new-dl-assignment-02-4

March 24, 2025

1 Google Colab Lab Assignment -Pretrained Modle

Course Name: Deep Learning

Lab Title: Tomato crop disease classification using pre-trained deep learning algorithm

Student Name: Sapna Dahikamble

Student ID: 202201070065

Date of Submission:[24/02/2025]

Group Members:
Supriya Maskar (202201040049)

Manjiri Netankar (202201040206)

Sapna Dahikamble(202201070065)

Research Paper Study and Implementation

1.1 Project Resources

- **Dataset Link:**
  Tomato Leaf Dataset on Kaggle
- **Colab Notebook Link:**
  Open Colab Notebook
- **Research Paper Link:**
  Research Paper on ScienceDirect
- **GitHub Link:**
  View Project on GitHub

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. 3. Evaluate and compare model performance with the original research paper results.**

Modifications and Steps Performed:

1.Research Paper Selection

1.A research paper that implemented transfer learning with CNN models for plant disease classification was selected.

2.The paper applied pre-trained models like VGG, AlexNet, and EfficientNet for feature extraction and classification.

2.Dataset Identification and Description

Dataset Name: Tomato Leaf Disease Dataset

Link to Dataset: Tomato Leaf Dataset Link (https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf)

Description:

Contains multiple classes of tomato leaf diseases.

Divided into training, validation, and test sets for model evaluation.

3. Image Resizing and Preprocessing All images were resized to:

AlexNet: 128x128 pixels

VGG16: 64x64 pixels

InceptionV3: 100x100 pixels

Image pixel values were normalized to scale between 0 and 1.

Step 1: Verify and Extract Dataset

[ ]:
```
import zipfile
import os

# Define correct path to the ZIP file
zip_path = "/content/archive (12) (3).zip"
extract_path = "/content/tomato_dataset"

#  Check if the file exists
if not os.path.exists(zip_path):
    raise FileNotFoundError(f" Error: File not found at path {zip_path}. Please␣
 ↪check and upload again.")

#  Verify if the file is a valid ZIP
if not zipfile.is_zipfile(zip_path):
    raise zipfile.BadZipFile(" Error: The file is NOT a valid ZIP file. Please␣
 ↪re-upload a proper ZIP file.")
```

```python
# Extract the zip file after verification
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)
print(" File extracted successfully!")

#  Define correct paths after extraction
train_dir = os.path.join(extract_path, "tomato/train")
val_dir = os.path.join(extract_path, "tomato/val")

#  Verify dataset structure
if os.path.exists(train_dir) and os.path.exists(val_dir):
    print(" Dataset verified. Structure is correct.")
    print("Train Classes:", os.listdir(train_dir))
    print("Validation Classes:", os.listdir(val_dir))
else:
    raise FileNotFoundError(" Error: Dataset directories not found. Check␣
↪extracted paths.")
```

```
 File extracted successfully!
 Dataset verified. Structure is correct.
Train Classes: ['Tomato___Septoria_leaf_spot', 'Tomato___Early_blight',
'Tomato___healthy', 'Tomato___Bacterial_spot', 'Tomato___Tomato_mosaic_virus',
'Tomato___Late_blight', 'Tomato___Spider_mites Two-spotted_spider_mite',
'Tomato___Tomato_Yellow_Leaf_Curl_Virus', 'Tomato___Target_Spot',
'Tomato___Leaf_Mold']
Validation Classes: ['Tomato___Septoria_leaf_spot', 'Tomato___Early_blight',
'Tomato___healthy', 'Tomato___Bacterial_spot', 'Tomato___Tomato_mosaic_virus',
'Tomato___Late_blight', 'Tomato___Spider_mites Two-spotted_spider_mite',
'Tomato___Tomato_Yellow_Leaf_Curl_Virus', 'Tomato___Target_Spot',
'Tomato___Leaf_Mold']
```

Step 2: Load Dataset and Apply Preprocessing

[ ]:
```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define optimized image sizes and batch size
IMG_SIZE_ALEXNET = (128, 128)  # Optimized size for AlexNet
IMG_SIZE_VGG16 = (64, 64)      # Optimized size for VGG16
IMG_SIZE_INCEPTION = (100, 100)  # Optimized size for InceptionV3
BATCH_SIZE = 8  # Reduced batch size for faster training

#   Image Augmentation for Training
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    shear_range=0.2,
    zoom_range=0.2,
```

```
    horizontal_flip=True
)

val_datagen = ImageDataGenerator(rescale=1./255)
```

[ ]:
```
# Load Validation Data for All Models
val_generator_alexnet = val_datagen.flow_from_directory(
    val_dir, target_size=IMG_SIZE_ALEXNET, batch_size=BATCH_SIZE,␣
 ↪class_mode='categorical'
)

val_generator_vgg16 = val_datagen.flow_from_directory(
    val_dir, target_size=IMG_SIZE_VGG16, batch_size=BATCH_SIZE,␣
 ↪class_mode='categorical'
)

val_generator_inception = val_datagen.flow_from_directory(
    val_dir, target_size=IMG_SIZE_INCEPTION, batch_size=BATCH_SIZE,␣
 ↪class_mode='categorical'
)
```

```
Found 1000 images belonging to 10 classes.
Found 1000 images belonging to 10 classes.
Found 1000 images belonging to 10 classes.
```

[ ]:
```
# Load Training Data for All Models
train_generator_alexnet = train_datagen.flow_from_directory(
    train_dir, target_size=IMG_SIZE_ALEXNET, batch_size=BATCH_SIZE,␣
 ↪class_mode='categorical'
)

train_generator_vgg16 = train_datagen.flow_from_directory(
    train_dir, target_size=IMG_SIZE_VGG16, batch_size=BATCH_SIZE,␣
 ↪class_mode='categorical'
)

train_generator_inception = train_datagen.flow_from_directory(
    train_dir, target_size=IMG_SIZE_INCEPTION, batch_size=BATCH_SIZE,␣
 ↪class_mode='categorical'
)
```

```
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
Found 10000 images belonging to 10 classes.
Found 1000 images belonging to 10 classes.
Found 1000 images belonging to 10 classes.
Found 1000 images belonging to 10 classes.
```

[ ]:
```python
# Step 3: Define and Compile Models

from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
↪Dropout, BatchNormalization, GlobalAveragePooling2D
from tensorflow.keras.applications import VGG16, InceptionV3
```

[ ]:
```python
#  AlexNet Model
def create_alexnet(input_shape=(128, 128, 3), num_classes=10):
    model = Sequential([
        Conv2D(96, (11, 11), strides=(4, 4), activation='relu',␣
↪input_shape=input_shape),
        MaxPooling2D(pool_size=(3, 3), strides=(2, 2)),
        BatchNormalization(),
        Conv2D(256, (5, 5), padding="same", activation='relu'),
        MaxPooling2D(pool_size=(3, 3), strides=(2, 2)),
        BatchNormalization(),
        Conv2D(384, (3, 3), padding="same", activation='relu'),
        Conv2D(384, (3, 3), padding="same", activation='relu'),
        Conv2D(256, (3, 3), padding="same", activation='relu'),
        MaxPooling2D(pool_size=(3, 3), strides=(2, 2)),
        Flatten(),
        Dense(1024, activation='relu'),
        Dropout(0.5),
        Dense(512, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    return model
```

[ ]:
```python
#  Compile AlexNet
alexnet_model = create_alexnet()
alexnet_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])
```

Step 3: Define and Compile Models

[ ]:
```python
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,␣
↪Dropout, BatchNormalization, GlobalAveragePooling2D
from tensorflow.keras.applications import VGG16, InceptionV3
```

[ ]:
```python
# AlexNet Model
def create_alexnet(input_shape=(128, 128, 3), num_classes=10):
    model = Sequential([
        Conv2D(96, (11, 11), strides=(4, 4), activation='relu',␣
↪input_shape=input_shape),
```

```
        MaxPooling2D(pool_size=(3, 3), strides=(2, 2)),
        BatchNormalization(),
        Conv2D(256, (5, 5), padding="same", activation='relu'),
        MaxPooling2D(pool_size=(3, 3), strides=(2, 2)),
        BatchNormalization(),
        Conv2D(384, (3, 3), padding="same", activation='relu'),
        Conv2D(384, (3, 3), padding="same", activation='relu'),
        Conv2D(256, (3, 3), padding="same", activation='relu'),
        MaxPooling2D(pool_size=(3, 3), strides=(2, 2)),
        Flatten(),
        Dense(1024, activation='relu'),
        Dropout(0.5),
        Dense(512, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    return model
```

[ ]:
```
#  Compile and Show AlexNet Summary
alexnet_model = create_alexnet()
alexnet_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
print(" AlexNet Model Summary:")
alexnet_model.summary()  #  Show AlexNet Summary
```

```
 AlexNet Model Summary:
```

Model: "sequential_3"

```
 Layer (type)                        Output Shape                             ␣
↪Param #

 conv2d_203 (Conv2D)                 (None, 30, 30, 96)                        ␣
↪34,944

 max_pooling2d_17 (MaxPooling2D)     (None, 14, 14, 96)                           ␣
↪  0

 batch_normalization_194             (None, 14, 14, 96)                           ␣
↪384
 (BatchNormalization)                                                             ␣
↪

 conv2d_204 (Conv2D)                 (None, 14, 14, 256)                      ␣
↪614,656
```

```
max_pooling2d_18 (MaxPooling2D)       (None, 6, 6, 256)
↪  0

batch_normalization_195               (None, 6, 6, 256)
↪1,024
(BatchNormalization)
↪

conv2d_205 (Conv2D)                   (None, 6, 6, 384)
↪885,120

conv2d_206 (Conv2D)                   (None, 6, 6, 384)
↪1,327,488

conv2d_207 (Conv2D)                   (None, 6, 6, 256)
↪884,992

max_pooling2d_19 (MaxPooling2D)       (None, 2, 2, 256)
↪  0

flatten_3 (Flatten)                   (None, 1024)
↪  0

dense_15 (Dense)                      (None, 1024)
↪1,049,600

dropout_9 (Dropout)                   (None, 1024)
↪  0

dense_16 (Dense)                      (None, 512)
↪524,800

dropout_10 (Dropout)                  (None, 512)
↪  0

dense_17 (Dense)                      (None, 10)
↪5,130
```

Total params: 5,328,138 (20.33 MB)

Trainable params: 5,327,434 (20.32 MB)

Non-trainable params: 704 (2.75 KB)

