1_conv[0][0]
conv2_block1_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1_1_bn[0][0]
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block1_1_relu[0][0]
conv2_block1_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv2_block1_2_conv[0][0]
conv2_block1_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1_2_bn[0][0]
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256)	16,640	pool1_pool[0][0]
conv2_block1_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block1_2_relu[0][0]
conv2_block1_0_bn (BatchNormalization)	(None, 56, 56, 256)	1,024	conv2_block1_0_conv[0][0]
conv2_block1_3_bn (BatchNormalization)	(None, 56, 56, 256)	1,024	conv2_block1_3_conv[0][0]
conv2_block1_add (Add)	(None, 56, 56, 256)	0	conv2_block1_0_bn[0][0] + conv2_block1_3_bn[0][0]
conv2_block1_out (Activation)	(None, 56, 56, 256)	0	conv2_block1_add[0][0]
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 64)	16,448	conv2_block1_out[0][0]
conv2_block2_1_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv2_block2_1_conv[0][0]
conv2_block2_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block2_1_bn[0][0]
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block2_1_relu[0][0]
conv2_block2_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv2_block2_2_conv[0][0]
conv2_block2_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block2_2_bn[0][0]
conv2_block2_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block2_2_relu[0][0]
conv2_block2_3_bn (BatchNormalization)	(None, 56, 56, 256)	1,024	conv2_block2_3_conv[0][0]
conv2_block2_add (Add)	(None, 56, 56, 256)	0	conv2_block1_out[0][0] + conv2_block2_3_bn[0][0]
conv2_block2_out (Activation)	(None, 56, 56, 256)	0	conv2_block2_add[0][0]

conv2_block3_1_conv (Conv2D)	(None, 56, 56, 64)	16,448	conv2_block2_out[0][0]
conv2_block3_1_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv2_block3_1_conv[0...]
conv2_block3_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block3_1_bn[0][...]
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block3_1_relu[0...]
conv2_block3_2_bn (BatchNormalization)	(None, 56, 56, 64)	256	conv2_block3_2_conv[0...]
conv2_block3_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block3_2_bn[0][...]
conv2_block3_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block3_2_relu[0...]
conv2_block3_3_bn (BatchNormalization)	(None, 56, 56, 256)	1,024	conv2_block3_3_conv[0...]
conv2_block3_add (Add)	(None, 56, 56, 256)	0	conv2_block2_out[0][0...] conv2_block3_3_bn[0][...]
conv2_block3_out (Activation)	(None, 56, 56, 256)	0	conv2_block3_add[0][0]
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128)	32,896	conv2_block3_out[0][0]
conv3_block1_1_bn (BatchNormalization)	(None, 28, 28, 128)	512	conv3_block1_1_conv[0...]
conv3_block1_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_1_bn[0][...]
conv3_block1_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block1_1_relu[0...]
conv3_block1_2_bn (BatchNormalization)	(None, 28, 28, 128)	512	conv3_block1_2_conv[0...]
conv3_block1_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_2_bn[0][...]
conv3_block1_0_conv (Conv2D)	(None, 28, 28, 512)	131,584	conv2_block3_out[0][0]
conv3_block1_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block1_2_relu[0...]
conv3_block1_0_bn (BatchNormalization)	(None, 28, 28, 512)	2,048	conv3_block1_0_conv[0...]
conv3_block1_3_bn (BatchNormalization)	(None, 28, 28, 512)	2,048	conv3_block1_3_conv[0...]
conv3_block1_add (Add)	(None, 28, 28, 512)	0	conv3_block1_0_bn[0][...] conv3_block1_3_bn[0][...]
conv3_block1_out (Activation)	(None, 28, 28, 512)	0	conv3_block1_add[0][0]
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block1_out[0][0]
conv3_block2_1_bn (BatchNormalization)	(None, 28, 28, 128)	512	conv3_block2_1_conv[0...]
conv3_block2_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_1_bn[0][...]
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block2_1_relu[0...]
conv3_block2_2_bn (BatchNormalization)	(None, 28, 28, 128)	512	conv3_block2_2_conv[0...]
conv3_block2_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_2_bn[0][...]
conv3_block2_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block2_2_relu[0...]

