Transfer Learning-Based Poultry Disease Classifier

Overview

This project applies transfer learning using the EfficientNetB0 architecture to classify poultry diseases into four categories: Salmonella, Newcastle Disease, Coccidiosis, and Healthy. The final model is intended for integration into a mobile application to assist farmers in early disease detection.

Environment Setup

STEP 1: Install Required Libraries

```
!pip install -q tensorflow matplotlib kaggle
```

STEP 2: Import Libraries

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import EfficientNetB0
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt
import os
import shutil
```

Data Access and Preparation

STEP 3-4: Upload and Configure Kaggle API

```
from google.colab import files
files.upload()

!mkdir -p ~/.kaggle
!mv kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

STEP 5: Download and Unzip Dataset

```
!kaggle datasets download -d allandclive/chicken-disease-1
!unzip -o chicken-disease-1.zip -d /content/poultry_dataset
```

STEP 6: Restructure Image Folders

```
src_folder = '/content/poultry_dataset/Train'
for filename in os.listdir(src_folder):
    if not filename.lower().endswith(('.jpg', '.jpeg', '.png')):
        continue
    label = filename.split('.')[0].lower()
    class_dir = os.path.join(src_folder, label)
```

```
    os.makedirs(class_dir, exist_ok=True)
    shutil.move(os.path.join(src_folder, filename), os.path.join(class_dir, filename))
```

Data Preprocessing

```
IMAGE_SIZE = (224, 224)
BATCH_SIZE = 32

datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,
    horizontal_flip=True,
    zoom_range=0.2,
    shear_range=0.2
)

train_gen = datagen.flow_from_directory(
    '/content/poultry_dataset/Train',
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

val_gen = datagen.flow_from_directory(
    '/content/poultry_dataset/Train',
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

Model Architecture

```
base_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dropout(0.5)(x)
predictions = Dense(train_gen.num_classes, activation='softmax')(x)

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

for layer in base_model.layers:
    layer.trainable = False
```

Compilation and Training

Compile the model

```
model.compile(
```

```
    optimizer=Adam(learning_rate=1e-4),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```

Train the model

```
history = model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=10
)
```

Performance Visualization

```
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.title('Training vs Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```

Future Work and Deployment

- Convert the trained model to TensorFlow Lite using tf.lite.TFLiteConverter.
- Deploy in a mobile app (e.g., Flutter or Android Studio).
- Allow users to upload or capture poultry images to get disease predictions.