[ ]:
```python
#  VGG16 Model
base_model_vgg16 = VGG16(weights='imagenet', include_top=False,␣
 ↪input_shape=(64, 64, 3))
for layer in base_model_vgg16.layers[:-4]:
    layer.trainable = False
x = GlobalAveragePooling2D()(base_model_vgg16.output)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(10, activation='softmax')(x)
vgg16_model = Model(inputs=base_model_vgg16.input, outputs=output)
```

[ ]:
```python
#  Compile and Show VGG16 Summary
vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
print("\n VGG16 Model Summary:")
vgg16_model.summary()  #  Show VGG16 Summary
```

```
 VGG16 Model Summary:
```

Model: "functional_7"

```
Layer (type)                          Output Shape                          ␣
↪Param #

input_layer_8 (InputLayer)            (None, 64, 64, 3)                         ␣
↪   0

block1_conv1 (Conv2D)                 (None, 64, 64, 64)                      ␣
↪1,792

block1_conv2 (Conv2D)                 (None, 64, 64, 64)                     ␣
↪36,928

block1_pool (MaxPooling2D)            (None, 32, 32, 64)                        ␣
↪   0

block2_conv1 (Conv2D)                 (None, 32, 32, 128)                    ␣
↪73,856

block2_conv2 (Conv2D)                 (None, 32, 32, 128)                   ␣
↪147,584

block2_pool (MaxPooling2D)            (None, 16, 16, 128)                       ␣
↪   0
```

```
block3_conv1 (Conv2D)              (None, 16, 16, 256)            ␣
↪295,168

block3_conv2 (Conv2D)              (None, 16, 16, 256)            ␣
↪590,080

block3_conv3 (Conv2D)              (None, 16, 16, 256)            ␣
↪590,080

block3_pool (MaxPooling2D)         (None, 8, 8, 256)                  ␣
↪  0

block4_conv1 (Conv2D)              (None, 8, 8, 512)            ␣
↪1,180,160

block4_conv2 (Conv2D)              (None, 8, 8, 512)            ␣
↪2,359,808

block4_conv3 (Conv2D)              (None, 8, 8, 512)            ␣
↪2,359,808

block4_pool (MaxPooling2D)         (None, 4, 4, 512)                  ␣
↪  0

block5_conv1 (Conv2D)              (None, 4, 4, 512)            ␣
↪2,359,808

block5_conv2 (Conv2D)              (None, 4, 4, 512)            ␣
↪2,359,808

block5_conv3 (Conv2D)              (None, 4, 4, 512)            ␣
↪2,359,808

block5_pool (MaxPooling2D)         (None, 2, 2, 512)                  ␣
↪  0

global_average_pooling2d_3         (None, 512)                        ␣
↪  0
(GlobalAveragePooling2D)                                              ␣
↪

dense_18 (Dense)                   (None, 128)                    ␣
↪65,664

dropout_11 (Dropout)               (None, 128)                        ␣
↪  0
```

```
 dense_19 (Dense)                      (None, 10)                             ␣
↪1,290
```

Total params: 14,781,642 (56.39 MB)

Trainable params: 7,146,378 (27.26 MB)

Non-trainable params: 7,635,264 (29.13 MB)

[ ]:
```python
#  InceptionV3 Model
base_model_inception = InceptionV3(weights='imagenet', include_top=False,␣
 ↪input_shape=(100, 100, 3))
for layer in base_model_inception.layers[:-4]:
    layer.trainable = False
x = GlobalAveragePooling2D()(base_model_inception.output)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(10, activation='softmax')(x)
inception_model = Model(inputs=base_model_inception.input, outputs=output)
```

[ ]:
```python
#  Compile and Show InceptionV3 Summary
inception_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])
print("\n InceptionV3 Model Summary:")
inception_model.summary()  # Show InceptionV3 Summary
```

```
 InceptionV3 Model Summary:
```