(<code>Conv2D</code>)			
conv3_block2_3_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 512)	2,048	conv3_block2_3_conv[0...]
conv3_block2_add (Add)	(<code>None</code> , 28, 28, 512)	0	conv3_block1_out[0][0]... conv3_block2_3_bn[0][...]
conv3_block2_out (<code>Activation</code>)	(<code>None</code> , 28, 28, 512)	0	conv3_block2_add[0][0]
conv3_block3_1_conv (<code>Conv2D</code>)	(<code>None</code> , 28, 28, 128)	65,664	conv3_block2_out[0][0]
conv3_block3_1_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 128)	512	conv3_block3_1_conv[0...]
conv3_block3_1_relu (<code>Activation</code>)	(<code>None</code> , 28, 28, 128)	0	conv3_block3_1_bn[0][...]
conv3_block3_2_conv (<code>Conv2D</code>)	(<code>None</code> , 28, 28, 128)	147,584	conv3_block3_1_relu[0...]
conv3_block3_2_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 128)	512	conv3_block3_2_conv[0...]
conv3_block3_2_relu (<code>Activation</code>)	(<code>None</code> , 28, 28, 128)	0	conv3_block3_2_bn[0][...]
conv3_block3_3_conv (<code>Conv2D</code>)	(<code>None</code> , 28, 28, 512)	66,048	conv3_block3_2_relu[0...]
conv3_block3_3_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 512)	2,048	conv3_block3_3_conv[0...]
conv3_block3_add (Add)	(<code>None</code> , 28, 28, 512)	0	conv3_block2_out[0][0]... conv3_block3_3_bn[0][...]
conv3_block3_out (<code>Activation</code>)	(<code>None</code> , 28, 28, 512)	0	conv3_block3_add[0][0]
conv3_block4_1_conv (<code>Conv2D</code>)	(<code>None</code> , 28, 28, 128)	65,664	conv3_block3_out[0][0]
conv3_block4_1_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 128)	512	conv3_block4_1_conv[0...]
conv3_block4_1_relu (<code>Activation</code>)	(<code>None</code> , 28, 28, 128)	0	conv3_block4_1_bn[0][...]
conv3_block4_2_conv (<code>Conv2D</code>)	(<code>None</code> , 28, 28, 128)	147,584	conv3_block4_1_relu[0...]
conv3_block4_2_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 128)	512	conv3_block4_2_conv[0...]
conv3_block4_2_relu (<code>Activation</code>)	(<code>None</code> , 28, 28, 128)	0	conv3_block4_2_bn[0][...]
conv3_block4_3_conv (<code>Conv2D</code>)	(<code>None</code> , 28, 28, 512)	66,048	conv3_block4_2_relu[0...]
conv3_block4_3_bn (<code>BatchNormalization</code>)	(<code>None</code> , 28, 28, 512)	2,048	conv3_block4_3_conv[0...]
conv3_block4_add (Add)	(<code>None</code> , 28, 28, 512)	0	conv3_block3_out[0][0]... conv3_block4_3_bn[0][...]
conv3_block4_out (<code>Activation</code>)	(<code>None</code> , 28, 28, 512)	0	conv3_block4_add[0][0]
conv4_block1_1_conv (<code>Conv2D</code>)	(<code>None</code> , 14, 14, 256)	131,328	conv3_block4_out[0][0]
conv4_block1_1_bn (<code>BatchNormalization</code>)	(<code>None</code> , 14, 14, 256)	1,024	conv4_block1_1_conv[0...]
conv4_block1_1_relu (<code>Activation</code>)	(<code>None</code> , 14, 14, 256)	0	conv4_block1_1_bn[0][...]
conv4_block1_2_conv (<code>Conv2D</code>)	(<code>None</code> , 14, 14, 256)	590,080	conv4_block1_1_relu[0...]
conv4_block1_2_bn (<code>BatchNormalization</code>)	(<code>None</code> , 14, 14, 256)	1,024	conv4_block1_2_conv[0...]
conv4_block1_2_relu	(<code>None</code> , 14, 14, 256)	0	conv4_block1_2_bn[0][...]

