Project 3 Analysis

Setup

In [ ]:

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.efficientnet import EfficientNetB0, preprocess_i

train_dir = '/home/hanyan/dev/4002/train'
test_dir = '/home/hanyan/dev/4002/test'

train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    validation_split=0.2
)

test_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input
)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    subset='validation'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical',
    shuffle=False
)
```

```
2024-04-22 10:10:49.116466: I tensorflow/core/util/port.cc:110] oneDNN custom operat
ions are on. You may see slightly different numerical results due to floating-point
round-off errors from different computation orders. To turn them off, set the enviro
nment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-04-22 10:10:49.118815: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find
cuda drivers on your machine, GPU will not be used.
2024-04-22 10:10:49.149140: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find
cuda drivers on your machine, GPU will not be used.
2024-04-22 10:10:49.149878: I tensorflow/core/platform/cpu_feature_guard.cc:182] Thi
s TensorFlow binary is optimized to use available CPU instructions in performance-cr
itical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations, rebuil
d TensorFlow with the appropriate compiler flags.
2024-04-22 10:10:49.683095: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38]
TF-TRT Warning: Could not find TensorRT
Found 153 images belonging to 2 classes.
Found 37 images belonging to 2 classes.
Found 47 images belonging to 2 classes.
```

Model Config

In [ ]:

```
base_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=(224

base_model.trainable = False

model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(1024, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(2, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb0 (Functional  (None, 7, 7, 1280)        4049571
 )

 global_average_pooling2d (  (None, 1280)              0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 1024)              1311744

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 2)                 2050

=================================================================
Total params: 5363365 (20.46 MB)
Trainable params: 1313794 (5.01 MB)
Non-trainable params: 4049571 (15.45 MB)
_________________________________________________________________
```

Model training

In [ ]:

```
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // validation_generator.batch_siz
)
```

```
Epoch 1/10
4/4 [==============================] - 6s 841ms/step - loss: 0.8993 - accuracy: 0.54
55 - val_loss: 0.3464 - val_accuracy: 0.8438
Epoch 2/10
4/4 [==============================] - 2s 594ms/step - loss: 0.2521 - accuracy: 0.90
91 - val_loss: 0.2171 - val_accuracy: 0.8750
Epoch 3/10
4/4 [==============================] - 2s 591ms/step - loss: 0.1704 - accuracy: 0.96
69 - val_loss: 0.0950 - val_accuracy: 0.9688
Epoch 4/10
4/4 [==============================] - 2s 580ms/step - loss: 0.0602 - accuracy: 0.99
17 - val_loss: 0.0623 - val_accuracy: 1.0000
Epoch 5/10
4/4 [==============================] - 2s 585ms/step - loss: 0.0642 - accuracy: 0.98
44 - val_loss: 0.0292 - val_accuracy: 1.0000
Epoch 6/10
4/4 [==============================] - 2s 545ms/step - loss: 0.0936 - accuracy: 0.98
35 - val_loss: 0.1772 - val_accuracy: 0.9062
Epoch 7/10
4/4 [==============================] - 2s 613ms/step - loss: 0.0885 - accuracy: 0.95
87 - val_loss: 0.0257 - val_accuracy: 1.0000
Epoch 8/10
4/4 [==============================] - 2s 577ms/step - loss: 0.0178 - accuracy: 1.00
00 - val_loss: 0.0180 - val_accuracy: 1.0000
Epoch 9/10
4/4 [==============================] - 2s 585ms/step - loss: 0.0279 - accuracy: 1.00
00 - val_loss: 0.0744 - val_accuracy: 0.9375
Epoch 10/10
4/4 [==============================] - 2s 539ms/step - loss: 0.0181 - accuracy: 1.00
00 - val_loss: 0.0917 - val_accuracy: 0.9375
```

Model Eval

In [ ]:

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")
```

```
2/2 [==============================] - 1s 165ms/step - loss: 0.1051 - accuracy: 0.95
74
Test Accuracy: 95.74%
```