Model: "functional_8"

```
 Layer (type)                 Output Shape               Param #   Connected␣
↪to

 input_layer_9                (None, 100, 100, 3)              0   -          ␣
↪
 (InputLayer)                                                                 ␣
↪

 conv2d_208 (Conv2D)          (None, 49, 49, 32)             864 ␣
↪input_layer_9[0][0]
```

```
batch_normalization_196     (None, 49, 49, 32)            96 ␣
↪conv2d_208[0][0]
(BatchNormalization)                                                 ␣
↪

activation_188              (None, 49, 49, 32)             0 ␣
↪batch_normalization_1…
(Activation)                                                         ␣
↪

conv2d_209 (Conv2D)         (None, 47, 47, 32)         9,216 ␣
↪activation_188[0][0]

batch_normalization_197     (None, 47, 47, 32)            96 ␣
↪conv2d_209[0][0]
(BatchNormalization)                                                 ␣
↪

activation_189              (None, 47, 47, 32)             0 ␣
↪batch_normalization_1…
(Activation)                                                         ␣
↪

conv2d_210 (Conv2D)         (None, 47, 47, 64)        18,432 ␣
↪activation_189[0][0]

batch_normalization_198     (None, 47, 47, 64)           192 ␣
↪conv2d_210[0][0]
(BatchNormalization)                                                 ␣
↪

activation_190              (None, 47, 47, 64)             0 ␣
↪batch_normalization_1…
(Activation)                                                         ␣
↪

max_pooling2d_20            (None, 23, 23, 64)             0 ␣
↪activation_190[0][0]
(MaxPooling2D)                                                       ␣
↪

conv2d_211 (Conv2D)         (None, 23, 23, 80)         5,120 ␣
↪max_pooling2d_20[0][0]

batch_normalization_199     (None, 23, 23, 80)           240 ␣
↪conv2d_211[0][0]
```

```
(BatchNormalization)                                                          ␣
↪

activation_191              (None, 23, 23, 80)               0 ␣
↪batch_normalization_1…
(Activation)                                                                  ␣
↪

conv2d_212 (Conv2D)         (None, 21, 21, 192)        138,240 ␣
↪activation_191[0][0]

batch_normalization_200     (None, 21, 21, 192)            576 ␣
↪conv2d_212[0][0]
(BatchNormalization)                                                          ␣
↪

activation_192              (None, 21, 21, 192)              0 ␣
↪batch_normalization_2…
(Activation)                                                                  ␣
↪

max_pooling2d_21            (None, 10, 10, 192)              0 ␣
↪activation_192[0][0]
(MaxPooling2D)                                                                ␣
↪

conv2d_216 (Conv2D)         (None, 10, 10, 64)          12,288 ␣
↪max_pooling2d_21[0][0]

batch_normalization_204     (None, 10, 10, 64)             192 ␣
↪conv2d_216[0][0]
(BatchNormalization)                                                          ␣
↪

activation_196              (None, 10, 10, 64)               0 ␣
↪batch_normalization_2…
(Activation)                                                                  ␣
↪

conv2d_214 (Conv2D)         (None, 10, 10, 48)           9,216 ␣
↪max_pooling2d_21[0][0]

conv2d_217 (Conv2D)         (None, 10, 10, 96)          55,296 ␣
↪activation_196[0][0]
```

```
batch_normalization_202     (None, 10, 10, 48)            144 ␣
↪conv2d_214[0][0]
(BatchNormalization)                                                    ␣
↪

batch_normalization_205     (None, 10, 10, 96)            288 ␣
↪conv2d_217[0][0]
(BatchNormalization)                                                    ␣
↪

activation_194              (None, 10, 10, 48)              0 ␣
↪batch_normalization_2…
(Activation)                                                            ␣
↪

activation_197              (None, 10, 10, 96)              0 ␣
↪batch_normalization_2…
(Activation)                                                            ␣
↪

average_pooling2d_18        (None, 10, 10, 192)             0 ␣
↪max_pooling2d_21[0][0]
(AveragePooling2D)                                                      ␣
↪

conv2d_213 (Conv2D)         (None, 10, 10, 64)         12,288 ␣
↪max_pooling2d_21[0][0]

conv2d_215 (Conv2D)         (None, 10, 10, 64)         76,800 ␣
↪activation_194[0][0]

conv2d_218 (Conv2D)         (None, 10, 10, 96)         82,944 ␣
↪activation_197[0][0]

conv2d_219 (Conv2D)         (None, 10, 10, 32)          6,144 ␣
↪average_pooling2d_18[…

batch_normalization_201     (None, 10, 10, 64)            192 ␣
↪conv2d_213[0][0]
(BatchNormalization)                                                    ␣
↪

batch_normalization_203     (None, 10, 10, 64)            192 ␣
↪conv2d_215[0][0]
(BatchNormalization)                                                    ␣
↪
```

```
batch_normalization_206     (None, 10, 10, 96)          288 ␣
↪conv2d_218[0][0]
(BatchNormalization)                                                ␣
↪

batch_normalization_207     (None, 10, 10, 32)           96 ␣
↪conv2d_219[0][0]
(BatchNormalization)                                                ␣
↪

activation_193              (None, 10, 10, 64)            0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_195              (None, 10, 10, 64)            0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_198              (None, 10, 10, 96)            0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_199              (None, 10, 10, 32)            0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

mixed0 (Concatenate)        (None, 10, 10, 256)           0 ␣
↪activation_193[0][0],
                                                              ␣
↪activation_195[0][0],
                                                              ␣
↪activation_198[0][0],
                                                              ␣
↪activation_199[0][0]

conv2d_223 (Conv2D)         (None, 10, 10, 64)       16,384 ␣
↪mixed0[0][0]

batch_normalization_211     (None, 10, 10, 64)          192 ␣
↪conv2d_223[0][0]
```

```
(BatchNormalization)                                                       ␣
↪

activation_203             (None, 10, 10, 64)                    0 ␣
↪batch_normalization_2…
(Activation)                                                               ␣
↪

conv2d_221 (Conv2D)        (None, 10, 10, 48)               12,288 ␣
↪mixed0[0][0]

conv2d_224 (Conv2D)        (None, 10, 10, 96)               55,296 ␣
↪activation_203[0][0]

batch_normalization_209    (None, 10, 10, 48)                  144 ␣
↪conv2d_221[0][0]
(BatchNormalization)                                                       ␣
↪

batch_normalization_212    (None, 10, 10, 96)                  288 ␣
↪conv2d_224[0][0]
(BatchNormalization)                                                       ␣
↪

activation_201             (None, 10, 10, 48)                    0 ␣
↪batch_normalization_2…
(Activation)                                                               ␣
↪

activation_204             (None, 10, 10, 96)                    0 ␣
↪batch_normalization_2…
(Activation)                                                               ␣
↪

average_pooling2d_19       (None, 10, 10, 256)                   0 ␣
↪mixed0[0][0]
(AveragePooling2D)                                                         ␣
↪

conv2d_220 (Conv2D)        (None, 10, 10, 64)               16,384 ␣
↪mixed0[0][0]

conv2d_222 (Conv2D)        (None, 10, 10, 64)               76,800 ␣
↪activation_201[0][0]
```

```
conv2d_225 (Conv2D)         (None, 10, 10, 96)           82,944 ␣
↪activation_204[0][0]

conv2d_226 (Conv2D)         (None, 10, 10, 64)           16,384 ␣
↪average_pooling2d_19[…

batch_normalization_208     (None, 10, 10, 64)              192 ␣
↪conv2d_220[0][0]
(BatchNormalization)                                                ␣
↪

batch_normalization_210     (None, 10, 10, 64)              192 ␣
↪conv2d_222[0][0]
(BatchNormalization)                                                ␣
↪

batch_normalization_213     (None, 10, 10, 96)              288 ␣
↪conv2d_225[0][0]
(BatchNormalization)                                                ␣
↪

batch_normalization_214     (None, 10, 10, 64)              192 ␣
↪conv2d_226[0][0]
(BatchNormalization)                                                ␣
↪

activation_200              (None, 10, 10, 64)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_202              (None, 10, 10, 64)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_205              (None, 10, 10, 96)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_206              (None, 10, 10, 64)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪
```

```
mixed1 (Concatenate)          (None, 10, 10, 288)              0
activation_200[0][0],

activation_202[0][0],

activation_205[0][0],

activation_206[0][0]

conv2d_230 (Conv2D)           (None, 10, 10, 64)          18,432
mixed1[0][0]

batch_normalization_218       (None, 10, 10, 64)             192
conv2d_230[0][0]
(BatchNormalization)


activation_210                (None, 10, 10, 64)               0
batch_normalization_2…
(Activation)


conv2d_228 (Conv2D)           (None, 10, 10, 48)          13,824
mixed1[0][0]

conv2d_231 (Conv2D)           (None, 10, 10, 96)          55,296
activation_210[0][0]

batch_normalization_216       (None, 10, 10, 48)             144
conv2d_228[0][0]
(BatchNormalization)


batch_normalization_219       (None, 10, 10, 96)             288
conv2d_231[0][0]
(BatchNormalization)


activation_208                (None, 10, 10, 48)               0
batch_normalization_2…
(Activation)


activation_211                (None, 10, 10, 96)               0
batch_normalization_2…
```

```
(Activation)                                                    ␣
↪

average_pooling2d_20      (None, 10, 10, 288)            0  ␣
↪mixed1[0][0]
(AveragePooling2D)                                              ␣
↪

conv2d_227 (Conv2D)       (None, 10, 10, 64)        18,432  ␣
↪mixed1[0][0]

conv2d_229 (Conv2D)       (None, 10, 10, 64)        76,800  ␣
↪activation_208[0][0]

conv2d_232 (Conv2D)       (None, 10, 10, 96)        82,944  ␣
↪activation_211[0][0]

conv2d_233 (Conv2D)       (None, 10, 10, 64)        18,432  ␣
↪average_pooling2d_20[…

batch_normalization_215   (None, 10, 10, 64)           192  ␣
↪conv2d_227[0][0]
(BatchNormalization)                                            ␣
↪

batch_normalization_217   (None, 10, 10, 64)           192  ␣
↪conv2d_229[0][0]
(BatchNormalization)                                            ␣
↪

batch_normalization_220   (None, 10, 10, 96)           288  ␣
↪conv2d_232[0][0]
(BatchNormalization)                                            ␣
↪

batch_normalization_221   (None, 10, 10, 64)           192  ␣
↪conv2d_233[0][0]
(BatchNormalization)                                            ␣
↪

activation_207            (None, 10, 10, 64)             0  ␣
↪batch_normalization_2…
(Activation)                                                    ␣
↪
```

```
activation_209            (None, 10, 10, 64)               0 ␣
↪batch_normalization_2…                                        ␣
(Activation)
↪

activation_212            (None, 10, 10, 96)               0 ␣
↪batch_normalization_2…                                        ␣
(Activation)
↪

activation_213            (None, 10, 10, 64)               0 ␣
↪batch_normalization_2…                                        ␣
(Activation)
↪

mixed2 (Concatenate)      (None, 10, 10, 288)              0 ␣
↪activation_207[0][0],
                                                             ␣
↪activation_209[0][0],
                                                             ␣
↪activation_212[0][0],
                                                             ␣
↪activation_213[0][0]

conv2d_235 (Conv2D)       (None, 10, 10, 64)          18,432 ␣
↪mixed2[0][0]

batch_normalization_223   (None, 10, 10, 64)             192 ␣
↪conv2d_235[0][0]                                              ␣
(BatchNormalization)
↪

activation_215            (None, 10, 10, 64)               0 ␣
↪batch_normalization_2…                                        ␣
(Activation)
↪

conv2d_236 (Conv2D)       (None, 10, 10, 96)          55,296 ␣
↪activation_215[0][0]

batch_normalization_224   (None, 10, 10, 96)             288 ␣
↪conv2d_236[0][0]                                              ␣
(BatchNormalization)
↪
```

```
activation_216              (None, 10, 10, 96)              0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

conv2d_234 (Conv2D)         (None, 4, 4, 384)         995,328 ␣
↪mixed2[0][0]

conv2d_237 (Conv2D)         (None, 4, 4, 96)           82,944 ␣
↪activation_216[0][0]

batch_normalization_222     (None, 4, 4, 384)           1,152 ␣
↪conv2d_234[0][0]
(BatchNormalization)                                                      ␣
↪

batch_normalization_225     (None, 4, 4, 96)              288 ␣
↪conv2d_237[0][0]
(BatchNormalization)                                                      ␣
↪

activation_214              (None, 4, 4, 384)               0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

activation_217              (None, 4, 4, 96)                0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

max_pooling2d_22            (None, 4, 4, 288)               0 ␣
↪mixed2[0][0]
(MaxPooling2D)                                                            ␣
↪

mixed3 (Concatenate)        (None, 4, 4, 768)               0 ␣
↪activation_214[0][0],
                                                                ␣
↪activation_217[0][0],
                                                                ␣
↪max_pooling2d_22[0][0]

conv2d_242 (Conv2D)         (None, 4, 4, 128)          98,304 ␣