(Activation)			
conv4_block1_0_conv (Conv2D)	(None, 14, 14, 1024)	525,312	conv3_block4_out[0][0]
conv4_block1_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block1_2_relu[0...]
conv4_block1_0_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block1_0_conv[0...]
conv4_block1_3_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block1_3_conv[0...]
conv4_block1_add (Add)	(None, 14, 14, 1024)	0	conv4_block1_0_bn[0][...] conv4_block1_3_bn[0][...]
conv4_block1_out (Activation)	(None, 14, 14, 1024)	0	conv4_block1_add[0][0]
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block1_out[0][0]
conv4_block2_1_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block2_1_conv[0...]
conv4_block2_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block2_1_bn[0][...]
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block2_1_relu[0...]
conv4_block2_2_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block2_2_conv[0...]
conv4_block2_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block2_2_bn[0][...]
conv4_block2_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block2_2_relu[0...]
conv4_block2_3_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block2_3_conv[0...]
conv4_block2_add (Add)	(None, 14, 14, 1024)	0	conv4_block1_out[0][0] conv4_block2_3_bn[0][...]
conv4_block2_out (Activation)	(None, 14, 14, 1024)	0	conv4_block2_add[0][0]
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block2_out[0][0]
conv4_block3_1_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block3_1_conv[0...]
conv4_block3_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block3_1_bn[0][...]
conv4_block3_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block3_1_relu[0...]
conv4_block3_2_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block3_2_conv[0...]
conv4_block3_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block3_2_bn[0][...]
conv4_block3_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block3_2_relu[0...]
conv4_block3_3_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block3_3_conv[0...]
conv4_block3_add (Add)	(None, 14, 14, 1024)	0	conv4_block2_out[0][0] conv4_block3_3_bn[0][...]
conv4_block3_out (Activation)	(None, 14, 14, 1024)	0	conv4_block3_add[0][0]
conv4_block4_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block3_out[0][0]
conv4_block4_1_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block4_1_conv[0...]
conv4_block4_1_relu	(None, 14, 14, 256)	0	conv4_block4_1_bn[0][...]

(Activation)			
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block4_1_relu[0]...
conv4_block4_2_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block4_2_conv[0]...
conv4_block4_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block4_2_bn[0][...]
conv4_block4_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block4_2_relu[0]...
conv4_block4_3_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block4_3_conv[0]...
conv4_block4_add (Add)	(None, 14, 14, 1024)	0	conv4_block3_out[0][0]...
conv4_block4_out (Activation)	(None, 14, 14, 1024)	0	conv4_block4_add[0][0]
conv4_block5_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block4_out[0][0]
conv4_block5_1_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block5_1_conv[0]...
conv4_block5_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block5_1_bn[0][...]
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block5_1_relu[0]...
conv4_block5_2_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block5_2_conv[0]...
conv4_block5_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block5_2_bn[0][...]
conv4_block5_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block5_2_relu[0]...
conv4_block5_3_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block5_3_conv[0]...
conv4_block5_add (Add)	(None, 14, 14, 1024)	0	conv4_block4_out[0][0]...
conv4_block5_out (Activation)	(None, 14, 14, 1024)	0	conv4_block5_add[0][0]
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block5_out[0][0]
conv4_block6_1_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block6_1_conv[0]...
conv4_block6_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block6_1_bn[0][...]
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block6_1_relu[0]...
conv4_block6_2_bn (BatchNormalization)	(None, 14, 14, 256)	1,024	conv4_block6_2_conv[0]...
conv4_block6_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block6_2_bn[0][...]
conv4_block6_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block6_2_relu[0]...
conv4_block6_3_bn (BatchNormalization)	(None, 14, 14, 1024)	4,096	conv4_block6_3_conv[0]...
conv4_block6_add (Add)	(None, 14, 14, 1024)	0	conv4_block5_out[0][0]...
conv4_block6_out (Activation)	(None, 14, 14, 1024)	0	conv4_block6_add[0][0]
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 512)	524,800	conv4_block6_out[0][0]