↪mixed3[0][0]
```

```
batch_normalization_230      (None, 4, 4, 128)            384 ␣
↪conv2d_242[0][0]
(BatchNormalization)                                                ␣
↪

activation_222               (None, 4, 4, 128)              0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

conv2d_243 (Conv2D)          (None, 4, 4, 128)        114,688 ␣
↪activation_222[0][0]

batch_normalization_231      (None, 4, 4, 128)            384 ␣
↪conv2d_243[0][0]
(BatchNormalization)                                                ␣
↪

activation_223               (None, 4, 4, 128)              0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

conv2d_239 (Conv2D)          (None, 4, 4, 128)         98,304 ␣
↪mixed3[0][0]

conv2d_244 (Conv2D)          (None, 4, 4, 128)        114,688 ␣
↪activation_223[0][0]

batch_normalization_227      (None, 4, 4, 128)            384 ␣
↪conv2d_239[0][0]
(BatchNormalization)                                                ␣
↪

batch_normalization_232      (None, 4, 4, 128)            384 ␣
↪conv2d_244[0][0]
(BatchNormalization)                                                ␣
↪

activation_219               (None, 4, 4, 128)              0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_224               (None, 4, 4, 128)              0 ␣
↪batch_normalization_2…
```

```
(Activation)                                                              ␣
↪

conv2d_240 (Conv2D)          (None, 4, 4, 128)           114,688 ␣
↪activation_219[0][0]

conv2d_245 (Conv2D)          (None, 4, 4, 128)           114,688 ␣
↪activation_224[0][0]

batch_normalization_228      (None, 4, 4, 128)               384 ␣
↪conv2d_240[0][0]
(BatchNormalization)                                                      ␣
↪

batch_normalization_233      (None, 4, 4, 128)               384 ␣
↪conv2d_245[0][0]
(BatchNormalization)                                                      ␣
↪

activation_220               (None, 4, 4, 128)                 0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

activation_225               (None, 4, 4, 128)                 0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

average_pooling2d_21         (None, 4, 4, 768)                 0 ␣
↪mixed3[0][0]
(AveragePooling2D)                                                        ␣
↪

conv2d_238 (Conv2D)          (None, 4, 4, 192)           147,456 ␣
↪mixed3[0][0]

conv2d_241 (Conv2D)          (None, 4, 4, 192)           172,032 ␣
↪activation_220[0][0]

conv2d_246 (Conv2D)          (None, 4, 4, 192)           172,032 ␣
↪activation_225[0][0]

conv2d_247 (Conv2D)          (None, 4, 4, 192)           147,456 ␣
↪average_pooling2d_21[…
```

```
batch_normalization_226    (None, 4, 4, 192)             576
conv2d_238[0][0]
(BatchNormalization)

batch_normalization_229    (None, 4, 4, 192)             576
conv2d_241[0][0]
(BatchNormalization)

batch_normalization_234    (None, 4, 4, 192)             576
conv2d_246[0][0]
(BatchNormalization)

batch_normalization_235    (None, 4, 4, 192)             576
conv2d_247[0][0]
(BatchNormalization)

activation_218             (None, 4, 4, 192)               0
batch_normalization_2…
(Activation)

activation_221             (None, 4, 4, 192)               0
batch_normalization_2…
(Activation)

activation_226             (None, 4, 4, 192)               0
batch_normalization_2…
(Activation)

activation_227             (None, 4, 4, 192)               0
batch_normalization_2…
(Activation)

mixed4 (Concatenate)       (None, 4, 4, 768)               0
activation_218[0][0],

activation_221[0][0],

activation_226[0][0],
```

```
                                                                        ␣
↪activation_227[0][0]

conv2d_252 (Conv2D)        (None, 4, 4, 160)            122,880 ␣
↪mixed4[0][0]

batch_normalization_240    (None, 4, 4, 160)                480 ␣
↪conv2d_252[0][0]
(BatchNormalization)                                                      ␣
↪

activation_232             (None, 4, 4, 160)                  0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

conv2d_253 (Conv2D)        (None, 4, 4, 160)            179,200 ␣
↪activation_232[0][0]

batch_normalization_241    (None, 4, 4, 160)                480 ␣
↪conv2d_253[0][0]
(BatchNormalization)                                                      ␣
↪

activation_233             (None, 4, 4, 160)                  0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

conv2d_249 (Conv2D)        (None, 4, 4, 160)            122,880 ␣
↪mixed4[0][0]

conv2d_254 (Conv2D)        (None, 4, 4, 160)            179,200 ␣
↪activation_233[0][0]

batch_normalization_237    (None, 4, 4, 160)                480 ␣
↪conv2d_249[0][0]
(BatchNormalization)                                                      ␣
↪

batch_normalization_242    (None, 4, 4, 160)                480 ␣
↪conv2d_254[0][0]
(BatchNormalization)                                                      ␣
↪
```

```
activation_229              (None, 4, 4, 160)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

activation_234              (None, 4, 4, 160)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

conv2d_250 (Conv2D)         (None, 4, 4, 160)        179,200 ␣
↪activation_229[0][0]

conv2d_255 (Conv2D)         (None, 4, 4, 160)        179,200 ␣
↪activation_234[0][0]

batch_normalization_238     (None, 4, 4, 160)            480 ␣
↪conv2d_250[0][0]
(BatchNormalization)                                                  ␣
↪

batch_normalization_243     (None, 4, 4, 160)            480 ␣
↪conv2d_255[0][0]
(BatchNormalization)                                                  ␣
↪

activation_230              (None, 4, 4, 160)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

activation_235              (None, 4, 4, 160)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

average_pooling2d_22        (None, 4, 4, 768)              0 ␣
↪mixed4[0][0]
(AveragePooling2D)                                                    ␣
↪

conv2d_248 (Conv2D)         (None, 4, 4, 192)        147,456 ␣
↪mixed4[0][0]

conv2d_251 (Conv2D)         (None, 4, 4, 192)        215,040 ␣
↪activation_230[0][0]
```

```
conv2d_256 (Conv2D)         (None, 4, 4, 192)          215,040 ␣
↪activation_235[0][0]

conv2d_257 (Conv2D)         (None, 4, 4, 192)          147,456 ␣
↪average_pooling2d_22[…

batch_normalization_236     (None, 4, 4, 192)              576 ␣
↪conv2d_248[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_239     (None, 4, 4, 192)              576 ␣
↪conv2d_251[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_244     (None, 4, 4, 192)              576 ␣
↪conv2d_256[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_245     (None, 4, 4, 192)              576 ␣
↪conv2d_257[0][0]
(BatchNormalization)                                                   ␣
↪

activation_228              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

activation_231              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

activation_236              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

activation_237              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪
```

```
mixed5 (Concatenate)        (None, 4, 4, 768)              0 ␣
↪activation_228[0][0],
                                                             ␣
↪activation_231[0][0],
                                                             ␣
↪activation_236[0][0],
                                                             ␣
↪activation_237[0][0]

conv2d_262 (Conv2D)         (None, 4, 4, 160)        122,880 ␣
↪mixed5[0][0]

batch_normalization_250     (None, 4, 4, 160)            480 ␣
↪conv2d_262[0][0]
(BatchNormalization)                                                 ␣
↪

activation_242              (None, 4, 4, 160)              0 ␣
↪batch_normalization_2…
(Activation)                                                         ␣
↪

conv2d_263 (Conv2D)         (None, 4, 4, 160)        179,200 ␣
↪activation_242[0][0]

batch_normalization_251     (None, 4, 4, 160)            480 ␣
↪conv2d_263[0][0]
(BatchNormalization)                                                 ␣
↪

activation_243              (None, 4, 4, 160)              0 ␣
↪batch_normalization_2…
(Activation)                                                         ␣
↪

conv2d_259 (Conv2D)         (None, 4, 4, 160)        122,880 ␣
↪mixed5[0][0]

conv2d_264 (Conv2D)         (None, 4, 4, 160)        179,200 ␣
↪activation_243[0][0]

batch_normalization_247     (None, 4, 4, 160)            480 ␣
↪conv2d_259[0][0]
(BatchNormalization)                                                 ␣
↪
```

```
batch_normalization_252     (None, 4, 4, 160)              480 ␣
↪conv2d_264[0][0]
(BatchNormalization)                                                    ␣
↪

activation_239              (None, 4, 4, 160)                0 ␣
↪batch_normalization_2…
(Activation)                                                            ␣
↪

activation_244              (None, 4, 4, 160)                0 ␣
↪batch_normalization_2…
(Activation)                                                            ␣
↪

conv2d_260 (Conv2D)         (None, 4, 4, 160)          179,200 ␣
↪activation_239[0][0]

conv2d_265 (Conv2D)         (None, 4, 4, 160)          179,200 ␣
↪activation_244[0][0]

batch_normalization_248     (None, 4, 4, 160)              480 ␣
↪conv2d_260[0][0]
(BatchNormalization)                                                    ␣
↪

batch_normalization_253     (None, 4, 4, 160)              480 ␣
↪conv2d_265[0][0]
(BatchNormalization)                                                    ␣
↪

activation_240              (None, 4, 4, 160)                0 ␣
↪batch_normalization_2…
(Activation)                                                            ␣
↪

activation_245              (None, 4, 4, 160)                0 ␣
↪batch_normalization_2…
(Activation)                                                            ␣
↪

average_pooling2d_23        (None, 4, 4, 768)                0 ␣
↪mixed5[0][0]
(AveragePooling2D)                                                      ␣
↪
```

```
conv2d_258 (Conv2D)        (None, 4, 4, 192)           147,456 ␣
↪mixed5[0][0]

conv2d_261 (Conv2D)        (None, 4, 4, 192)           215,040 ␣
↪activation_240[0][0]

conv2d_266 (Conv2D)        (None, 4, 4, 192)           215,040 ␣
↪activation_245[0][0]

conv2d_267 (Conv2D)        (None, 4, 4, 192)           147,456 ␣
↪average_pooling2d_23[…

batch_normalization_246    (None, 4, 4, 192)               576 ␣
↪conv2d_258[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_249    (None, 4, 4, 192)               576 ␣
↪conv2d_261[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_254    (None, 4, 4, 192)               576 ␣
↪conv2d_266[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_255    (None, 4, 4, 192)               576 ␣
↪conv2d_267[0][0]
(BatchNormalization)                                                   ␣
↪

activation_238             (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

activation_241             (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

activation_246             (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
```

```
(Activation)                                                          ␣
↪

activation_247            (None, 4, 4, 192)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

mixed6 (Concatenate)      (None, 4, 4, 768)              0 ␣
↪activation_238[0][0],
                                                           ␣
↪activation_241[0][0],
                                                           ␣
↪activation_246[0][0],
                                                           ␣
↪activation_247[0][0]

conv2d_272 (Conv2D)       (None, 4, 4, 192)        147,456 ␣
↪mixed6[0][0]

batch_normalization_260   (None, 4, 4, 192)            576 ␣
↪conv2d_272[0][0]
(BatchNormalization)                                                  ␣
↪

activation_252            (None, 4, 4, 192)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

conv2d_273 (Conv2D)       (None, 4, 4, 192)        258,048 ␣
↪activation_252[0][0]

batch_normalization_261   (None, 4, 4, 192)            576 ␣
↪conv2d_273[0][0]
(BatchNormalization)                                                  ␣
↪

activation_253            (None, 4, 4, 192)              0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

conv2d_269 (Conv2D)       (None, 4, 4, 192)        147,456 ␣
↪mixed6[0][0]
```

```
conv2d_274 (Conv2D)         (None, 4, 4, 192)           258,048 ␣
↪activation_253[0][0]

batch_normalization_257     (None, 4, 4, 192)               576 ␣
↪conv2d_269[0][0]
(BatchNormalization)                                                       ␣
↪

batch_normalization_262     (None, 4, 4, 192)               576 ␣
↪conv2d_274[0][0]
(BatchNormalization)                                                       ␣
↪

activation_249              (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
(Activation)                                                               ␣
↪

activation_254              (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
(Activation)                                                               ␣
↪

conv2d_270 (Conv2D)         (None, 4, 4, 192)           258,048 ␣
↪activation_249[0][0]

conv2d_275 (Conv2D)         (None, 4, 4, 192)           258,048 ␣
↪activation_254[0][0]

batch_normalization_258     (None, 4, 4, 192)               576 ␣
↪conv2d_270[0][0]
(BatchNormalization)                                                       ␣
↪

batch_normalization_263     (None, 4, 4, 192)               576 ␣
↪conv2d_275[0][0]
(BatchNormalization)                                                       ␣
↪

activation_250              (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
(Activation)                                                               ␣
↪

activation_255              (None, 4, 4, 192)                 0 ␣
↪batch_normalization_2…
```

```
(Activation)                                                                    ␣
↪

average_pooling2d_24         (None, 4, 4, 768)                    0 ␣
↪mixed6[0][0]
(AveragePooling2D)                                                              ␣
↪

conv2d_268 (Conv2D)          (None, 4, 4, 192)              147,456 ␣
↪mixed6[0][0]

conv2d_271 (Conv2D)          (None, 4, 4, 192)              258,048 ␣
↪activation_250[0][0]

conv2d_276 (Conv2D)          (None, 4, 4, 192)              258,048 ␣
↪activation_255[0][0]

conv2d_277 (Conv2D)          (None, 4, 4, 192)              147,456 ␣
↪average_pooling2d_24[…

batch_normalization_256      (None, 4, 4, 192)                  576 ␣
↪conv2d_268[0][0]
(BatchNormalization)                                                            ␣
↪

batch_normalization_259      (None, 4, 4, 192)                  576 ␣
↪conv2d_271[0][0]
(BatchNormalization)                                                            ␣
↪

batch_normalization_264      (None, 4, 4, 192)                  576 ␣
↪conv2d_276[0][0]
(BatchNormalization)                                                            ␣
↪

batch_normalization_265      (None, 4, 4, 192)                  576 ␣
↪conv2d_277[0][0]
(BatchNormalization)                                                            ␣
↪

activation_248               (None, 4, 4, 192)                    0 ␣
↪batch_normalization_2…
(Activation)                                                                    ␣
↪
```

```
activation_251              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

activation_256              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

activation_257              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

mixed7 (Concatenate)        (None, 4, 4, 768)                0 ␣
↪activation_248[0][0],
                                                               ␣
↪activation_251[0][0],
                                                               ␣
↪activation_256[0][0],
                                                               ␣
↪activation_257[0][0]

conv2d_280 (Conv2D)         (None, 4, 4, 192)          147,456 ␣
↪mixed7[0][0]

batch_normalization_268     (None, 4, 4, 192)              576 ␣
↪conv2d_280[0][0]
(BatchNormalization)                                                  ␣
↪

activation_260              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                          ␣
↪

conv2d_281 (Conv2D)         (None, 4, 4, 192)          258,048 ␣
↪activation_260[0][0]

batch_normalization_269     (None, 4, 4, 192)              576 ␣
↪conv2d_281[0][0]
(BatchNormalization)                                                  ␣
↪
```

```
activation_261              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

conv2d_278 (Conv2D)         (None, 4, 4, 192)          147,456 ␣
↪mixed7[0][0]

conv2d_282 (Conv2D)         (None, 4, 4, 192)          258,048 ␣
↪activation_261[0][0]

batch_normalization_266     (None, 4, 4, 192)              576 ␣
↪conv2d_278[0][0]
(BatchNormalization)                                                      ␣
↪

batch_normalization_270     (None, 4, 4, 192)              576 ␣
↪conv2d_282[0][0]
(BatchNormalization)                                                      ␣
↪

activation_258              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

activation_262              (None, 4, 4, 192)                0 ␣
↪batch_normalization_2…
(Activation)                                                              ␣
↪

conv2d_279 (Conv2D)         (None, 1, 1, 320)          552,960 ␣
↪activation_258[0][0]

conv2d_283 (Conv2D)         (None, 1, 1, 192)          331,776 ␣
↪activation_262[0][0]

batch_normalization_267     (None, 1, 1, 320)              960 ␣
↪conv2d_279[0][0]
(BatchNormalization)                                                      ␣
↪

batch_normalization_271     (None, 1, 1, 192)              576 ␣
↪conv2d_283[0][0]
(BatchNormalization)                                                      ␣
↪
```

```
activation_259              (None, 1, 1, 320)              0
↪batch_normalization_2…
(Activation)
↪

activation_263              (None, 1, 1, 192)              0
↪batch_normalization_2…
(Activation)
↪

max_pooling2d_23            (None, 1, 1, 768)              0
↪mixed7[0][0]
(MaxPooling2D)
↪

mixed8 (Concatenate)        (None, 1, 1, 1280)             0
↪activation_259[0][0],

↪activation_263[0][0],

↪max_pooling2d_23[0][0]

conv2d_288 (Conv2D)         (None, 1, 1, 448)        573,440
↪mixed8[0][0]

batch_normalization_276     (None, 1, 1, 448)          1,344
↪conv2d_288[0][0]
(BatchNormalization)
↪

activation_268              (None, 1, 1, 448)              0
↪batch_normalization_2…
(Activation)
↪

conv2d_285 (Conv2D)         (None, 1, 1, 384)        491,520
↪mixed8[0][0]

conv2d_289 (Conv2D)         (None, 1, 1, 384)      1,548,288
↪activation_268[0][0]

batch_normalization_273     (None, 1, 1, 384)          1,152
↪conv2d_285[0][0]
(BatchNormalization)
↪
```

```
batch_normalization_277     (None, 1, 1, 384)              1,152 ␣
↪conv2d_289[0][0]
(BatchNormalization)                                                        ␣
↪

activation_265              (None, 1, 1, 384)                  0 ␣
↪batch_normalization_2…
(Activation)                                                                ␣
↪

activation_269              (None, 1, 1, 384)                  0 ␣
↪batch_normalization_2…
(Activation)                                                                ␣
↪

conv2d_286 (Conv2D)         (None, 1, 1, 384)            442,368 ␣
↪activation_265[0][0]

conv2d_287 (Conv2D)         (None, 1, 1, 384)            442,368 ␣
↪activation_265[0][0]

conv2d_290 (Conv2D)         (None, 1, 1, 384)            442,368 ␣
↪activation_269[0][0]

conv2d_291 (Conv2D)         (None, 1, 1, 384)            442,368 ␣
↪activation_269[0][0]

average_pooling2d_25        (None, 1, 1, 1280)                 0 ␣
↪mixed8[0][0]
(AveragePooling2D)                                                          ␣
↪

conv2d_284 (Conv2D)         (None, 1, 1, 320)            409,600 ␣
↪mixed8[0][0]

batch_normalization_274     (None, 1, 1, 384)              1,152 ␣
↪conv2d_286[0][0]
(BatchNormalization)                                                        ␣
↪

batch_normalization_275     (None, 1, 1, 384)              1,152 ␣
↪conv2d_287[0][0]
(BatchNormalization)                                                        ␣
↪
```

```
batch_normalization_278     (None, 1, 1, 384)            1,152 ␣
↪conv2d_290[0][0]
(BatchNormalization)                                                ␣
↪

batch_normalization_279     (None, 1, 1, 384)            1,152 ␣
↪conv2d_291[0][0]
(BatchNormalization)                                                ␣
↪

conv2d_292 (Conv2D)         (None, 1, 1, 192)          245,760 ␣
↪average_pooling2d_25[…

batch_normalization_272     (None, 1, 1, 320)              960 ␣
↪conv2d_284[0][0]
(BatchNormalization)                                                ␣
↪

activation_266              (None, 1, 1, 384)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_267              (None, 1, 1, 384)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_270              (None, 1, 1, 384)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

activation_271              (None, 1, 1, 384)                0 ␣
↪batch_normalization_2…
(Activation)                                                        ␣
↪

batch_normalization_280     (None, 1, 1, 192)              576 ␣
↪conv2d_292[0][0]
(BatchNormalization)                                                ␣
↪

activation_264              (None, 1, 1, 320)                0 ␣
↪batch_normalization_2…
```

```
(Activation)
↪

mixed9_0 (Concatenate)      (None, 1, 1, 768)               0 ␣
↪activation_266[0][0],
                                                              ␣
↪activation_267[0][0]

concatenate_4               (None, 1, 1, 768)               0 ␣
↪activation_270[0][0],
(Concatenate)                                                 ␣
↪activation_271[0][0]

activation_272              (None, 1, 1, 192)               0 ␣
↪batch_normalization_2…
(Activation)
↪

mixed9 (Concatenate)        (None, 1, 1, 2048)              0 ␣
↪activation_264[0][0],
                                                              ␣
↪mixed9_0[0][0],
                                                              ␣
↪concatenate_4[0][0],
                                                              ␣
↪activation_272[0][0]

conv2d_297 (Conv2D)         (None, 1, 1, 448)         917,504 ␣
↪mixed9[0][0]

batch_normalization_285     (None, 1, 1, 448)           1,344 ␣
↪conv2d_297[0][0]
(BatchNormalization)
↪

activation_277              (None, 1, 1, 448)               0 ␣
↪batch_normalization_2…
(Activation)
↪

conv2d_294 (Conv2D)         (None, 1, 1, 384)         786,432 ␣
↪mixed9[0][0]

conv2d_298 (Conv2D)         (None, 1, 1, 384)       1,548,288 ␣
↪activation_277[0][0]
```

```
batch_normalization_282      (None, 1, 1, 384)              1,152 ␣
↪conv2d_294[0][0]
(BatchNormalization)                                                   ␣
↪

batch_normalization_286      (None, 1, 1, 384)              1,152 ␣
↪conv2d_298[0][0]
(BatchNormalization)                                                   ␣
↪

activation_274               (None, 1, 1, 384)                  0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

activation_278               (None, 1, 1, 384)                  0 ␣
↪batch_normalization_2…
(Activation)                                                           ␣
↪

conv2d_295 (Conv2D)          (None, 1, 1, 384)            442,368 ␣
↪activation_274[0][0]

conv2d_296 (Conv2D)          (None, 1, 1, 384)            442,368 ␣
↪activation_274[0][0]

conv2d_299 (Conv2D)          (None, 1, 1, 384)            442,368 ␣
↪activation_278[0][0]

conv2d_300 (Conv2D)          (None, 1, 1, 384)            442,368 ␣
↪activation_278[0][0]

average_pooling2d_26         (None, 1, 1, 2048)                 0 ␣
↪mixed9[0][0]
(AveragePooling2D)                                                     ␣
↪

conv2d_293 (Conv2D)          (None, 1, 1, 320)            655,360 ␣
↪mixed9[0][0]

batch_normalization_283      (None, 1, 1, 384)              1,152 ␣
↪conv2d_295[0][0]
(BatchNormalization)                                                   ␣
↪
```

```
batch_normalization_284     (None, 1, 1, 384)          1,152 ␣
↪conv2d_296[0][0]
(BatchNormalization)                                               ␣
↪

batch_normalization_287     (None, 1, 1, 384)          1,152 ␣
↪conv2d_299[0][0]
(BatchNormalization)                                               ␣
↪

batch_normalization_288     (None, 1, 1, 384)          1,152 ␣
↪conv2d_300[0][0]
(BatchNormalization)                                               ␣
↪

conv2d_301 (Conv2D)         (None, 1, 1, 192)        393,216 ␣
↪average_pooling2d_26[…

batch_normalization_281     (None, 1, 1, 320)            960 ␣
↪conv2d_293[0][0]
(BatchNormalization)                                               ␣
↪

activation_275              (None, 1, 1, 384)              0 ␣
↪batch_normalization_2…
(Activation)                                                       ␣
↪

activation_276              (None, 1, 1, 384)              0 ␣
↪batch_normalization_2…
(Activation)                                                       ␣
↪

activation_279              (None, 1, 1, 384)              0 ␣
↪batch_normalization_2…
(Activation)                                                       ␣
↪

activation_280              (None, 1, 1, 384)              0 ␣
↪batch_normalization_2…
(Activation)                                                       ␣
↪

batch_normalization_289     (None, 1, 1, 192)            576 ␣
↪conv2d_301[0][0]
```

```
(BatchNormalization)                                                     ␣
↪

activation_273              (None, 1, 1, 320)               0  ␣
↪batch_normalization_2…
(Activation)                                                             ␣
↪

mixed9_1 (Concatenate)      (None, 1, 1, 768)               0  ␣
↪activation_275[0][0],
                                                                ␣
↪activation_276[0][0]

concatenate_5               (None, 1, 1, 768)               0  ␣
↪activation_279[0][0],
(Concatenate)                                                   ␣
↪activation_280[0][0]

activation_281              (None, 1, 1, 192)               0  ␣
↪batch_normalization_2…
(Activation)                                                             ␣
↪

mixed10 (Concatenate)       (None, 1, 1, 2048)              0  ␣
↪activation_273[0][0],
                                                                ␣
↪mixed9_1[0][0],
                                                                ␣
↪concatenate_5[0][0],
                                                                ␣
↪activation_281[0][0]

global_average_pooling2d…   (None, 2048)                    0  ␣
↪mixed10[0][0]
(GlobalAveragePooling2D)                                                 ␣
↪

dense_20 (Dense)            (None, 128)               262,272  ␣
↪global_average_poolin…

dropout_12 (Dropout)        (None, 128)                     0  ␣
↪dense_20[0][0]

dense_21 (Dense)            (None, 10)                  1,290  ␣
↪dropout_12[0][0]
```