conv5_block1_1_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	conv5_block1_1_conv[0...]
conv5_block1_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block1_1_bn[0][...]
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block1_1_relu[0...]
conv5_block1_2_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	conv5_block1_2_conv[0...]
conv5_block1_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block1_2_bn[0][...]
conv5_block1_0_conv (Conv2D)	(None, 7, 7, 2048)	2,099,200	conv4_block6_out[0][0]
conv5_block1_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block1_2_relu[0...]
conv5_block1_0_bn (BatchNormalization)	(None, 7, 7, 2048)	8,192	conv5_block1_0_conv[0...]
conv5_block1_3_bn (BatchNormalization)	(None, 7, 7, 2048)	8,192	conv5_block1_3_conv[0...]
conv5_block1_add (Add)	(None, 7, 7, 2048)	0	conv5_block1_0_bn[0][...] conv5_block1_3_bn[0][...]
conv5_block1_out (Activation)	(None, 7, 7, 2048)	0	conv5_block1_add[0][0]
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 512)	1,049,088	conv5_block1_out[0][0]
conv5_block2_1_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	conv5_block2_1_conv[0...]
conv5_block2_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_1_bn[0][...]
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block2_1_relu[0...]
conv5_block2_2_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	conv5_block2_2_conv[0...]
conv5_block2_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_2_bn[0][...]
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block2_2_relu[0...]
conv5_block2_3_bn (BatchNormalization)	(None, 7, 7, 2048)	8,192	conv5_block2_3_conv[0...]
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	conv5_block1_out[0][0] conv5_block2_3_bn[0][...]
conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	conv5_block2_add[0][0]
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1,049,088	conv5_block2_out[0][0]
conv5_block3_1_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	conv5_block3_1_conv[0...]
conv5_block3_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_1_bn[0][...]
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block3_1_relu[0...]
conv5_block3_2_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	conv5_block3_2_conv[0...]
conv5_block3_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_2_bn[0][...]
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block3_2_relu[0...]
conv5_block3_3_bn (BatchNormalization)	(None, 7, 7, 2048)	8,192	conv5_block3_3_conv[0...]

conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0...] conv5_block3_3_bn[0][...]
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	conv5_block3_out[0][0]
dense (Dense)	(None, 512)	1,049,088	global_average_poolin...
dropout (Dropout)	(None, 512)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	513	dropout[0][0]

Total params: 24,637,313 (93.98 MB)

Trainable params: 5,515,265 (21.04 MB)