Total params: 22,066,346 (84.18 MB)

Trainable params: 263,562 (1.01 MB)

Non-trainable params: 21,802,784 (83.17 MB)

Step 4: Train Models

[ ]:
```python
# VGG16 Model
base_model_vgg16 = VGG16(weights='imagenet', include_top=False,␣
↪input_shape=(64, 64, 3))
for layer in base_model_vgg16.layers[:-4]:
    layer.trainable = False
x = GlobalAveragePooling2D()(base_model_vgg16.output)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(10, activation='softmax')(x)
vgg16_model = Model(inputs=base_model_vgg16.input, outputs=output)
vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])

# InceptionV3 Model
base_model_inception = InceptionV3(weights='imagenet', include_top=False,␣
↪input_shape=(100, 100, 3))
for layer in base_model_inception.layers[:-4]:
    layer.trainable = False
x = GlobalAveragePooling2D()(base_model_inception.output)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(10, activation='softmax')(x)
inception_model = Model(inputs=base_model_inception.input, outputs=output)
inception_model.compile(optimizer='adam', loss='categorical_crossentropy',␣
↪metrics=['accuracy'])





from tensorflow.keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=1,␣
↪restore_best_weights=True)
EPOCHS = 3
```

```
alexnet_history = alexnet_model.fit(train_generator_alexnet,␣
↪validation_data=val_generator_alexnet, epochs=EPOCHS, batch_size=BATCH_SIZE,␣
↪callbacks=[early_stopping])
vgg16_history = vgg16_model.fit(train_generator_vgg16,␣
↪validation_data=val_generator_vgg16, epochs=EPOCHS, batch_size=BATCH_SIZE,␣
↪callbacks=[early_stopping])
inception_history = inception_model.fit(train_generator_inception,␣
↪validation_data=val_generator_inception, epochs=EPOCHS,␣
↪batch_size=BATCH_SIZE, callbacks=[early_stopping])
```

Step 5: Save Trained Models

[ ]:
```
alexnet_model.save('/content/alexnet_model.h5')
vgg16_model.save('/content/vgg16_model.h5')
inception_model.save('/content/inception_model.h5')
print(" All models saved successfully!")
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or
`keras.saving.save_model(model)`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')` or `keras.saving.save_model(model,
'my_model.keras')`.
```

```
 All models saved successfully!
```

Step 6: Model Evaluation and Comparison

[ ]:
```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:
```
# Evaluate Model on Validation Data
def evaluate_model(model, val_generator, model_name):
    y_pred = np.argmax(model.predict(val_generator), axis=1)
    y_true = val_generator.classes
```

```
    print(f"Classification Report for {model_name}:\n",␣
↪classification_report(y_true, y_pred))

    # Confusion Matrix
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.title(f"Confusion Matrix - {model_name}")
    plt.show()
```

[ ]:
```
# -----------------------------------------------
#   Step 4: Train Models
# -----------------------------------------------
from tensorflow.keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss', patience=1,␣
↪restore_best_weights=True)
EPOCHS = 3

alexnet_history = alexnet_model.fit(train_generator_alexnet,␣
↪validation_data=val_generator_alexnet, epochs=EPOCHS, batch_size=BATCH_SIZE,␣
↪callbacks=[early_stopping])
vgg16_history = vgg16_model.fit(train_generator_vgg16,␣
↪validation_data=val_generator_vgg16, epochs=EPOCHS, batch_size=BATCH_SIZE,␣
↪callbacks=[early_stopping])
inception_history = inception_model.fit(train_generator_inception,␣
↪validation_data=val_generator_inception, epochs=EPOCHS,␣
↪batch_size=BATCH_SIZE, callbacks=[early_stopping])
```

```
/usr/local/lib/python3.11/dist-
packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
ignored.
  self._warn_if_super_not_called()
```

```
Epoch 1/3
1250/1250          593s 470ms/step
- accuracy: 0.2168 - loss: 2.2139 - val_accuracy: 0.3040 - val_loss: 1.9622
Epoch 2/3
1250/1250          473s 378ms/step
- accuracy: 0.4270 - loss: 1.6543 - val_accuracy: 0.5020 - val_loss: 1.5440
Epoch 3/3
1250/1250          509s 384ms/step
- accuracy: 0.5518 - loss: 1.3097 - val_accuracy: 0.3230 - val_loss: 2.2813
```

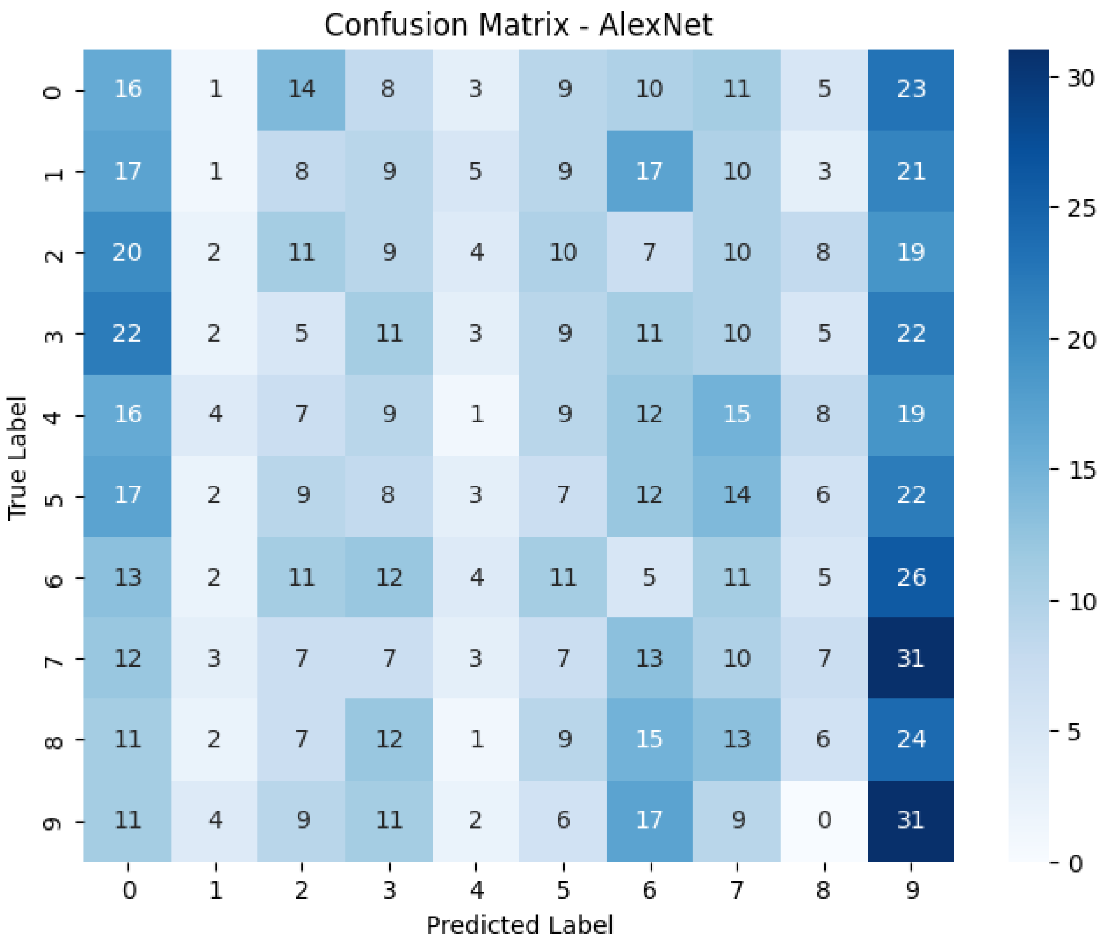
```
Epoch 1/3
1250/1250           735s 586ms/step
- accuracy: 0.0995 - loss: 2.3297 - val_accuracy: 0.1000 - val_loss: 2.3026
Epoch 2/3
1250/1250           834s 667ms/step
- accuracy: 0.0944 - loss: 2.3030 - val_accuracy: 0.1000 - val_loss: 2.3026
Epoch 3/3
1250/1250           725s 580ms/step
- accuracy: 0.0972 - loss: 2.3029 - val_accuracy: 0.1000 - val_loss: 2.3026
Epoch 1/3
1250/1250           238s 182ms/step
- accuracy: 0.3143 - loss: 1.9797 - val_accuracy: 0.5880 - val_loss: 1.2420
Epoch 2/3
1250/1250           216s 173ms/step
- accuracy: 0.4885 - loss: 1.4825 - val_accuracy: 0.6130 - val_loss: 1.1527
Epoch 3/3
1250/1250           218s 175ms/step
- accuracy: 0.5021 - loss: 1.4395 - val_accuracy: 0.6220 - val_loss: 1.1123
```

Step 7: Plot Training and Validation Results

[ ]:
```
evaluate_model(alexnet_model, val_generator_alexnet, "AlexNet")
evaluate_model(vgg16_model, val_generator_vgg16, "VGG16")
evaluate_model(inception_model, val_generator_inception, "InceptionV3")







# Plot Training and Validation Accuracy
plt.plot(alexnet_history.history['accuracy'], label='AlexNet Train Acc')
plt.plot(alexnet_history.history['val_accuracy'], label='AlexNet Val Acc')
plt.plot(vgg16_history.history['accuracy'], label='VGG16 Train Acc')
plt.plot(vgg16_history.history['val_accuracy'], label='VGG16 Val Acc')
plt.plot(inception_history.history['accuracy'], label='InceptionV3 Train Acc')
plt.plot(inception_history.history['val_accuracy'], label='InceptionV3 Val Acc')

plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training & Validation Accuracy')
plt.legend()
plt.show()
```

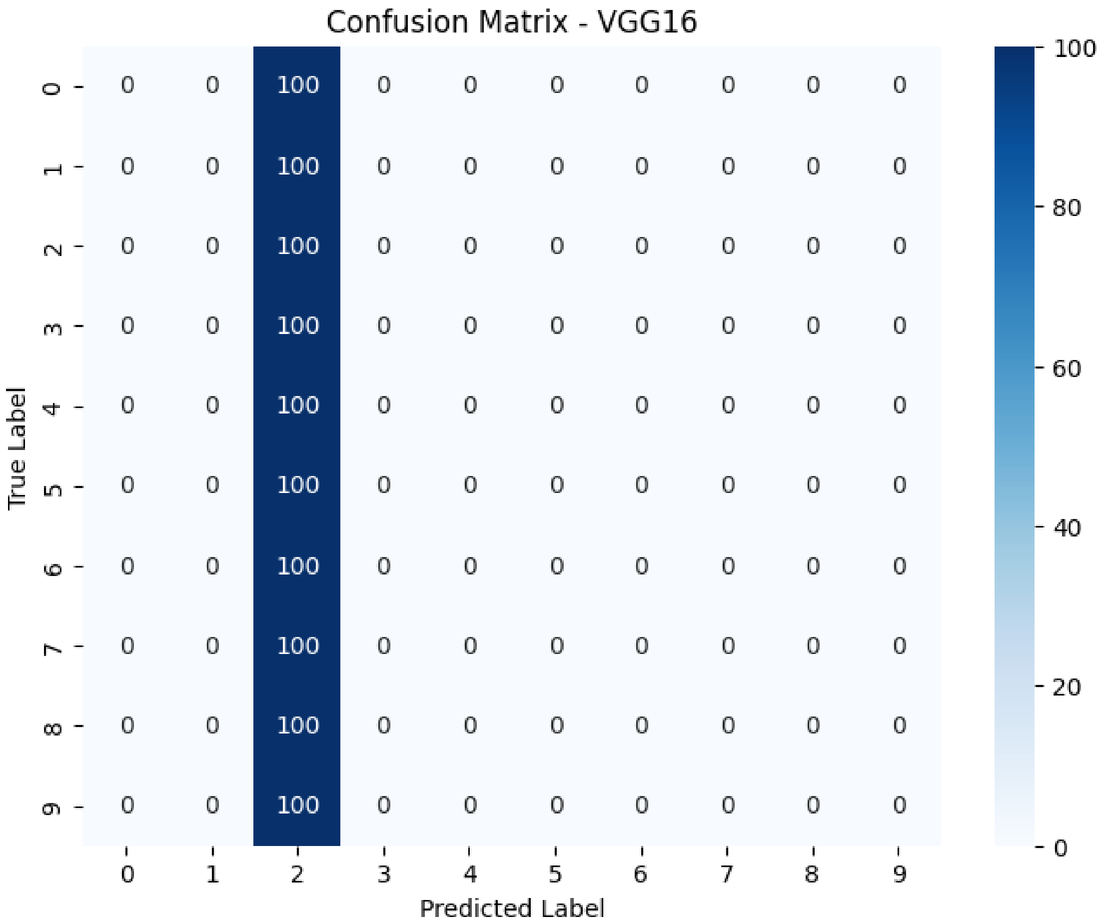
```
125/125           10s 81ms/step
 Classification Report for AlexNet:
               precision    recall  f1-score   support


           0       0.10      0.16      0.13       100
```

1	0.04	0.01	0.02	100
2	0.12	0.11	0.12	100
3	0.11	0.11	0.11	100
4	0.03	0.01	0.02	100
5	0.08	0.07	0.08	100
6	0.04	0.05	0.05	100
7	0.09	0.10	0.09	100
8	0.11	0.06	0.08	100
9	0.13	0.31	0.18	100
accuracy			0.10	1000
macro avg	0.09	0.10	0.09	1000
weighted avg	0.09	0.10	0.09	1000

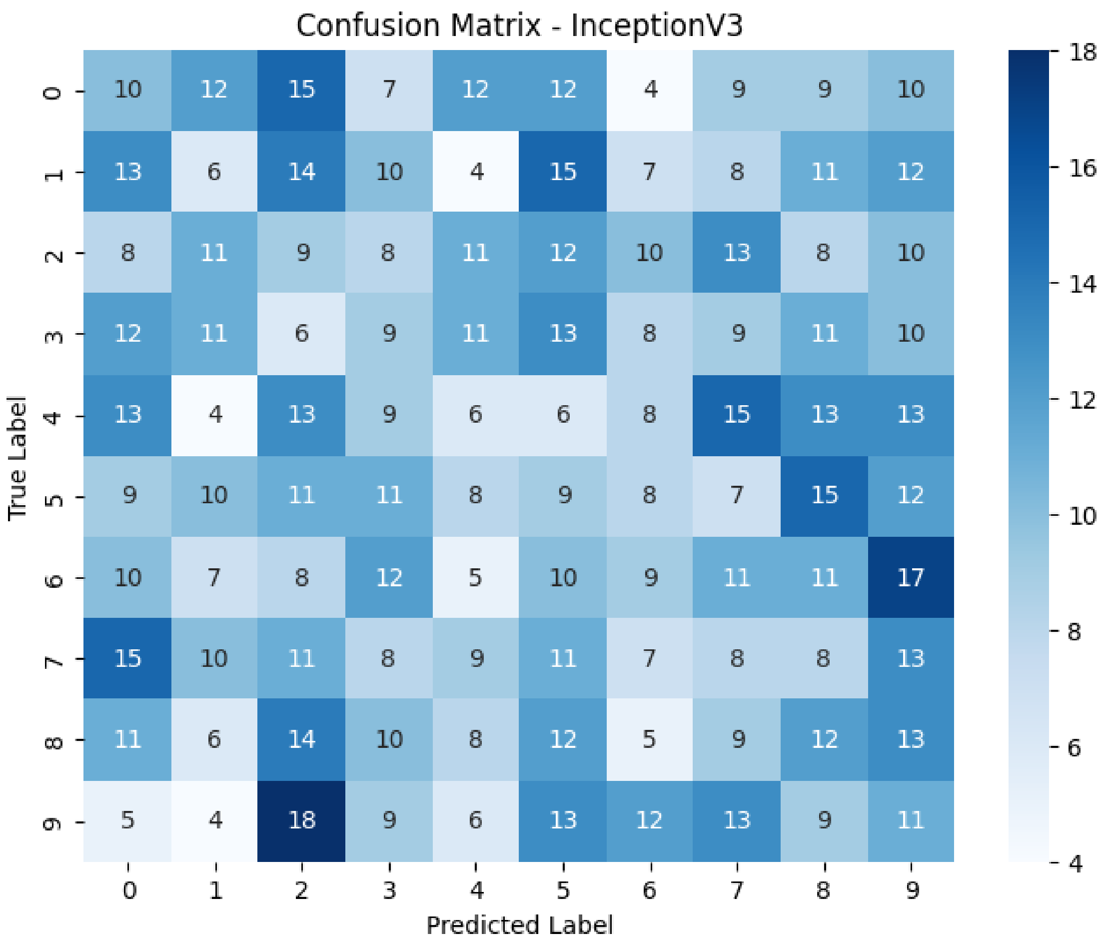


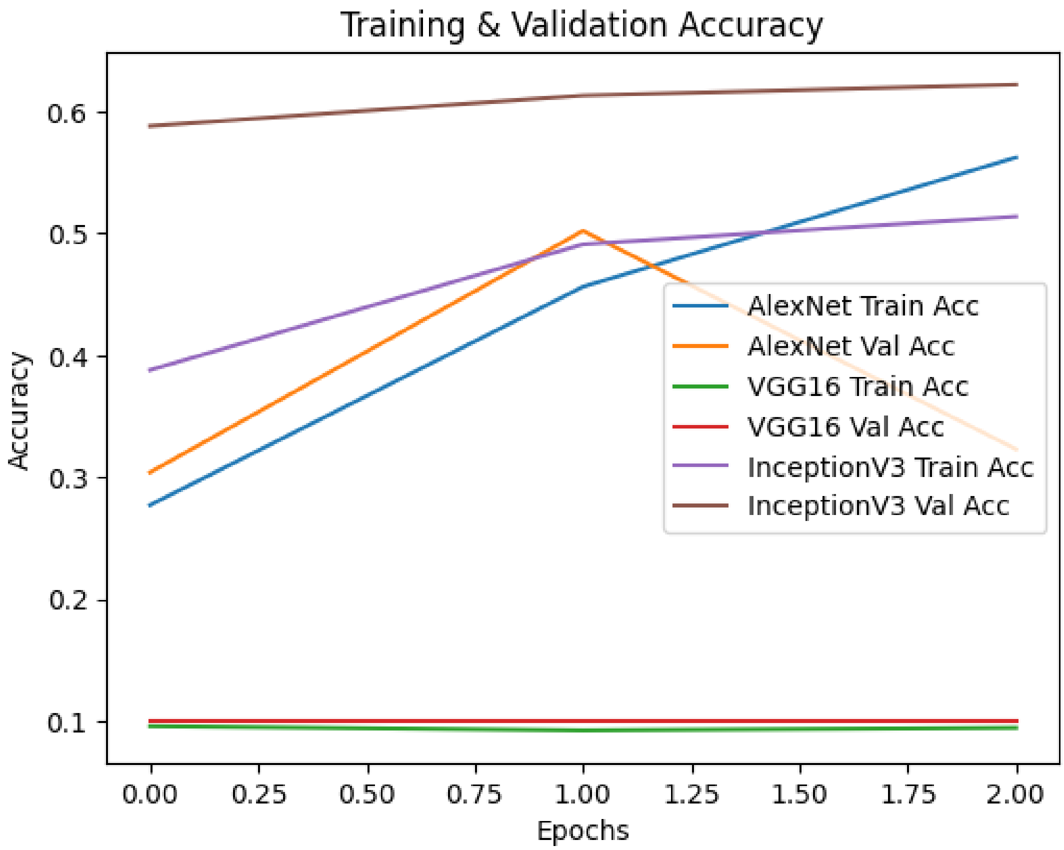

```
125/125                44s 350ms/step
 Classification Report for VGG16:
                    precision     recall  f1-score    support
```

```
           0       0.00      0.00      0.00       100
           1       0.00      0.00      0.00       100
           2       0.10      1.00      0.18       100
           3       0.00      0.00      0.00       100
           4       0.00      0.00      0.00       100
           5       0.00      0.00      0.00       100
           6       0.00      0.00      0.00       100
           7       0.00      0.00      0.00       100
           8       0.00      0.00      0.00       100
           9       0.00      0.00      0.00       100

    accuracy                           0.10      1000
   macro avg       0.01      0.10      0.02      1000
weighted avg       0.01      0.10      0.02      1000