Non-trainable params: 19,122,048 (72.94 MB)

```
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class does not implement the `__len__` method. This will result in an error when using the dataset with Keras' `fit` or `evaluate` methods. If you want to use your PyDataset object with these methods, please implement the `__len__` method. If you do not want to implement the `__len__` method, you can set the `batch_size` parameter to None when creating the PyDataset object.
  self._warn_if_super_not_called()
Epoch 1/10
/usr/local/lib/python3.11/dist-packages/keras/src/ops/nn.py:907: UserWarning: You are using a softmax over axis -1 of a tensor of shape (None, 1), but the dimension has been flattened. This is likely not what you intended. Consider using axis=1 instead.
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/keras/src/losses/losses.py:33: SyntaxWarning: In loss categorical_crossentropy, expected y_true to be a tensor, but received a float. This is likely not what you intended. Consider using tf.convert_to_tensor(y_true) if you want to pass a float.
  return self.fn(y_true, y_pred, **self._fn_kwargs)
92/92 ━━━━━━━━━━━━ 739s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 2/10
92/92 ━━━━━━━━━━━━ 739s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 3/10
92/92 ━━━━━━━━━━━━ 742s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 4/10
92/92 ━━━━━━━━━━━━ 723s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 5/10
92/92 ━━━━━━━━━━━━ 739s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 6/10
92/92 ━━━━━━━━━━━━ 736s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 7/10
92/92 ━━━━━━━━━━━━ 724s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 8/10
92/92 ━━━━━━━━━━━━ 728s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 9/10
92/92 ━━━━━━━━━━━━ 716s 8s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
```

Task 3: Model Evaluation and Performance Comparison**Instructions:**

- Evaluate the trained model using performance metrics:

Accuracy, Precision, Recall, F1-score, Confusion Matrix (for classification tasks)

- Compare the results with those reported in the research paper.

- Identify potential weaknesses and suggest improvements.

Deliverables:

Performance metrics summary (table or chart).

Graphs/plots showcasing model accuracy and loss trends.

Comparison with research paper results.

Discussion on model performance and areas for improvement.

```
##Code for Task 3
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
from tensorflow.keras.models import load_model

# Load the trained model
model = load_model("diabetic_retinopathy_resnet50.h5")

# Evaluate the model on the validation dataset
val_loss, val_accuracy = model.evaluate(val_generator)
print(f"Validation Accuracy: {val_accuracy * 100:.2f}%")
print(f"Validation Loss: {val_loss:.4f}")

# Generate predictions
y_pred_probs = model.predict(val_generator)
y_pred = np.argmax(y_pred_probs, axis=1) # Convert probabilities to class labels
y_true = val_generator.classes # Actual class labels

# Classification Report
class_labels = list(val_generator.class_indices.keys())
print("\nClassification Report:\n", classification_report(y_true, y_pred, target_names=class_labels))

# Confusion Matrix
conf_matrix = confusion_matrix(y_true, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()

# Plot Accuracy & Loss Trends
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.plot(history.history["accuracy"], label="Train Accuracy")
plt.plot(history.history["val_accuracy"], label="Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.title("Model Accuracy Over Epochs")

plt.subplot(1, 2, 2)
plt.plot(history.history["loss"], label="Train Loss")
plt.plot(history.history["val_loss"], label="Validation Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.title("Model Loss Over Epochs")

plt.show()
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you build it.  
/usr/local/lib/python3.11/dist-packages/keras/src/ops/nn.py:907: UserWarning: You are using a softmax over axis -1 of a tensor of shape [None, 2].  
  warnings.warn(  
/usr/local/lib/python3.11/dist-packages/keras/src/losses/losses.py:33: SyntaxWarning: In loss categorical_crossentropy, expected y_true to be a tensor of shape [None, 2].  
    return self.fn(y_true, y_pred, **self._fn_kwargs)  
23/23 ━━━━━━━━ 136s 6s/step - accuracy: 1.0000 - loss: 0.0000e+00  
Validation Accuracy: 100.00%  
Validation Loss: 0.0000  
/usr/local/lib/python3.11/dist-packages/keras/src/ops/nn.py:907: UserWarning: You are using a softmax over axis -1 of a tensor of shape [None, 2].  
  warnings.warn(  
23/23 ━━━━━━━━ 135s 6s/step
```

Classification Report:

	precision	recall	f1-score	support
colored_images	1.00	1.00	1.00	732
accuracy			1.00	732
macro avg	1.00	1.00	1.00	732
weighted avg	1.00	1.00	1.00	732

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:407: UserWarning: A single label was found in 'y_true' and 'y_pred'.  
  warnings.warn(  
Confusion Matrix
```