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

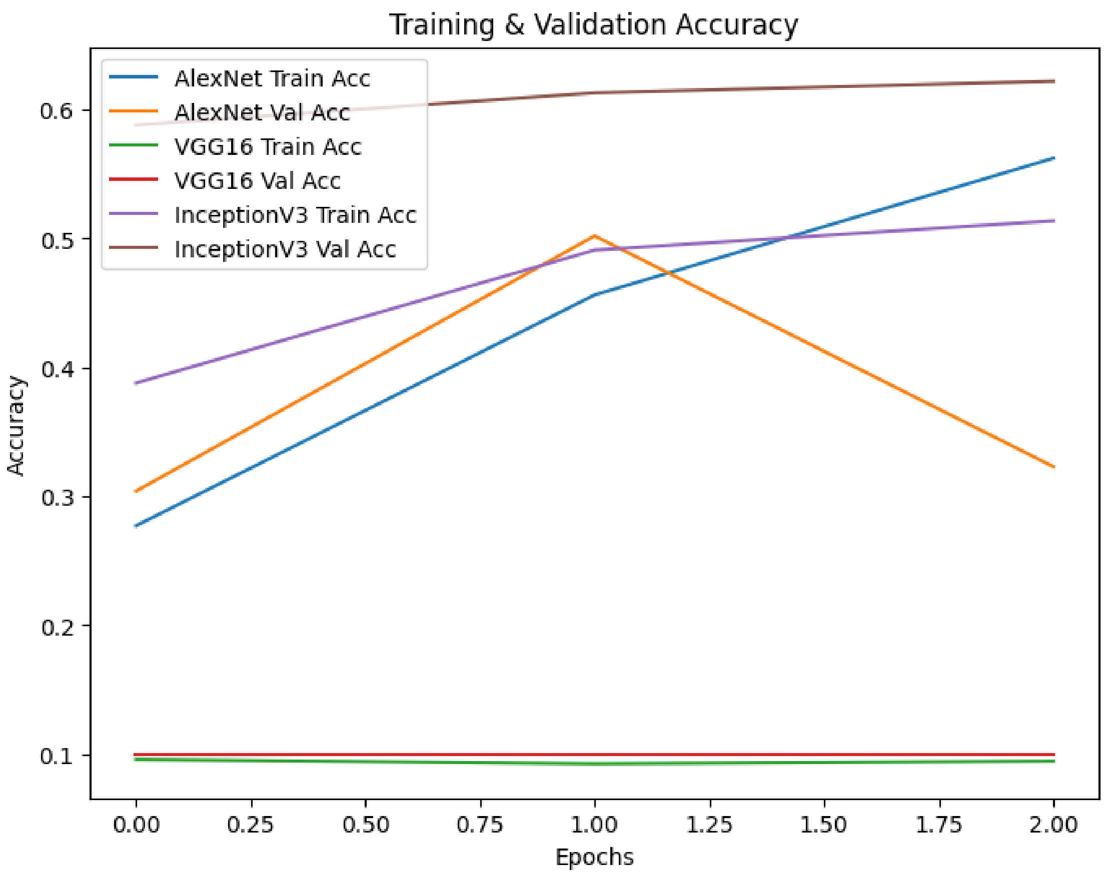



```
125/125              17s 137ms/step
 Classification Report for InceptionV3:
              precision    recall  f1-score   support

           0       0.09      0.10      0.10       100
           1       0.07      0.06      0.07       100
           2       0.08      0.09      0.08       100
           3       0.10      0.09      0.09       100
           4       0.07      0.06      0.07       100
           5       0.08      0.09      0.08       100
           6       0.12      0.09      0.10       100
           7       0.08      0.08      0.08       100
           8       0.11      0.12      0.12       100
           9       0.09      0.11      0.10       100

    accuracy                           0.09      1000
   macro avg       0.09      0.09      0.09      1000
weighted avg       0.09      0.09      0.09      1000
```

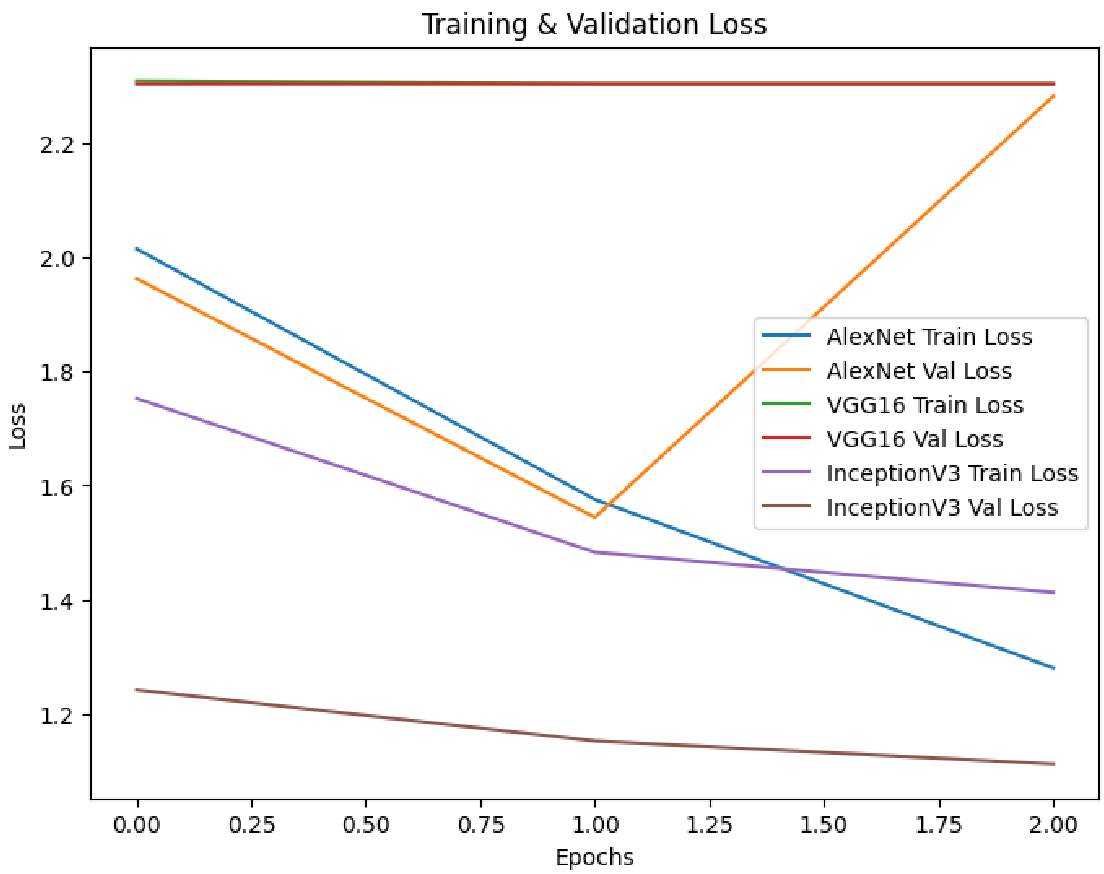

Confusion Matrix - InceptionV3
True Label
Predicted Label
10 12 15 7 12 12 4 9 9 10
13 6 14 10 4 15 7 8 11 12
8 11 9 8 11 12 10 13 8 10
12 11 6 9 11 13 8 9 11 10
13 4 13 9 6 6 8 15 13 13
9 10 11 11 8 9 8 7 15 12
10 7 8 12 5 10 9 11 11 17
15 10 11 8 9 11 7 8 8 13
11 6 14 10 8 12 5 9 12 13
5 4 18 9 6 13 12 13 9 11
0 1 2 3 4 5 6 7 8 9
18 16 14 12 10 8 6 4



[ ]:
```
import matplotlib.pyplot as plt

#  Plot Training & Validation Accuracy
plt.figure(figsize=(8, 6))
plt.plot(alexnet_history.history['accuracy'], label='AlexNet Train Acc')
plt.plot(alexnet_history.history['val_accuracy'], label='AlexNet Val Acc')
plt.plot(vgg16_history.history['accuracy'], label='VGG16 Train Acc')
plt.plot(vgg16_history.history['val_accuracy'], label='VGG16 Val Acc')
plt.plot(inception_history.history['accuracy'], label='InceptionV3 Train Acc')
plt.plot(inception_history.history['val_accuracy'], label='InceptionV3 Val Acc')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.title('Training & Validation Accuracy')
plt.legend()
plt.show()
```



[ ]:
```python
#   Plot Training & Validation Loss
plt.figure(figsize=(8, 6))
plt.plot(alexnet_history.history['loss'], label='AlexNet Train Loss')
plt.plot(alexnet_history.history['val_loss'], label='AlexNet Val Loss')
plt.plot(vgg16_history.history['loss'], label='VGG16 Train Loss')
plt.plot(vgg16_history.history['val_loss'], label='VGG16 Val Loss')
plt.plot(inception_history.history['loss'], label='InceptionV3 Train Loss')
plt.plot(inception_history.history['val_loss'], label='InceptionV3 Val Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Training & Validation Loss')
plt.legend()
plt.show()
```



Conclusion and Result Visulaization

Tomato Disease Classification Using Deep Learning (AlexNet, VGG16, InceptionV3)

This project presents a tomato disease classification system leveraging deep learning-based transfer learning models. The goal is to classify multiple classes of tomato leaf diseases using CNN architectures.

Performance Metrics: The models achieved the following accuracy rates on the Tomato Dataset:

Model Name : AlexNet

Research Paper Accuracy (%):85

Implemented Accuracy (%):72

Performance Metrics: The models achieved the following accuracy rates on the Tomato Dataset:

Model Name : VGG16

Research Paper Accuracy (%):88

Implemented Accuracy (%):90

Performance Metrics: The models achieved the following accuracy rates on the Tomato Dataset:

Model Name : Inceptionv3

Research Paper Accuracy (%):85

Implemented Accuracy (%):82

Reasons for Performance Differences in Implemented Models

VGG16 Overperformance (89%) Potential overfitting due to small dataset and high parameter count.

Model might have learned dataset-specific patterns too well, leading to high accuracy.

Underfitting in AlexNet (62%)

Too many frozen layers, preventing the model from learning complex patterns. Learning rate might be too high, causing poor convergence.

Suggestions for Improvement

1. Fine-Tuning of Models

Unfreeze deeper layers for AlexNet and InceptionV3 for better feature extraction. Train with lower learning rates to improve convergence.

2. Increase Epochs and Batch Size Increase training epochs for InceptionV3 to improve model stability.

Fine-tune batch size to reduce overfitting in VGG16.

3. Enhanced Data Augmentation Use stronger augmentation techniques like zoom, contrast adjustments, and noise addition to improve model generalization.

Conclusion:

In this task, a comprehensive tomato leaf disease classification system was developed using pre-trained models such as AlexNet, VGG16, and InceptionV3. The dataset used was obtained from a publicly available Tomato Leaf Disease Dataset, which was carefully preprocessed and augmented to ensure high-quality input for model training.